

Les Matemàtiques en Joc



institut
Brugulat

Martí Juanola Ametller
2n Batxillerat C
Tutor: Jaume Bayó
Curs: 2017-2019

Síntesi

En aquest treball s'exploren les matemàtiques, la informàtica i els jocs de taula simultàniament. La recerca del treball es du a terme mitjançant una sèrie de projectes individuals i utilitzant llibres i plataformes en línia com a font d'informació. A més a més, cada projecte té la intenció d'aprofundir en diferents temes de la matemàtica per entendre la presència d'aquesta, en tota la seva extensió en els jocs. De manera introductòria, també hi ha uns apartats dedicats a comprendre conceptes més abstractes com l'atzar o el perquè dels jocs.

Este trabajo se basa en la exploración de la matemática, la informática y los juegos de mesa simultáneamente. La estructura se resume en la creación de proyectos individuales, y como fuentes de información, se han utilizado libros especializados en los campos de investigación y plataformas en línea. Cada uno de los proyectos tiene como foco la ampliación de aspectos de la matemática, con el objetivo de hacer mención a la relación entre las matemáticas y los juegos. Otros conceptos abstractos, como el azar, también son estudiados como objetivo menor del trabajo.

The aim of this project is to study the topics of mathematics, informatics and board games, blending all three themes simultaneously. The information used in the different sections of this project is extracted from specialized books, other studies and online platforms. The individual objectives of the different projects within this work, are to mix aspects of mathematics with games in general, and as a minor objective, this work also deepens into abstract concepts as randomness and the games themselves.

Agraïments

Primer de tot, m'agradaria agrair al meu tutor d'aquest treball de recerca, en Jaume Bayó. Ell ha estat la persona que m'ha guiat durant tot aquest procés i m'ha aconsellat de manera que aquest treball de recerca arribés a bon port. No només això, sinó que ell també és un dels culpables de que m'interessin les matemàtiques, i per tant, també que el treball s'hagi dedicat a aquest tema.

Mil gràcies també als meus pares i a la meva germana pel seu suport durant tot aquest treball. Ells han estat els que m'han transmès les ganes i l'interès per explorar tots aquests temes sorprenents i interessants en els quals he pogut aprofundir en aquest treball. Gràcies als seus consells, les seves idees i el temps que m'han dedicat, aquest projecte és el que és.

També agrair a en Jaume Padrés per facilitar-me el procés de la creació de la web, cedint-me l'espai web que s'ha utilitzat en el projecte.

Dono gràcies també a l'Arnau Padrés, pel seu suport constant en la creació de la pàgina web. Per ajudar a construir-la i per encarregar-se de certs aspectes del disseny d'aquesta.

Per últim, donar les gràcies a tots els altres amics, companys i família amb qui he tingut el plaer de poder jugar.

Índex

Introducció	6
1. Jocs i jocs de taula	8
1.1. Per què juguem?	8
1.2. Història i popularització dels jocs de taula	11
1.3. Tipus de jocs de taula	15
2. Jocs i matemàtiques	18
2.1. Història dels jocs i les matemàtiques	18
2.1.1. Tchuka ruma	19
2.1.2. Torre de Hanoi	25
2.2. Jocs de taula moderns i les matemàtiques	28
2.3. Atzar i sort	29
2.3.1. Què entenem per atzar?	29
2.3.2. Tipus d'atzar	30
2.3.3. L'atzar limita l'estratègia?	32
3. Creació d'una pàgina web	36
3.1. Objectiu	36
3.2. Procediment i desenvolupament	36
3.3. Estructura general	39
3.3.1. Fonaments	39
3.3.1.1. HTML	39
3.3.1.2. CSS	42
3.3.1.3. PHP	44
3.3.1.4. SQL	45
3.3.1.5. JavaScript	47
3.3.2. Base de dades	48
3.3.3. Diferents programes i estructures	51
3.3.3.1. Estructura bàsica i barra lateral	52
3.3.3.2. Seguretat	53
3.3.3.3. Usuaris i jugadors	54
3.3.3.4. Jocs i marques	55
3.3.3.5. Partides	56

3.4. Producte final i funcionalitat	57
3.5. Possibles canvis i millores	58
4. Temes d'anàlisi	59
4.1. Conjunts i Sets	59
4.1.1. Aplicació en jocs (Fillers: Set)	59
4.1.2. Anàlisi	60
4.2. Poliòminos i tessel·lacions	61
4.2.1. Aplicació en jocs (Blokus)	63
4.2.2. Projecte: ocupació d'espai	63
4.3. Combinatòria i probabilitat	65
4.3.1. Aplicació en jocs (Sushi Go!)	66
4.3.2. Projecte: creació d'un algorisme	66
4.3.2.1. Objectiu	66
4.3.2.2. Fonaments i estructura	67
4.3.2.3. Producte final i funcionalitat	70
4.3.2.4. Possibles canvis i millores	70
5. Conclusions	72
Bibliografia i webgrafia	74
Annexos	1
Annex I: normes del joc Set	1
Annex II: tessel·lacions	2
Annex III: normes del joc Blokus	4
Annex IV: tessel·lacions amb els poliòminos del joc Blokus:	5
Annex V: normes del joc Sushi Go!	9
Annex VI: programa sg5	12

Introducció

Motivació

Des del primer moment, hem tingut clar que volíem dedicar aquest treball a explorar un tema en el qual tingués un cert interès, per poder-hi aprofundir i explorar-lo a fons. Alguns dels temes que teníem presents a l'hora de plantejar el treball eren la música, algun àmbit de les ciències en general i altres aficions com els jocs de taula, o la informàtica. Al final, no ens vam centrar en un sol tema sinó que en aquest treball es tracten dos temes principals: les matemàtiques i els jocs de taula.

Aquests dos àmbits, les matemàtiques i els jocs, ja es mesclen en una branca de la matemàtica anomenada Teoria de Joc, aplicada majoritàriament en camps d'economia, i on es poden trobar matemàtics com John von Neumann i John Nash que varen realitzar importants avenços en aquests àmbits. Aquesta branca de la matemàtica, per tant, té un paper molt important en aquest treball, juntament amb la informàtica, ja que és un dels fonaments que hem utilitzat per realitzar diferents projectes.

Objectius

Els objectius del treball són els següents:

- El primer i com a objectiu més important del treball, és realitzar dos projectes on es relaciona la matemàtica, els jocs de taula i la informàtica, per explorar aquests temes de manera simultània. Els dos projectes són la creació d'una pàgina web amb l'objectiu de recollir resultats de partides de jocs de taula i poder-ne extreure dades estadístiques, i com a segon projecte, la creació d'un algorisme capaç de jugar a un joc de taula de tipus probabilístic. Aquests dos projectes, per tant, tenen molta base informàtica però els tres temes principals del treball (les matemàtiques, la informàtica i els jocs) hi són sempre presents.
- En segon lloc, i d'una manera més general, aquest treball té l'objectiu de trobar les relacions que tenen les matemàtiques i els jocs de taula, o jocs en general. En aquest objectiu també hi té part l'aprenentatge i l'anàlisi de conceptes com l'atzar i de temes com la combinatòria o les tessellacions aplicades en jocs.

Metodologia

La metodologia d'aquest treball es resumeix en l'aprenentatge de conceptes relacionats en els tres temes del treball, el plantejament dels dos projectes de l'objectiu principal, la creació d'aquests dos projectes i l'exploració dels temes d'anàlisi. També hi té lloc una part de recerca teòrica realitzada mitjançant llibres especialitzats en camps de la matemàtica i diferents llocs web d'Internet.

La part d'aprenentatge de conceptes bàsics i altres coneixements per la creació del web s'ha realitzat de manera autodidàctica, utilitzant plataformes virtuals. En concret, la plataforma d'impacte mundial *Khanacademy*.

Estructura

Si ens centrem en l'estructura del treball, el podem dividir en quatre parts diferenciades. En primer lloc, tenim un bloc sobre els jocs de taula en general, amb un objectiu introductori al tema on es parla de perquè juguem com a concepte global, sobre què són els jocs de taula, com s'han popularitzat i els diferents tipus que existeixen. El segon bloc és una continuació del primer on es parla de matemàtica i joc. El tercer bloc tracta del projecte de la creació del web, amb tot el procediment explicat i els diferents programes i coneixement utilitzats. I per últim, hi ha el quart bloc en el qual es parla de diferents temes d'anàlisi, que consisteixen a escollir un tema matemàtic, com ve a ser la combinatòria, i aplicar-lo en jocs de taula, on el tema en si és present. Per exemple, en el cas de la combinatòria, podem trobar jocs com el *Sushi Go!* o el *7Wonders* que hi tenen una estreta relació i que s'expliquen en l'apartat corresponent.

D'aquests quatre blocs, els dos primers són més teòrics i conceptuals, i en canvi, els dos següents incorporen la part pràctica del treball on es pot trobar la realització dels dos projectes.

1. Jocs i jocs de taula

1.1. Per què juguem?

Els humans, com altres animals, juguem. El fet de jugar és una característica pròpia dels éssers humans, un tret que adquirim des que naixem i que ha estat present en l'espècie humana des del començament. Però com és que juguem a jocs de taula, a videojocs o competim en esports? Com és que tenim una necessitat de competir amb altres humans quan no és un factor aparentment important per a la nostra supervivència?

Chris Crawford¹ (1990-1991) descriu un esquema per definir el concepte de joc. Crawford diu que un element que entreté i amb el qual es pot jugar, però no té un objectiu, es pot considerar una joguina. I en el cas contrari, si existeix un objectiu, es pot considerar un repte. Per tant, seguint aquesta classificació, també podem esmentar que si una joguina adquireix un objectiu o unes normes pot ser considerada un repte també. Continuant, si un repte no té un altre jugador amb qui competir, es considera un puzzle, i en cas de tenir-ne es considera un conflicte. Per últim, si un conflicte no permet la interacció entre participants, és una competició. I en cas que es permeti aquesta interacció, és un joc.

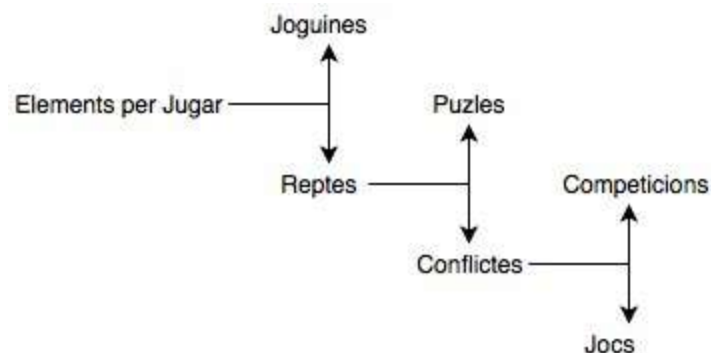


FIGURA 1. Esquema de definició de joc segons Chris Crawford. FONT. Elaboració pròpia mitjançant *draw.io*: plataforma oberta per crear i compartir diagrames.

Per tant un joc segons Crawford és un element amb el qual es pot jugar, que té un objectiu i unes normes clares i on hi ha altres jugadors amb els quals es creen interaccions. Però, seguint aquesta estructura, també podem considerar la vida en si un joc: en la vida hi ha unes normes i uns objectius, hi ha altres “jugadors” involucrats i pots relacionar-te amb aquests.

¹ Dissenyador de videojocs americà conegut per fundar *The Journal of Computer Game Design* i la *Game Developers Conference*.

Aquesta similitud és molt important a l'hora d'entendre perquè juguem. Segons Abraham Maslow² (1954), els humans tenim una sèrie de necessitats amb un ordre de preferències. Aquesta jerarquia té diferents pisos:

1. **Necessitats Físiques:** les necessitats més bàsiques com menjar, beure, dormir, o respirar.
2. **Necessitats de seguretat:** corresponen a necessitats com sentir-se segur, sentir-se protegit o controlar els plans de futur pròxims.
3. **Necessitats de pertinença i d'amor:** el fet de sentir-se acceptat i part d'un grup.
4. **Necessitats baixes d'estima:** el fet de sentir-se respectat i de pertànyer en un estatus social. També hi té lloc la possessió de poder i la dignitat d'un mateix.
5. **Necessitats altes d'estima:** respectar-se a un mateix, ser competent i aconseguir la independència i la llibertat com a individual.



FIGURA 2. Piràmide de les necessitats humanes de Maslow. Font. Article de *Wikipedia - Maslow's hierarchy of needs*

Totes aquestes necessitats les tenim presents en el nostre cos d'una manera o altra, i per aconseguir que els individus les resolguin, el nostre cervell té un sistema de recompenses. En els casos dels primers nivells és molt obvi: si menges quan tens gana o beus quan tens set, aquestes sensacions incòmodes desapareixen i certes hormones relacionades amb els sentiments, com la dopamina³, s'alliberen per aconseguir un estat de plaer i agradable. En els casos dels nivells més alts, on les necessitats ja són psicològiques, aquest procés és més complex però acaba sent el mateix. Per tant, els objectius de la vida dels éssers humans es podrien definir com

² Psicòleg americà conegut per haver creat la piràmide de Maslow, on indica una jerarquia de les necessitats humanes.

³ Neurotransmissor produït en diverses parts del sistema nerviós i encarregat d'estimular el cervell creant sensacions de plaer.

la cerca de plaer i l'evasió del dolor, mitjançant certes recompenses per decisions i comportaments que satisfan necessitats.

Tornant a la qüestió de per què juguen els humans, podem veure que els jocs són maneres d'obtenir aquestes recompenses i aquests estats de plaer d'una manera més simple. Mentre la vida, que es pot considerar el joc més complicat i complex que existeix, té unes normes complicades, objectius indeterminats, recompenses dubtoses i respostes a les decisions preses a llarg termini o fins i tot inexistent, els jocs et proporcionen unes normes i objectius molt clars i concisos, i un *feedback* immediat que permet avaluar les teves decisions. Per tant, vist des d'aquest punt de vista, és lògic que els humans hàgim creat jocs dintre el joc global de la vida per poder obtenir unes recompenses fàcils d'aconseguir, entenedores i clares.

Aquests jocs poden ser tant videojocs, com jocs de taula o esports, però tots tenen unes característiques per aconseguir una particular atracció. Perquè un joc t'agradi i produeixi un estat de plaer, cal que es produeix un efecte descrit per Mihály Csíkszentmihályi⁴ (1990) com un estat on s'està completament immers en una activitat, on qualsevol preocupació relacionada amb la vida d'un desapareix i la percepció del temps canvia. Csíkszentmihályi també compara aquest estat amb la sensació de tocar Jazz, on una acció ve immediatament després de l'altre, i on tot un està dedicat a aconseguir un objectiu. Perquè aquest efecte succeeixi Csíkszentmihályi va plantejar 8 característiques que han de tenir els jocs:

1. **Objectiu assequible i atractiu:** perquè un joc sigui atractiu a un jugador i perquè es produeixi aquest estat de plaer necessita exigir unes certes habilitats als jugadors. D'aquesta manera aquest no se sent ni avorrit perquè és exageradament fàcil, ni desanimat perquè el nivell requerit és massa gran pel jugador.
2. **Capacitat de concentració:** un joc també necessita que els jugadors només s'hagin de preocupar per les decisions que han de prendre, i no per altres detalls com controls, en el cas de videojocs, o preparacions laborioses. Això desferia aquest estat de despreocupació i el jugador no podria immersir-se tant en el joc.
3. **Objectius clars:** en qualsevol cas un joc ha de tenir un/s objectiu/s clar/s perquè el jugador pugui dirigir les seves accions cap a una meta. Aquests objectius també han de ser adequats pels jugadors perquè el repte no sigui ni impossible ni massa fàcil, tal com s'ha dit en la característica u.

⁴ Psicòleg considerat per molts l'investigador més important en el tema de la psicologia positiva.

4. **Feedback constant:** per a poder dictaminar si les decisions que s'han preseren encertades o no, el joc ha de respondre als teus actes constantment i de manera clara.
5. **Gran immersió:** en ordre d'estar immersos en el joc aquest no pot tenir elements que són obligats als jugadors o que esdevenen repetitius i pesats.
6. **Capacitat de controlar l'entorn:** aquesta és una sensació de tranquil·litat i d'estabilitat, i per tant un joc ha de proporcionar aquest control als jugadors. En cas que hi hagi una manca de control, un dels objectius del joc ha de ser obtenir aquest control o avançar en aquesta direcció.
7. **Disminució de l'autoreflexió:** un joc ha d'eliminar, dintre del que sigui possible, l'autoreflexió. El jugador s'ha de centrar únicament en el joc.
8. **Percepció del temps alterada:** un altre efecte d'un bon joc és que la percepció del temps dels jugadors varia, i sembla que les estones passin més ràpidament.

En definitiva, humans i animals juguem a jocs a causa de com està pensat el nostre cervell i per com el nostre organisme ens recompensa en satisfer les nostres necessitats. La diferència entre la complexitat dels jocs entre humans i altres animals es produeix a causa de la nostra posició estable en la piràmide de les necessitats de Maslow. Com que nosaltres, els humans, no hem de patir tant per la nostra supervivència, tenim més temps per jugar o estimular-nos amb els jocs, i per això tenim tants esports i jocs diferents amb una importància tan gran en la nostra societat.

1.2. Història i popularització dels jocs de taula

Com s'ha dit a l'apartat 1.1, els humans hem jugat des del nostre inici, per tant és lògic que els jocs de taula hagin existit amb nosaltres durant molt de temps (“*Board Game*” - Wikipedia, 2018). En realitat, els primers jocs coneguts que es consideren jocs de sobretaula daten aproximadament fa 5000 anys. El més antic s'anomena Senet i es va trobar en uns enterraments predinàstics d'Egipte, als voltants del 3500 aC. També se'n trobaren d'altres a Iran on hi havia la civilització Jiroft. I també al voltant d'aquesta època es va originar el clàssic joc *Backgamon* a Pèrsia.

Saltant endavant en el temps, també trobem el joc de taula clàssic més conegut arreu del món, els escacs. Juntament amb *Pachisi* o *Chaupar*, altres jocs de taula semblants al parxís d'avui en dia, es van originar a l'Índia medieval. A la Xina es van originar jocs com el *Go* o el *Liubo*, a l'Àfrica de l'est jocs com el *Bao* i altres també a la Mesoamèrica. Per tant, observant la història podem veure que en qualsevol civilització prou desenvolupada hi van aparèixer jocs de taula que representaven el poder i la influència que tenien dintre la societat, ja que demostraven que la

supervivència no era la seva primera preocupació, sinó que vivien suficientment bé per poder dedicar temps a jugar.

Tot i això, no va ser fins al segle XX que els jocs de taula van obtenir una certa popularitat. Això era degut al fet que durant l'edat mitjana i en part de l'edat moderna les societats no eren prou estables i hi havia preocupacions més importants que no pas crear una indústria de jocs. En aquestes èpoques els jocs de taula estaven reservats a classes més altes, exceptuant els jocs populars de cartes o daus. Però aquests no van destacar gaire degut a certes condicions i prohibicions. Per exemple, a l'Amèrica colonial del segle XVII i XVIII, la vida agrícola no permetia gaire temps per a jugar, i elements com els daus, o les cartes eren relacionats amb el diable segons descendents religiosos del Regne Unit.

Per tant, a partir del segle XIX i XX comencem a trobar dues grans potències en la creació dels jocs de taula. La primera gran potència va ser els Estats Units d'Amèrica. A mesura que la societat va anar canviant d'agrícola a urbana, el benestar general també es va incrementar i les famílies tenien més temps de lleure. Això va impulsar la creació de certs jocs de taula com *Traveller's Tour Through the United States*, que es considera el primer joc de taula publicat als Estats Units, l'any 1822. Durant la resta del segle XIX s'anaren publicant jocs de taula de temàtica o moralitat cristiana com *La mansió de la Felicitat* l'any 1843. Però els jocs de taula dels Estats Units d'Amèrica al segle XX van canviar de temàtica, i en comptes de moralitat religiosa, trobem jocs basats en el capitalisme. El primer d'aquests va ser *The Game of the District Messenger Boy*, que tractava de com un nen de classe baixa podia esdevenir una persona rica i benestant. Aquests jocs de competició capitalista van culminar l'any 1935 amb la publicació de *Monopoly*, el joc més venut dels Estats Units i probablement de tot el món. Es considera l'edat d'or dels jocs de taula a Amèrica l'etapa entre l'any 1880 i 1920.

La segona potència i la més important, va ser Europa, més específicament Alemanya. Al voltant dels anys 60', a Alemanya, país on els jocs de taula tenien una importància especial, es començarien a produir uns jocs de taula moderns que posteriorment s'anomenarien *Eurogames* i esdevindrien els jocs de taula més populars i coneguts arreu del món. A causa dels efectes de la postguerra a Alemanya, aquests jocs sortien dels models de jocs americans que es basaven en guerres i batalles i se centraven més en nous estils de joc on tots els jugadors poden jugar fins al final, de manera que els que tenen una posició més bona en la partida se'ls exigeix més. Aquests jocs de caràcter més abstracte, van anar agafant força durant els anys 70' i 80' a Alemanya, país que esdevindria el que vendria més jocs per càpita del món l'any 2009. Ràpidament aquests tipus de jocs de taula van sortir d'Alemanya escampant-se per altres països d'Europa com França o Holanda.

L'any 1995, amb la publicació de *The Settlers of Catan*⁵, aquest moviment s'escamparia arreu del món i causaria que la indústria de la producció de jocs de taula agafés força. A causa de l'èxit d'aquest joc de taula, aquests obtindrien certa popularitat dintre la societat. *The Settlers of Catan*, però, no va ser el primer *Eurogame* que va sortir d'Europa, però sí que va ser el que va expandir el moviment i va precedir altres jocs molt populars com *Carcassonne*, *Puerto Rico* o *Ticket to Ride*. Juntament amb la popularització del nom *Eurogame*, també se'n va afegir un altre per denominar els jocs que venien dels Estats Units d'Amèrica, que consistia en *Ameritrash*. Aquest nom es va crear de la unió de les paraules Amèrica i *Trash*⁶, ja que els jocs americans comparats amb els *Eurogames* no se centraven tant en les mecàniques dels jocs i en com aconseguir una bona dinàmica entre els jugadors. En comptes d'aquests requisits, en tenien d'altres com l'estètica, els combats un contra un i la gran presència de sort.

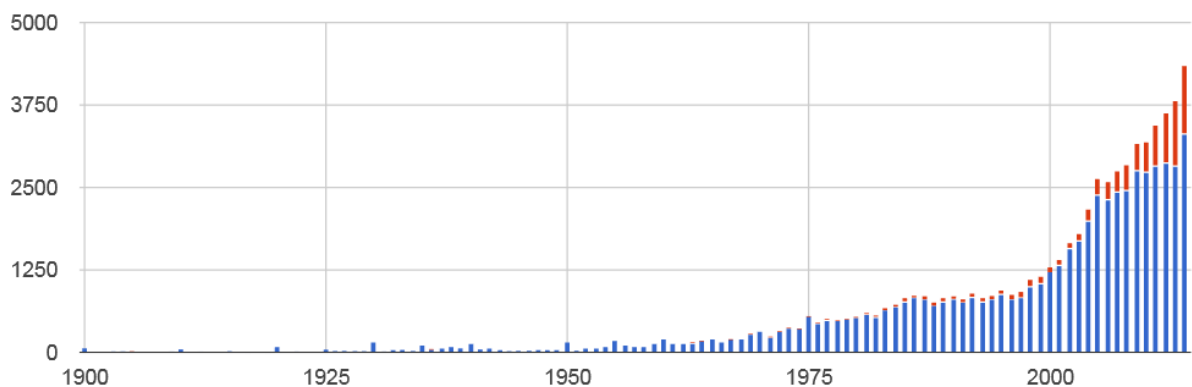


FIGURA 3. Publicació de jocs de taula per any (1900-2014). Font. Article de *Wikipedia - Board game*

Amb l'expansió dels *Eurogames*, els jocs de taula en general van agafar popularitat, i a partir dels anys 90', tant el nombre de publicacions com el nombre de compres de jocs de taula es van incrementar considerablement en el que s'anomena l'edat d'or dels jocs de taula. Però un altre factor important en la popularització dels jocs de taula és la tecnologia, especialment la plataforma d'Internet.

Una primera raó és el fet que Internet va permetre unir gent de tot el món amb idees i maneres de pensar iguals o similars, i així fer que aquests jocs de taula que només es coneixien a Alemanya, s'escampessin per tot el món i alhora inspirassin altres a

⁵ *Eurogame* molt popular de l'any 1995, del dissenyador de jocs de taula Klaus Teuber. Se n'han fet diverses expansions i versions, totes basades en el mateix concepte de gestió de recursos i territoris hexagonals.

⁶ Paraula anglesa que significa escombraries i usada vulgarment per referir-se a elements inútils. Altres paraules utilitzades en el mateix context poden ser *garbage* o *rubbish*.

crear-ne també (Cross, 2017). Matt Leacock⁷, per exemple, explicava "*I remember we used to rely on these little hobbyist websites that would do amateur translations into English of all the new German games that were coming out*"⁸. Són aquestes plataformes les que van ajudar a escampar la cultura del joc.

Per altra banda Internet va permetre que jugadors i autors o dissenyadors poguessin comunicar-se fàcilment. Això va servir perquè els dissenyadors de jocs fossin capaços de provar els jocs de taula nous amb una mostra més petita de la gent i poder corregir-los en cas de ser necessari. També va permetre que jugadors poguessin saber quan es publicaven nous jocs i també quan es realitzava qualsevol trobada o exposició de jocs de taula.

Un altre punt molt important va ser que, amb les noves tecnologies i amb Internet, la indústria de videojocs també va agafar una força molt gran a la societat. I això, encara que no ho pugui semblar, també va fer que alguns busquessin una alternativa a aquests jocs de pantalla menys personals. Els jocs de taula eren i són una alternativa perfecta on la relació entre els jugadors no és a distància i permeten situacions com la que diu D’Juan Irvin, especialista en un curs de disseny de jocs . "*It’s a beautiful thing to be able to get people together in the same place, playing the same game. There’s a camaraderie aspect to it*"⁹.

Per últim, a causa de tots aquests canvis que ha causat la tecnologia a aquesta indústria, s’ha aconseguit que es desenvolupin jocs nous i diferents, que no tenen cap relació amb els clàssics. També s’entreguen premis als millors jocs de cada any per incitar a noves idees i noves maneres de jugar. Uns dels més coneguts són *Spiel des Jahres*, *Deutscher Spiele Preis*, *International Gamers Award* o *Origins Award*.

Per tant, ara estem en un punt on els jocs de taula tenen una importància en la nostra societat que no havien tingut mai abans a causa d’una millor qualitat i unes característiques que altres jocs, com els d’ordinador, no poden tenir. I encara que en un moment aquests només fossin coneguts a Alemanya i Amèrica, actualment s’han estès arreu del món. Gràcies al seu èxit, avui en dia també es celebren trobades i competicions, i en qualsevol gran ciutat es poden trobar associacions que difonen els jocs de taula.

⁷ Dissenyador de jocs de taula, conegut per haver creat jocs com *Pandemia*, *Forbidden Island* o *Forbidden Sky*.

⁸ "Recordo quan havíem d'utilitzar una d'aquelles pàgines web d'aficionats que ens proporcionaven traduccions poc professionals a l'anglès de tots els nous jocs alemanys que sortien a la venda"

⁹ És preciós poder reunir persones en el mateix lloc, jugant el mateix joc. Té un sentit de camaraderia."

1.3. Tipus de jocs de taula

Per ordenar els nombrosos jocs de taula, han sorgit classificacions segons diversos factors com la que fa *ViaLúdica*¹⁰. Tot i això, no hi ha una classificació estàndard, ja que no hi ha una necessitat d'ordenar-los d'una manera universal. Aquí se n'esmentarà una realitzada per un usuari del web *BoardGameGeek*¹¹ (David F., 2013) que té un nivell de detall sorprenent i que pot servir per entendre l'abast dels jocs de taula.

Aquesta classificació alternativa ordena els jocs de 3 maneres diferents, segons el format de joc, el seu gènere i finalment la mecànica del joc. La classificació segons format és la més global i fa referència a la interacció entre els jugadors. Es defineixen 7 divisions.

- **Competitiu:** jocs on tots els jugadors són rivals. S'hi poden englobar la gran majoria de jocs de taula.
- **Cooperatiu:** tots els jugadors formen un equip i tenen un mateix objectiu. Ex. *Illa misteriosa*
- **Coordinatiu:** tots els jugadors formen un equip però cada jugador té un rol únic i no intercanviable entre la resta de jugadors. Els jugadors juguen contra el joc de taula en si. Ex. *Mysterium*
- **Semi-cooperatiu:** jocs on un jugador competeix contra tota la resta que formen un sol equip. Ex. *Descendència*
- **Equips i parelles:** diferents equips o grups de jugadors es disputen la victòria. Ex. *Pictionary*
- **Dos contra dos:** dos equips de dos jugadors competeixen entre ells. Els jocs d'aquests casos també poden ser coordinatius. Ex. *Codi Secret*
- **Relacions secretes:** els jugadors competeixen entre ells seguint uns equips i relacions que no es coneixen i que a mesura que avança la partida es poden anar deduint. Ex. *Bang*

La segona classificació fa referència al gènere i al tema del joc. Aquesta es divideix en 6 tipus, els quals tenen una sèrie de subtipus més concrets:

¹⁰ Botiga especialitzada en jocs de taula moderns i grup centrat en la promoció de la cultura del joc.

¹¹ La plataforma virtual més popular centrada en els jocs de taula. Proporcionen tant servei de compra i venda com informació de jocs de taula com blogs públics destinats a aficionats i professionals.

- **Estratègia abstracta:** aquests tipus de jocs tenen una gran presència d'estratègia i la majoria tenen relació amb geometria i/o tessel·lacions¹². Hi podem trobar jocs d'alineació, de captura, de desgast, d'escapar, de limitació de zones, de control d'àrees o de connexions. Uns jocs que entrarien en aquesta divisió podrien ser el *Blokus*, els escacs o el *Connect Four*.
- **Eurogame/estratègia/familiar:** aquests jocs són també d'estratègia però destinats a un àmbit més familiar, i no tenen components abstractes com en el punt anterior. També s'anomenen *Eurogames* perquè la majoria d'aquests són creats a Europa. Alguns dels subtipus d'aquest apartat són jocs de creació de xarxes, de col·locació de lloses, de gestió de riscos o d'eficiència de sistemes.
- **Ameritrash/temàtic:** en contraposició dels *Eurogames*, trobem la categoria d'*Ameritrash*, on trobem jocs encarats més cap a la negociació, la deducció o el desenvolupament de personatges. Són jocs amb una mecànica regular però amb una estètica i una immersió més elaborada que els *Eurogames*.
- **Jocs de cartes:** aquests jocs tenen una base material de cartes i hi podem trobar subtipus com jocs de trucs o combinacions o de duels.
- **Jocs de guerra:** aquests jocs estan centrats en guerres i exèrcits, i en podem trobar de dos tipus: un és d'influència d'àrees, on el control de certes zones del taulell aporten beneficis per obtenir exèrcits més grans i alhora més poder, on podem trobar jocs com el *Risk*. L'altre tracta de guerres o batalles centrades en la planificació i la tàctica. En aquests darrers no es pot guanyar només tenint un exèrcit gran sinó que es necessita deliberar sobre les accions que es prendran.
- **Socials/de festes:** en aquests jocs les habilitats físiques i la pràctica hi tenen un paper més important. Hi trobem jocs de destresa com el *Twister*, de velocitat i capacitat de reconeixement com el *Jungle Speed* o el *Set*, de caràcter més psicològic com el *Dixit* o d'altres centrats en les paraules com *Scrabble*.

Per últim, tenim la classificació segons mecàniques de joc, on trobem 8 divisions principals.

- **Accions espacials:** jocs amb mecàniques centrades en un espai, normalment en dues dimensions, sobre el qual es desenvolupa el joc. Dintre d'aquesta divisió hi trobem jocs de moviment espacial com els escacs, de posicionament espacial com el *Carcassone* o el *Blokus*, de la cerca de camins o rutes, de moviments forçats o de punts en l'espai. En el cas del moviment i el posicionament espacial encara es poden dividir més segons com pots moure o col·locar les peces o els elements del joc.
- **Accions no-espacials:** aquestes mecàniques se centren en accions que no modifiquen l'estructura de l'espai del joc, o simplement que no en tenen. Algun exemple pot ser *Dominion* o *Pandemic*.

¹² Consultar apartat 4.2.

- **Altres accions no-espacials:** a diferència de les anteriors, les accions dintre d'aquestes mecàniques tenen una relació amb l'espai del joc però no de manera tan directa com en el cas de les accions espacials. Podem trobar jocs de *drafts*¹³ d'accions com *Agricola*, de col·locació de treballadors com *Els pilars de la Terra*, o d'altres com jocs d'accions simultànies, de presa de decisions o de votacions.
- **Subhastes:** aquests jocs estan centrats en les subhastes i conseqüentment en algun tipus de moneda. Jocs com *Chicago Express* o *Modern Art* entrarien dintre d'aquesta categoria.
- **De cartes:** on trobem jocs de *drafts* de cartes, de col·locació de cartes, de trucs o d'apilar cartes. Hi trobem jocs com *7Wonders* o *The Settlers of Catan*, encara que aquest últim les cartes no siguin l'únic component material del joc.
- **De daus:** mecàniques de joc on els daus serveixen per indicar accions de moviment o altres accions de les quals s'ha d'escollir només una. *Monopoly* en seria un exemple.
- **Socials:** a diferència de les anteriors divisions, tenim jocs on la interacció i interpretació dels jugadors és clau per poder jugar. Trobem jocs basats en la creativitat, en les accions d'endevinar, d'interpretacions, d'acords entre jugadors o d'apostes. *Dixit* o *The Settlers of Catan* en serien uns exemples evidents.
- **Sistemes de joc:** per últim, les mecàniques de jocs que no han entrat en cap altra categoria les trobem aquí, amb jocs de pagament de recursos, d'especulacions, de torns de jugadors variables, de subministraments i demandes o d'envelliment. Hi podem trobar jocs com *Ciudadelas* o *Puerto Rico*.

¹³ Repartició o assignació d'elements als jugadors. Es poden realitzar *drafts* d'accions, de cartes, etcètera.

2. Jocs i matemàtiques

2.1. Història dels jocs i les matemàtiques

L'anàlisi dels jocs és un element propi dels matemàtics (Ibáñez, 2015). Preguntar-se com puc guanyar aquest joc, quina és la millor estratègia o quina probabilitat hi ha que passin aquesta sèrie d'esdeveniments, són preguntes que estan dintre els instints naturals dels matemàtics. Aquestes preguntes i d'altres s'han estat formulant en diferents jocs constantment.

Aquest impuls de resoldre trencaclosques o trobar una estratègia guanyadora d'un joc té una similitud molt gran a la resolució de problemes matemàtics. El procediment en els dos casos és molt similar i el coneixement d'un dels dos t'ajuda a entendre l'altre. Per exemple, algunes bases matemàtiques et poden ajudar a trobar millors estratègies en els jocs, i encara que no ho sembli, quan jugues estàs aplicant certes relacions i intuïcions matemàtiques en les teves decisions. Aquestes dues maneres d'afrontar problemes, s'anomena pensament matemàtic.

Per resoldre un problema matemàtic, podem dividir el procediment en 4 simples passos:

1. **Familiaritzar-se amb la situació:** el primer pas per resoldre un problema o afrontar un joc de taula és conèixer-lo bé i entendre'n les normes i el context. També és útil conèixer els elements i les característiques d'aquests reptes.
2. **Buscar estratègies:** una vegada familiaritzats amb el problema o joc, el següent pas és trobar un camí per arribar a la solució. Aquesta estratègia es pot aconseguir seguint diferents procediments: experimentar primer i trobar relacions que es poden aprofitar per obtenir uns resultats volguts; simplificar el problema, i intentar resoldre aquests de manera més senzilla; utilitzar estratègies d'altres jocs que es puguin aplicar de manera total o parcial en el que s'està treballant; utilitzar el recurs matemàtic de la reducció a l'absurd, és a dir la suposició d'un resultat contrari al que es vol obtenir. En el cas dels jocs de taula o els trencaclosques es consideraria que el joc no té solució.
3. **Aplicar l'estratègia:** el següent pas és aplicar l'estratègia que s'ha trobat, o la que s'ha considerat millor.
4. **Aprendre i generalitzar:** per últim, després d'aplicar l'estratègia, se n'han d'extreure conclusions, i amb aquestes millorar l'estratègia o canviar-la per complet.

Podem veure que aquests passos són perfectament aplicables quan es juga a un joc de taula o quan s'intenta resoldre un trencaclosques. No només comparteixen els mateixos passos de resolució, sinó que gràcies a la proximitat dels jocs de taula i les matemàtiques, aquestes últimes ajuden a crear jocs nous. En la majoria de casos, els dissenyadors de jocs utilitzen les matemàtiques per controlar aspectes com la probabilitat de certs esdeveniments o la dificultat de possibles reptes.

En els següents dos apartats es parlarà de dos trencaclosques o puzles (segons l'esquema de Crawford) amb unes normes simples per poder-los entendre amb més profunditat, i com altres matemàtics els han analitzat i hi han trobat estratègies i relacions. Aquests trencaclosques seran utilitzats com a *toygames* tal com els defineix Richard Gardner¹⁴ que són petits trencaclosques o jocs, que serveixen per demostrar algun fet o alguna relació però que no són interessants per jugar. Alguns altres exemples poden ser *Rando-Chess*¹⁵ o altres petits jocs com el proposat per Gardner on donat un nombre, els jugadors han d'intentar endevinar el dígit de pi en la posició decimal que correspon el nombre. Aquests jocs no són interessants per jugar, però sí que poden demostrar o reafirmar conceptes com l'atzar.

2.1.1. *Tchuka ruma*

El trencaclosques *Tchuka ruma* és un joc de la família *Mancala*. Aquests són uns dels jocs de taula més antics que s'han pogut trobar, originats a diferents zones del món com en el sud de l'Índia, països com Kenya o Tanzània de l'est d'Àfrica o països del sud d'Àsia. Al voltant de totes aquestes regions s'han creat centenars de versions, on totes tenen unes característiques en comú, però amb maneres de jugar diferents. Els jocs de la família *Mancala* també s'anomenen de "sebra i recol·lecció", i tots tenen dos elements claus: uns forats, depressions o cases,



FIGURA 4. Joc *Mancala* africà. Font. Article de Sadigh Gallery - *BOARD GAMES/Mancala*

¹⁴ Dissenyador de jocs de taula i matemàtic conegut per haver creat el popular joc de cartes *Mafic: The Gathering* entre molts altres.

¹⁵ Joc que consisteix en una partida d'escacs normal, però al final del joc, el que seria el guanyador tira un dau. En cas que el dau resulti amb un 1, el jugador que hauria guanyat perd i l'altre guanya. En els altres casos el jugador "guanyador" guanya.

gravades al taulell que sol ser de fusta, i unes fitxes que normalment són pedres. Aquests dos elements poden representar els camps de cultiu i les llavors per sembrar.

Amb aquestes característiques de base, s'han creat diferents jocs tan competitius o d'un sol jugador o amb taulells simples o més complexos.

Tchuka ruma és un joc per un jugador i d'una sola fila de forats referenciat per primera vegada en la publicació *Jeu de la Tchouka* de Henri-Auguste Delannoy l'any 1892, on menciona la seva coneixença del joc a través d'Éduard Lucas. Per tant, aquest joc en particular és molt més recent comparat amb d'altres com *Ali Guli Mane*¹⁶, d'origen més antic. Segons experts es pensa que aquest joc podria provenir d'algun indret de l'Índia, Filipines, Maldives o Rússia, decantant-se per les regions de l'Índia o les Maldives a causa de l'etimologia del nom¹⁷. Tot i això no s'ha trobat cap joc amb aquestes normes fins i tot després d'haver realitzat una intensa recerca etnogràfica i, per tant, el joc es considera una creació d'Éduard Lucas. Es pensa però, que Lucas va conèixer primer un joc anomenat *Dakon* i a partir d'aquest va plantejar *Tchuka ruma* per adaptar-se més a la seva recreació matemàtica.

El taulell de *Tchuka ruma* consisteix d'una sola fila de quatre forats idèntics amb un forat de dimensions més grans que representa la *ruma*, una espècia de rebost, al final. Dintre cada forat petit, s'hi col·loquen dues llavors, i es deixa la *ruma* buida a la posició inicial del joc. Per tant en total tenim vuit llavors repartides per igual en quatre forats i, a part, la *ruma*.

El joc en si té per objectiu dipositar totes les llavors a la *ruma* mitjançant una sèrie de moviments de les llavors. Aquests moviments venen determinats per dues normes:

1. El jugador pot escollir les llavors d'un dels quatre forats i repartir-les una a una seguint un ordre ascendent en direcció a la *ruma*. Si després d'arribar a la *ruma* el jugador encara té llavors per dipositar, continua per la casella oposada a la *ruma*.

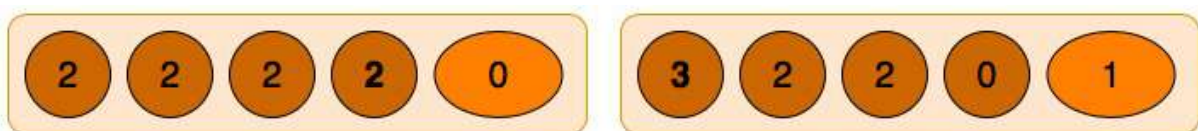


FIGURA 5. Posició inicial del joc *Tchuka ruma* i la següent en cas d'escollir el quart forat. FONT. Elaboració pròpia mitjançant *draw.io*: plataforma oberta per crear i compartir diagrames.

¹⁶ El nom d'aquest joc significa “bloc de fusta amb forats” i consisteix en un joc de la família *Mancala* per a dos jugadors amb un taulell de dues files de forats.

¹⁷ *rumah* significa “casa” i *chuka* és el nom d'un vinagre produït d'unes llavors.

2. Un cop el jugador es queda sense llavors per dipositar, poden esdevenir tres casos:

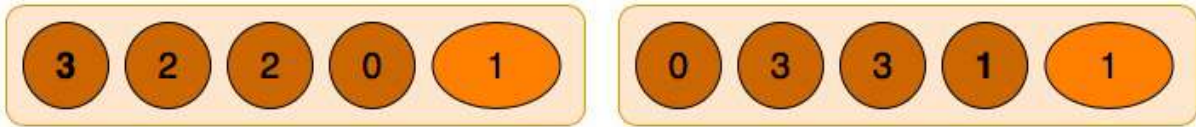


FIGURA 6. Continuació de la figura 5, on es pot veure un enllaç de dos sembres seguides. FONT. Elaboració pròpia mitjançant *draw.io*: plataforma oberta per crear i compartir diagrames.

- El jugador ha dipositat l'última llavor en una casella que contenia més llavors, o sigui que en conté més d'una. En aquest cas, el jugador ha d'escollir aquest mateix forat i repartir les llavors de la manera que s'ha explicat abans.
- L'última llavor s'ha col·locat a la *ruma*. En aquests casos el jugador pot triar les llavors dels forats que vulgui (que estigui disponible) per començar de nou a repartir les llavors.

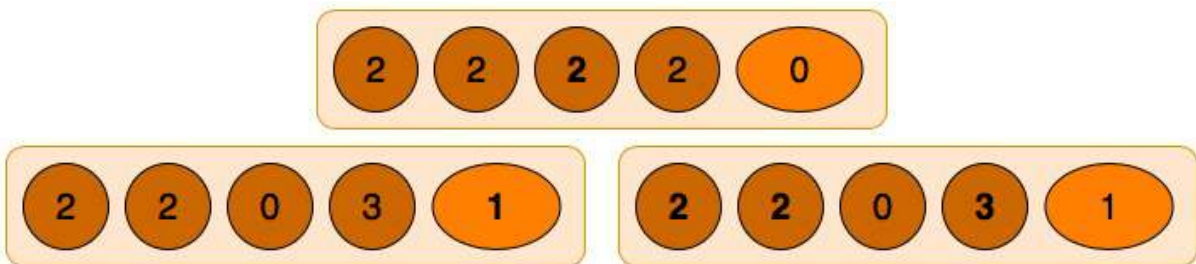


FIGURA 7. Seqüència de moviments que permet al jugador triar qualsevol forat (menys la *ruma*). FONT. Elaboració pròpia mitjançant *draw.io*: plataforma oberta per crear i compartir diagrames.

- Per últim, pot ser que l'última llavor es dipositi en una casella buida, deixant-la així amb només una llavor. En aquests casos el jugador perd la partida.

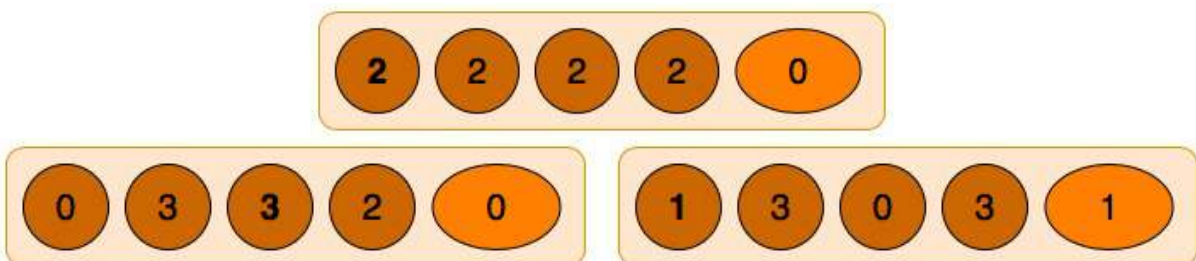


FIGURA 8. Seqüència de moviments que porten al jugador a la derrota. FONT. Elaboració pròpia mitjançant *draw.io*: plataforma oberta per crear i compartir diagrames.

Un cop se saben les normes i els elements del joc, ja es pot jugar. Es podria guanyar simplement usant el mètode de prova i error, però ja que el nombre de forats i llavors

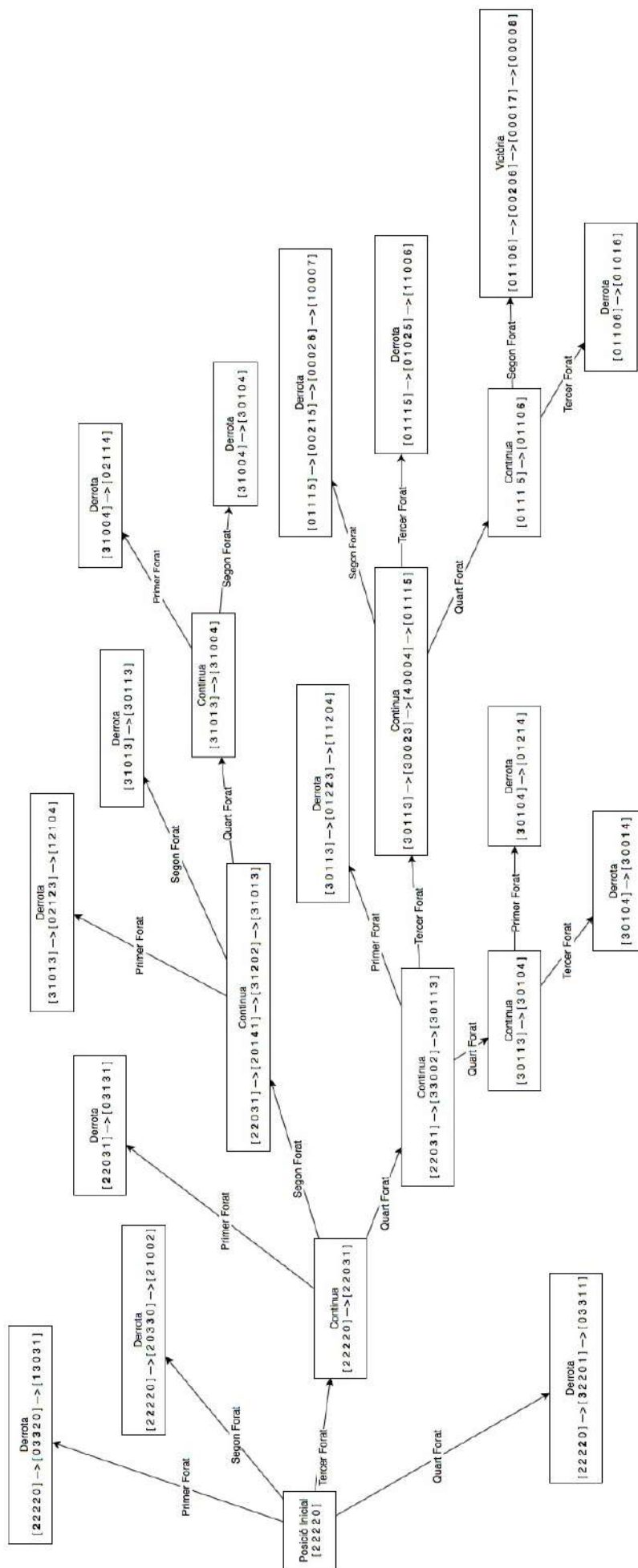


FIGURA 9. Tots els possibles moviments en el joc *Tchuka ruma* amb 4 forats i 8 llavors. FONT: Elaboració pròpia mitjançant *draw.io*: plataforma oberta per crear i compartir diagrames.

no és gaire elevat es poden observar totes les possibles partides i decisions¹⁸.

En la figura 9 podem veure tots els possibles resultats al joc *Tchuka ruma*, de les quals només una esdevé en victòria, i tota la resta són derrotes pel jugador. La seqüència d'estats que tindria el taulell durant la partida guanyadora és la següent:

[2 2 2 2 0] → [2 2 0 3 1] → [3 3 0 0 2] → [3 0 1 1 3] → [3 0 0 2 3] → [4 0 0 0 4] → [0 1 1 1 5] → [0 1 1 0 6] → [0 0 2 0 6] → [0 0 0 1 7] → [0 0 0 0 8]

Amb aquesta gràfica podem entendre tot el que pot passar en un joc de *Tchuma ruma* de 4 forats i 2 llavors, però també se sol jugar amb 6 llavors en comptes de 2 en cada forat. En aquest cas, hi hauria altres resultats possibles, i ens podríem plantejar si es pot guanyar el joc, i si en cas afirmatiu hi hagués només una sola ruta a la victòria, com el cas que hem analitzat, o més d'una.

Però això ens pot portar a pensar en totes les altres combinacions de forats i llavors que permet el joc. Per exemple, es pot guanyar un joc amb 5 forats i 4 llavors a cada forat? Els matemàtics P.J. Campbell i D.P. Chavey van tenir aquesta mateixa curiositat, i l'any 1995 van realitzar un estudi sobre aquest joc, considerant totes les combinacions de forats i llavors (Ibáñez, 2015). Dit d'una altra manera, van escriure uns teoremes generalitzant el joc *Tchuka ruma*. Un d'aquests teoremes assignava al nombre de forats sense contar la *ruma* la lletra n i al nombre de llavors per forat la lletra k , i deia:

Teorema: Per una $n \geq 2$ el joc de *Tchuka ruma* no té solució en els casos on $k = (n + 1)^i$, si $i \geq 1$ i on $k = n(n + 1)^i$, si $i \geq 0$.

Per entendre la base del teorema podem recolzar-nos en aquesta la figura 10 on es mostra el nombre de llavors que pots dipositar a la *ruma* donats un cert nombre de forats i llavors.

La primera afirmació del teorema diu "*Tchuka ruma* no té solució en els casos on $k = (n + 1)^i$, si $i \geq 1$ ". Si ens fixem en els casos on $i = 1$ trobem l'equació $k = n + 1$ i podem veure clarament perquè no es pot guanyar amb aquestes combinacions. Si el nombre de llavors és una unitat superior al nombre de forats, o sigui igual al nombre de forats contant la *ruma*, qualsevol forat que s'esculli repartirà l'última llavor en aquest mateix, que com que estarà buit, el jugador perdrà. També

¹⁸ Cada estat del taulell es representa amb 5 números, un per cada forat, i on el de la dreta representa la *ruma*. El nombre amb negreta representa la casella seleccionada pel jugador.

Semillas por hoyo / hoyos	1	2	3	4	5	6	7	8	9	10
1	1	1	3	3	4	6	7	7	8	9
2	1	2	1	8	10	7	14	16	4	20
3	3	4	2	1	15	17	21	24	27	30
4	3	8	7	2	1	24	28	32	36	40
5	3	7	14	6	2	1	35	40	45	50
6	3	10	15	24	30	2	1	48	54	60
7	5	14	19	27	35	13	2	1	63	70
8	5	12	24	32	40	48	9	2	1	80
9	5	15	27	36	45	54	63	72	2	1
10	5	17	30	40	50	60	70	80	90	2
11	7	19	33	44	55	66	77	88	99	110
12	7	19	36	48	60	72	84	96	108	120

FIGURA 10. Nombre de llavors que es poden emmagatzemar a la *ruma* per un nombre determinat de forats i llavors. Els casos en taronja els jocs que tenen solució. FONT. article de Cultura Científica - Tchuka Ruma, el mancala solitario

podem veure que només quedarà una llavor a la *ruma* tal com es pot veure en la figura 10¹⁹.

En els casos de $i > 1$ l'efecte és semblant però una mica més complex. Si ens fixem en l'exemple on $n = 1$ i $i = 2$ trobem que $k = 4$. La seqüència del joc aniria de la següent manera.

[4 0] → [2 2] → [1 3]

En el mateix exemple, però ara amb $i = 3$, tenim:

[8 0] → [4 0] → [2 6] → [1 7]

Podem observar que cada vegada l'última llavor es queda en el mateix forat, però realitzant un nombre de voltes igual a $k/(n + 1)$. D'aquesta manera el procés es torna a repetir. Podem veure que les llavors del forat escollit equivalen a les potències de $n + 1$ o el nombre total de forats contant la *ruma*. El nombre total de llavors que poden quedar a la *ruma* es pot trobar amb aquesta expressió:

$$(n + 1)^{i-1} + (n + 1)^{i-2} + \dots + (n + 1) + 1 = [(n + 1)^i - 1]/(n - 1)$$

¹⁹ S'observa un patró en diagonal començant a la casella $k=1, n=2$ fins a la $k=9, n=10$.

En el cas de la segona igualtat quan $i = 0$, podem afirmar que $k = n$ i en aquests casos tenim un estat que qualsevol moviment inicial, ens portarà a un dels casos que s'han explicat prèviament. Un exemple és un *Tchuka ruma* amb 4 forats i 4 llavors. A continuació hi ha les quatre primeres maneres de començar:

[4 4 4 4 0] → [0 5 5 5 1] →
 [4 4 4 4 0] → [5 0 5 5 1] →
 [4 4 4 4 0] → [5 5 0 5 1] →
 [4 4 4 4 0] → [5 5 5 0 1] →

En qualsevol dels quatre casos, els forats que es poden triar a continuació tenen $n + 1$ llavors forçant el jugador a la derrota.

Per últim, en els casos de $i \geq 1$, el funcionament és semblant: en la primera sembra es fa un nombre de voltes al taulell igual a $(n + 1)^{i-1}$ per acabar deixant a la mateixa casella $n(n + 1)^{i-1}$. Aquest procediment es repeteix fins que en el forat quedin només n llavors, i a continuació l'última llavor es dipositarà a la casella anterior (menys en el cas de l'última casella) que tindrà un nombre de llavors igual al nombre de llavors inicial més les afegides per les altres sembres, o sigui:

$$n(n + 1)^i + n(n + 1)^{i-1} + \dots + n(n + 1)^1 + n + 1 = (n + 1)^{i+1}$$

Fórmula que, com l'anterior, es trobava a partir de la successió geomètrica²⁰ $1 + \dots + r^{s-1} = (r^s - 1)/(r - 1)$. Si ens fixem en el nombre de llavors en aquesta casella, podem veure que compleix la primera condició del teorema, i per tant el jugador perdrà.

L'estudi de Campbell i Chavey també va trobar altres relacions que en cas de ser complides el jugador perdia, però per trobar relacions victorioses van haver de fer-ho utilitzant la computació. D'aquesta manera van poder obtenir una taula com l'anterior amb els diferents resultats de cada combinació.

2.1.2. Torre de Hanoi

La Torre de Hanoi és un altre clàssic trencaclosques comercialitzat per primera vegada a França, l'any 1883. Aquest trencaclosques va ser inventat per Édouard Lucas, qui també va escriure una petita llegenda i va crear un context inventat per al joc dient que el professor N. Claus era la persona que havia descobert el joc. La

²⁰ Successions de creixement més ràpid que les successions aritmètiques i que cada un dels elements d'aquestes successions tenen un quocient en comú, anomenat raó.

llegenda que acompanyava el joc parlava d'un temple a Benarés (Índia). Sota la cúpula que marcava el centre del món, hi havia una safata metàl·lica amb tres agulles de diamant. Déu, quan va crear el món, deia, que va col·locar 64 discos d'or en una de les agulles ordenats de manera que el més gran estava a sota i el més petit a dalt. Segons les lleis de Brahma, els monjos del temple eren els encarregats de traslladar aquests 64 discos de la primera agulla a la tercera sense descans amb unes normes: els monjos només podien moure un disc a la vegada, i mai un disc amb radi més gran podria estar a sobre d'un altre amb un radi més petit. Un cop els monjos haguessin traslladat tots els 64 discos a la tercera agulla el món s'acabaria.

Aquest joc per tant, està format per 3 petites columnes en fila, i amb 8 discos de diferent diàmetre (a diferència dels 64 de la història per una raó que s'explicarà més endavant). Aquests discos es col·locaven en l'ordre esmentat a la primera columna i l'objectiu del joc era passar-los a la tercera, seguint les normes del joc, i amb el mínim nombre de moviments.

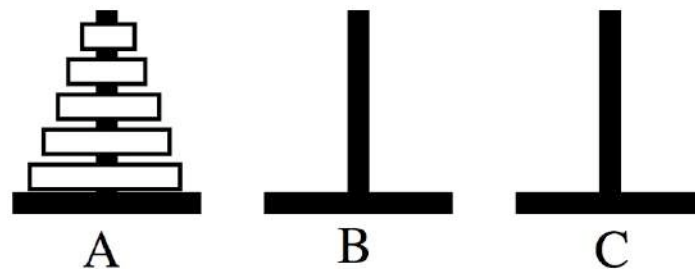


FIGURA 11. Esquema del joc de les Torres de Hanoi amb només 5 discs. FONT. *Flatiron School - Tracing Recursion Through the Towers of Hanoi Problem.*

Des d'un punt de vista matemàtic, aquest trencaclosques és perfecte per parlar de la recursivitat, ja que utilitzant aquest procés es pot resoldre el joc de la manera més senzilla. Un procés recursiu és "aquell que o bé es pot repetir teòricament un nombre infinit de vegades mitjançant l'aplicació d'una mateixa regla o que es crida a si mateix" (DIEC). D'una manera senzilla, un procés que sigui recursiu vol dir que funciona utilitzant una sola norma operació o funció que es repeteix d'una manera determinada. Un dels exemples més utilitzats és el càlcul de factorials²¹ a través de la recursivitat.

Si la funció recursiva es diu $f(x)$, direm que si $x = 0$ la funció retorni el valor 1 i acabi la instrucció. Si no el valor serà igual a $x \times f(x - 1)$. Aquest petit algorisme és capaç de retornar qualsevol factorial utilitzant només una mateixa instrucció que es crida unes quantes vegades.

²¹ Operació que equival a la multiplicació de tots els nombres començant pel número 1 fins a un de determinat. Se simbolitza amb el signe "!" ex: $4! = 1 \times 2 \times 3 \times 4 = 24$

Segons el nostre algorisme, el factorial de 3 es calcularia de la següent manera:

$$f(3) = 3 \times f(2) = 3 \times 2 \times f(1) = 3 \times 2 \times 1 = 6$$

Per tant podem aprofitar aquests processos per resoldre aquest trencaclosques. Per fer-ho també hem de plantejar un algorisme o sèrie d'instruccions que seran utilitzades en el procés. De manera general, aquestes regles serien les següents, utilitzant n per representar el nombre de discs inicials:

1. Si només hi ha un disc ($n = 1$) es mou aquest disc de la columna A fins a la C i s'acaba la seqüència d'instruccions.
2. Si hi ha més d'un disc ($n > 1$) es mouen tots els discs $1/2/.../n - 1$ de la columna A fins a l'auxiliar) utilitzant la columna C de suport.
3. Moure el disc n de la columna A fins a la C.
4. Moure els discs $1/2/.../n - 1$ de la columna B fins a la C, utilitzant la A de suport.

Si apliquem aquestes instruccions en el cas de $n = 3$, haurem de seguir aquests els següents passos on per fer-los més clars el conjunt de les quatre instruccions es representa com la funció $f(x, y, z)$ on x és igual al nombre de discs que s'han de moure, y la seva columna inicial i z la columna de destí:

$$\begin{aligned} f(3,1,3) &= f(2,1,2) + f(1,1,3) + f(2,2,3) \\ &= f(1,1,3) + f(1,1,2) + f(1,3,2) + f(1,1,3) + f(1,2,1) + f(1,2,3) + f(1,1,3) \end{aligned}$$

La recursivitat no només ens permet trobars els moviments necessaris d'una manera senzilla, sinó que també ens pot dir la quantitat mínima de moviments per resoldre un trencaclosques amb n discs. Sabem que si $f(n)$ es resol amb m moviments, $f(n + 1)$ es resoldrà amb $2m + 1$. Amb l'algorisme esmentat, primer es mourien els discs d'1 a $n - 1$ a la columna auxiliar (m moviments), després es mouria el disc n (1 moviment) i finalment es mourien la resta de discs de la columna auxiliar a la final (m moviments).

Per tant si sabem que $f(1)$ es resol amb un sol moviment, podem deduir els següents:

n	1	2	3	4	5	6
m	$0 + 1 + 0 = 1$	$1 + 1 + 1 = 3$	$3 + 1 + 3 = 7$	$7 + 1 + 7 = 15$	$15 + 1 + 15 = 31$	$31 + 1 + 31 = 63$

FIGURA 12. Nombre de moviments (m) per n discs.

Observant els nombres de moviments podem veure que equivalent l'expressió $2^n - 1$ quan hi ha n discs. Per tant els 64 discs de la llegenda s'haurien pogut moure amb 18.446.744.073.709.551.615 moviments, és a dir que si els monjos poguessin moure un disc en tan sols 1 segon, tardarien aproximadament 584 mil milions d'anys (Ibáñez, 2015).

2.2. Jocs de taula moderns i les matemàtiques

Els jocs de taula actuals són diferents respecte als jocs més clàssics o antics per diverses raons. Primer de tot podem veure que els dos jocs/trencaclosques que s'han introduït a l'apartat anterior no tenen atzar, es basen totalment en estratègia i en les decisions que pren el jugador. Per aquesta raó aquests jocs són previsibles, és a dir, que si jugues un joc de *Tchuka ruma* amb 4 forats i 2 llavors sempre tindràs les mateixes probabilitats de guanyar, i aconseguiràs la victòria sempre de la mateixa manera. Per ser més interessants, els jocs d'avui en dia eviten aquestes dinàmiques.

Per evitar jocs previsibles, la majoria de jocs actuals estan pensats per ser jugats en grup. El nombre de jugadors sol ser entre 2 i 4, però també es poden trobar jocs els quals es poden jugar individualment, com l'*Agricola*, amb 2 jugadors o més de 4. El fet d'afegir més jugadors no només aconsegueix convertir el trencaclosques en un joc, sinó que també afegeix atzar en el joc. Com s'explicarà en el pròxim apartat l'atzar no es troba només en daus, sinó que les interaccions entre jugadors també en poden generar. I justament l'atzar és un dels factors essencials dels jocs de taula moderns.

L'atzar, a més, està lligat amb la sort i amb l'estratègia i per tant els jocs buscaran equilibris per aconseguir certes dinàmiques dintre el joc. Per exemple un joc amb una gran presència d'atzar podria ser *Monopoly* o *Risk* on els daus són uns grans determinants del resultat de la partida. D'altres on l'atzar no té tanta influència sobre les estratègies del jugador poden ser jocs com *Agricola*. En el pròxim apartat també es parlarà perquè no existeixen jocs sense atzar.

Un altre aspecte que diferencia els jocs moderns dels jocs clàssics és la diplomàcia. El fet de negociar i debatre entre els jugadors durant la mateixa partida s'ha anat afegint en els jocs de taula com una dinàmica nova. Un dels jocs que va incitar més a aquestes relacions és *The Settlers of Catan* on es dona via lliure als jugadors per fer intercanvis dels seus materials.

Tot i això, els jocs de taula moderns no són tots iguals. Com es va comentar en l'apartat d'història i popularització dels jocs de taula, hi ha dues grans potències que desenvolupen jocs de taula, els jocs europeus o de dissenyador i els americans o de mercat. Aquests dos tipus de jocs presenten les següents diferències (Hillengas, 2015):

- Els jocs de dissenyador estan pensats i produïts per adults, i en canvi els de mercat estan destinats a famílies o nens.
- Els jocs de mercat tenen una major presència d'atzar i de sort que els jocs europeus. Per tant els jocs europeus donen més importància a l'estratègia i l'habilitat dels jugadors.
- Majoritàriament els jocs europeus tenen expansions, i en canvi els de mercat tenen variants. Per exemple *Monopoly* un joc americà i de mercat té diferents variants però no expansions com *7Wonders*, joc europeu que sí que en té.

2.3. Atzar i sort

2.3.1. Què entenem per atzar?

L'atzar és un dels grans aspectes dels jocs de taula que els relaciona més directament amb les matemàtiques. El terme sort és molt utilitzat quan juguem. Per exemple quan perdem sense saber-ne la causa, acusem el jugador que ha guanyat de tenir sort o d'haver guanyat per casualitat o per atzar. També s'utilitzen aquests termes per classificar jocs. Diem que un joc és de sort quan hi ha molt d'atzar o que un joc és d'estratègia quan l'atzar no té tanta importància. Un tema també molt debatut és si l'atzar i l'estratègia són contraris, o es complementen. Per contestar aquestes preguntes, però, s'ha d'entendre què significa atzar com a concepte abstracte.

L'atzar es defineix com a “causa assignada als fets dels quals ens escapa la causa real” (DIEC) o com a “*The quality or state of lacking a pattern or principle of organization; unpredictability.*”²² (Oxford). L'atzar, des d'un punt de vista matemàtic, és una manera de representar la incertesa (Gilbert, 2017). Una funció determinista vol dir que per a qualsevol valor d'entrada en pots saber el valor de sortida amb un 100% de certesa. En canvi amb una funció aleatòria o basada amb l'atzar, els resultats no es poden predir completament. Aquests resultats, en canvi, es representen amb distribucions i percentatges. Prioritzes certs valors per la probabilitat d'aquests de succeir. Això es pot exemplificar amb un cas físic com la tirada d'un dau. A causa de la nostra manca de coneixement exacte dels

²² La qualitat o estat de manca d'un patró o d'un principi d'organització; imprevisibilitat.

comportaments físics de cossos, els humans no podem predir un valor de sortida exacte, i en canvi donem a cada resultat un percentatge que expressa el nombre de casos favorables dividit pels casos totals. En el cas d'un dau de 6 cares cada resultat o nombre d'1 al 6 té una probabilitat del $16,6\%$ que succeeixi.

Per tant utilitzem l'atzar per simplificar esdeveniments en el món real que no podem predir a causa de la seva complexitat.

Però que un esdeveniment sigui aleatori no necessàriament indica que sigui uniformement aleatori. Els esdeveniments aleatoris tant poden ser uniformement aleatoris, com la tirada d'un sol dau, on la probabilitat que surti qualsevol resultat és igual a la resta, com parcialment aleatoris, com en el cas de tirar 2 daus i sumar els resultats. En aquest últim cas, les probabilitats que surti un 7 són molt més altes que el resultat sigui un 2 o un 12, ja que perquè el resultat sigui un 7 podem tenir diferents combinacions de daus com 1 i 6, 2 i 5 o 3 i 4, mentre que perquè surti un 2 els dos daus han de donar un 1. Per tant, $1/6$ de les vegades el resultat serà 7 i només $1/36$ serà 2 o 12. Aquesta diferència de probabilitats fa que sigui un esdeveniment parcialment aleatori, ja que posseïm informació del resultat que ens permet establir preferències. L'exemple de la tirada de 2 daus es pot observar en el joc *The Settlers of Catan* on l'obtenció de recursos es basa en aquest esdeveniment.

Tot i que aquests conceptes estan força establerts, utilitzem certes expressions que no són tan fàcil d'interpretar, com el terme sort o la utilització de les expressions "més o menys aleatori". La sort es defineix com a "*Success or failure apparently brought by chance rather than through one's own actions*"²³ (Oxford). Aparentment podríem dir que la definició d'atzar i de sort són pràcticament iguals, però com es comenta en el següent apartat no és exactament així.

Però encara que no ho sembli, l'atzar és present en tots els jocs de taula d'avui en dia. Aquesta afirmació pot semblar exagerada però després d'entendre els tipus d'atzar del següent apartat té molt de sentit.

2.3.2. Tipus d'atzar

Segons Richard Garfield (2012) l'atzar es pot dividir en 3 tipus. Aquesta divisió ajuda a entendre perquè tirar un dau o jugar una partida d'escacs poden tenir atzar tot i no tenir cap semblança clara.

²³ Esdeveniments d'èxit o de fracàs aparentment causat per casualitat i no per les accions d'un mateix.

Primer de tot tenim l'atzar de tipus 1, l'atzar més conegut i lògic. Aquesta aleatorietat es produeix a causa d'indeterminacions físiques, i de prediccions fora del nostre abast. Si el nostre món, segons un punt de vista determinista²⁴, pogués ser entès al 100% pels humans, aquest atzar desapareixeria, ja que es podria predir el resultat d'una tirada de dau, simplement calculant les forces físiques que intervindrien. Tot i això, actualment no podem fer aquest tipus de prediccions, i per tant tirades de daus, tirades de monedes o barreges de cartes són esdeveniments de resultat incert. Aquest atzar per tant és causat per les nostres limitacions com a éssers humans.

El segon tipus d'atzar, sorgeix en situacions on hi ha una informació amagada al jugador o s'han de fer decisions simultànies amb altres jugadors. Un possible cas d'informació amagada serien jocs amb rols secrets com *Bang!* o *Saboteur* on cada persona té un objectiu que no és conegut pels altres jugadors. Per tant, les decisions que prengui un jugador seran incertes per qualsevol altre jugador. En el cas de decisions simultànies passa el mateix, tots els jugadors han de prendre decisions pensant en el que faran els altres, creant així una indeterminació. No només hi ha aquesta aleatorietat, sinó que segons una anàlisi teòrica de jocs anomenats jocs de von Neumann²⁵, les millors estratègies en alguns d'aquests casos és una decisió aleatòria (no necessàriament una decisió uniformement aleatòria).

Aquests jocs de Von Neumann són jocs de decisions simultànies on l'èxit dels jugadors depèn de la decisió dels altres jugadors. Algun d'aquests jocs són el dilema del presoner²⁶ o el pedra, paper i tisores.

Per últim tenim el tipus d'atzar número 3 que és el més abstracte i que està al límit de ser considerat atzar. Aquest atzar és el responsable de les indeterminacions en les partides d'escacs, ja que en aquest joc no hi ha ni tipus 1 ni 2 d'atzar. Aquest atzar doncs sorgeix de les limitacions humanes a l'hora de crear estratègies.

Utilitzant la teoria de joc, els matemàtics poden realitzar les millors estratègies per a jocs. D'aquesta manera el jugador pot prendre les decisions òptimes per poder guanyar. Però qualsevol joc de taula que sigui interessant per jugar, serà prou

²⁴ Doctrina filosòfica segons la qual tots els fenòmens de l'univers, o una categoria d'aquests, depenen tan estretament dels que els precedeixen que per a cada moment només hi ha un resultat possible (DIEC).

²⁵ Jocs anomenats després de John von Neumann, importantíssim matemàtic especialitzat en teoria de joc.

²⁶ Dilema on dos jugadors (presoners) han de triar simultàniament una de les dos opcions possibles. Una de les opcions és confessar i l'altre callar. Ara bé, si els dos callen, se'ls condemna als dos a 6 mesos de presó, si els dos confessen, els dos són condemnats a 6 anys, i si només un dels dos confessa, aquest surt lliure, i l'altre es condemnat a 10 anys de presó.

complex perquè aquestes estratègies no es puguin calcular. Continuant amb l'exemple dels escacs, en comptes d'observar totes les combinacions de moviments, que seria una tasca inútil a causa de l'enorme quantitat de "branques", s'utilitzen heurístics. L'heurística és una forma de treball que no garanteix una solució òptima, però sí que en proporciona una de bona i en un curt temps. En el cas dels escacs es realitzen unes anàlisis que no observen totes les possibilitats però sí que són capaços de determinar si uns moviments seran millors que altres. Això funciona, i es pot veure amb la màquina *Deep Blue*²⁷ que va guanyar el millor jugador d'escacs l'any 1996, però tot i això pot arribar a causar decisions errònies, ja que segons algunes heurístiques dues decisions poden ser igual de bones, però pot acabar portant a la victòria i l'altre no. Això és perquè aquestes decisions no contemplen les seves repercussions a llarg termini.

Així doncs, el tercer tipus d'atzar és molt semblant al primer. Les indeterminacions causades per la manca d'informació a l'hora de formular estratègies són semblants a les indeterminacions físiques d'una tirada de daus.

Després d'haver definit aquests 3 tipus d'atzar es pot afirmar que en qualsevol joc de taula modern en trobarem algun. El fonament d'aquesta afirmació és que tots els jocs de taula tenen una certa complexitat perquè siguin atractius per jugar, creant així el tercer tipus d'atzar. D'aquesta manera, i afegint els altres dos tipus d'atzar, podem dir que qualsevol joc de taula conté i necessita l'atzar.

El concepte de sort però, l'utilitzem d'una manera diferent. Si l'atzar es pot manifestar en aquests 3 tipus comentats, que algú tingui sort majoritàriament fa referència al primer tipus d'atzar. Una persona afortunada és la que obté el resultat escollit quan es tira un dau, o escull la carta que volia d'una baralla barrejada. En canvi no se sol considerar sort que algú guanyi per alguna decisió simultània o per algun cas del tercer tipus. Tot i que seria possible relacionar tots els conceptes d'atzar amb la sort, no és usual. Per aquesta raó en aquest treball es fa més referència a atzar o aleatorietat, ja que és un terme més definit i més clar, tot i ser una qualitat abstracta.

2.3.3. L'atzar limita l'estratègia?

Una de les afirmacions més comunes sobre l'atzar és que l'atzar és el contrari que l'estratègia i que per tant quan més atzar hi ha en un joc menys estratègia. Aquesta afirmació no és totalment correcta: l'atzar sí que pot tenir una relació inversament proporcional amb l'estratègia en jocs on hi ha una gran presència d'atzar perquè els

²⁷ Primer ordinador destinat a jugar a escacs que va ser capaç de guanyar el millor jugador d'escacs del moment en una competició i en una partida individual (1997).

jugadors no poden desenvolupar estratègies gaire complexes. Però els dos termes no són contraris. L'atzar ajuda a crear complexitat, que alhora permet que els jugadors puguin dedicar més temps a l'estratègia. Per tant, algun tipus d'atzar amb excés com el de tipus 1, pot fer que un joc tingui menys presència d'estratègia, però la majoria d'atzar, ja sigui en petites quantitats de tipus 1 o amb altres dosis de tipus 2 i 3 inciten a crear estratègies més complexes i no a limitar el jugador de fer-ne. Resumint, no només hi ha jocs amb molta presència d'atzar i poca d'estratègia o poca d'atzar i molta d'estratègia, sinó que també existeixen jocs amb poca atzar i poca estratègia o molta presència de les dues coses.

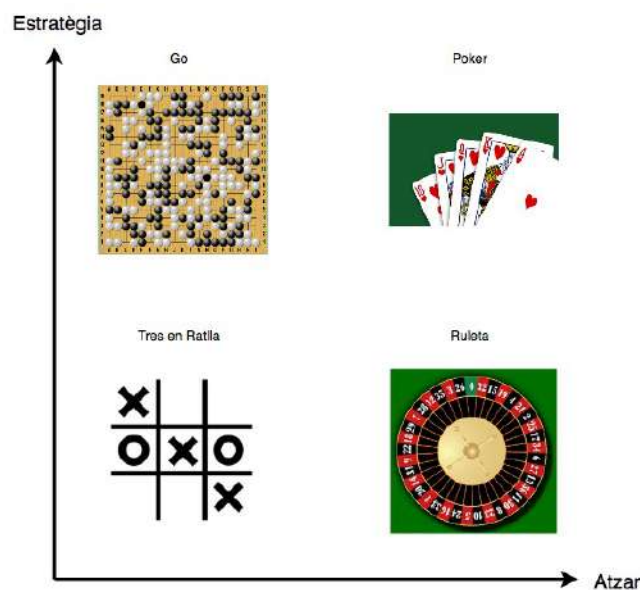


FIGURA 13. Quatre jocs classificats per la seva proporció d'atzar i estratègia. FONT. Elaboració pròpia mitjançant *draw.io*: plataforma oberta per crear i compartir diagrames.

Tot depèn de la influència de l'atzar en el joc. Per exemple en el cas de la variant d'escacs *Rando-Chess*, si un jugador guanya la partida d'escacs però perd amb l'última tirada de daus, aquest atzar no té cap valor en el joc perquè no canvia la manera de jugar d'un jugador i no afegeix complexitat al joc, simplement fa que una vegada entre sis, el jugador que guanya la partida acabarà perdent d'una manera injusta. En canvi atzar com les tirades de daus en *The Settlers of Catan*, utilitzant el mateix tipus d'atzar, influeixen en com els jugadors juguen i els obliga a posicionar-se en caselles on l'obtenció de materials és més eficient.

Un atzar amb una l'objectiu de crear complexitat i amb poca influència en l'habilitat i l'estratègia dels jugadors és un atzar que afegeix valor al joc. En canvi un atzar que té una funció selectiva, no lligada a les decisions prèvies del jugador o amb una gran influència al resultat final de la partida no aporta cap valor al joc. És un atzar que es podria classificar com a injust.

A part d'afegir complexitat, l'atzar també pot tenir la funció de crear varietat. Si tornem al joc dels escacs, exemple perfecte d'un joc amb molta estratègia i molt poc atzar, trobem que a mesura que un jugador juga a aquest joc i s'hi torna expert, utilitzarà cada vegada més estratègies estudiades prèviament, com obertures o finals de partida. Això pot comportar que el veritable joc comenci 20 torns després que hagi començat el joc. Per evitar aquests inicis idèntics o molt semblants, Bobby Fischer²⁸ va crear una versió del joc clàssic on, a part de la fila dels peons, totes les altres peces es desordenen conservant una simetria amb l'altre jugador. D'aquesta manera es poden crear 960²⁹ inicis diferents i no es poden utilitzar obertures o moviments inicials o finals prèviament estudiats.

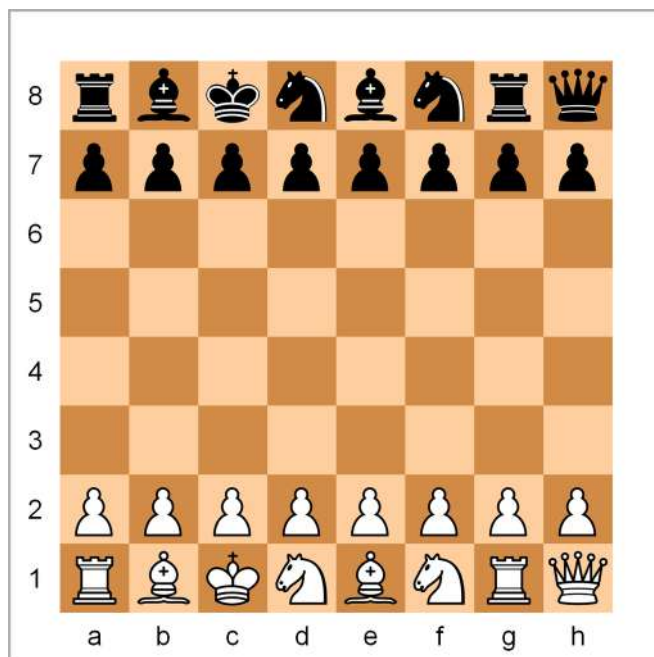


FIGURA 14. Un tauler dels escenaris d'escacs aleatoris de Fischer. FONT. Article de *Wikipedia* - Escacs aleatoris de Fischer.

Per últim l'atzar també pot tenir la funció de limitar l'estratègia, ja que pot ajudar a encarar un joc a un cert públic. Si es vol fer un joc destinat a la família o per a nens, el joc no pot ser molt complicat, i una manera de reduir aquesta complexitat és afegir més atzar per acabar eliminant part de l'estratègia del joc. Algun exemple pot ser *El Hobbit*, joc que té els 3 tipus d'atzars: el joc es divideix en 4 fases i en cada fase hi ha dues parts. En la primera els jugadors es troben amb decisions simultànies (tipus 2), en la segona part hi ha una gran importància dels daus (tipus 1), i a causa de la

²⁸ Jugador d'escacs professional i campió del món entre 1972 i 1975.

²⁹ Seguint les normes que els dos alfils han d'estar en caselles de colors diferents i que el rei ha d'estar en una casella entre les dues torres. L'operació final per tant és $(4 \times 4 \times 6 \times 5 \times 4 \times 1) / 2 = 960$ on es multipliquen les possibles posicions dels dos alfils, de la reina, dels dos cavalls i per últim es col·loquen les torres i el rei de l'única manera restant. Es divideix el resultat en dos per evitar simetries horitzontals.

combinació d'aquests dos el joc és prou complex perquè els jugadors no puguin fer estratègies òptimes, creant així l'atzar de tipus 3. A causa de la gran presència d'atzar aquest joc està recomanat per a persones de més de 8 anys (edat relativament baixa) i segons els usuaris de *BoardGameGeek* la complexitat del joc és de 1.71 sobre 5.

Resumint, l'atzar pot afavorir la creació d'estratègies, sigui afegint complexitat al joc o creant variants, però també pot fer el contrari i disminuir la dificultat del joc. Si s'ajusta l'atzar correctament, on aquest interactua amb el jugador en comptes d'ignorar-lo, aquest pot aportar un aspecte positiu al joc.

3. Creació d'una pàgina web

3.1. Objectiu

Aquest projecte de creació d'una pàgina web té un objectiu clar: crear una plataforma virtual, disponible mitjançant Internet i que permeti donar informació als jugadors de jocs de taula sobre les seves partides. Aquesta plataforma, per tant, ha de ser capaç de guardar dades introduïdes per diversos usuaris, i capaç de processar-les i tornar-ne resultats en forma d'estadístiques o gràfiques. Per exemple, un resultat obtingut de la web podria ser el percentatge de victòries d'un jugador en un joc concret. Per tant la intenció d'aquesta web és d'esdevenir una espècie d'eina per a jugadors, on puguin controlar els seus resultats i tendències a l'hora de jugar a jocs de taula, o simplement per aportar una sèrie de dades curioses i interessants.

Aquesta plataforma s'hauria pogut plantejar i realitzar en altres formats, com per exemple en forma d'aplicació per mòbil, però es va decidir realitzar el projecte en format web. Això és a causa del fet que, en el cas de les apps mòbils, hi ha moltes restriccions i problemes de compatibilitat a l'hora d'utilitzar-les amb diferents dispositius. Per exemple, en els sistemes operatius IOS de l'empresa *Apple*, l'aplicació s'hauria de publicar a l'*Apple Store*, però només es podria fer servir des d'altres dispositius IOS i per tant des de sistemes operatius com *Android* no s'hi podria accedir. Fer més d'una versió de l'aplicació tampoc valia la pena, ja que s'haurien de programar en diferents llenguatges de programació, cosa que multiplicaria la feina considerablement.

Per tant el format de pàgina web era l'opció més viable i eficient. Tot i això, molts dels llenguatges de programació eren nous per nosaltres, i per tant hi va haver una feina d'aprenentatge important.

3.2. Procediment i desenvolupament

Aquest projecte es va començar sense tenir cap coneixement previ dels codis de programació de pàgines web, llenguatges que serien necessaris perquè tot el web s'escriuria directament en codi. També era necessari comprendre l'estructura general de les pàgines web per poder pensar i construir totes les parts. Amb aquest punt de partida, el primer pas del projecte va ser l'aprenentatge d'un dels codis de programació bàsics: HTML³⁰. Aquest aprenentatge es va fer de manera autodidàctica utilitzant el curs en línia del web *KhanAcademy*. A part de HTML,

³⁰ Consultar apartat 3.3.3.1.

també es varen obtenir els coneixements bàsics de l'ús d'estils en pàgines web, que més endavant s'aplicarien amb CSS³¹ al web final.

Una vegada acabat el curs, es va desenvolupar la primera versió del web. L'objectiu d'aquesta primera versió era poder aplicar els coneixements apresos de SQL³² i de PHP³³ (dos altres llenguatges de programació necessaris), i alhora aplicar els fonaments d'HTML. Tot això amb la creació de programes que es poguessin utilitzar en el moment de fer la segona versió. La recerca d'informació i del funcionament de SQL també es va realitzar utilitzant altres plataformes educatives en línia com *StackOverflow* o *w3schools*. L'estructura d'aquesta versió del web consistia en una sèrie de fitxers programats en HTML i PHP i en una base de dades mitjançant un *localhost*³⁴, semblant a l'estructura de la versió final del web, que s'explicarà en els pròxims apartats, però simplificada.







	Marca	Imatge	Descripció
editeix	calborne		Hans im Glück: Verlag created by a German board and card game publisher. Though many of their own games are language-independent they nevertheless publish only coverage for the German market which include only German-language rules. English language editions of their games have been published primarily by Rio Grande Games, Dutch versions by 999 Games and so on. From: https://en.wikipedia.org/wiki/Hans_im_Gluek
editeix	calborne		Founded in Brazil in 1987, the Devir group ranked its products more in the imaginative spectrum of fiction and fantasy, offering products that involve wit, knowledge, wit, and social interaction. Beyond education, culture, and fun... these products keep the strength of Brazilian clubs and alive in all of us. From: http://devir.com
editeix	calborne		Gigamic was founded in 1991 by 3 brothers. Settled in Wenzhou, North of France, the company began by producing the famous abstract game Quarto, which has sold over 1 million units worldwide. In 2016, Gigamic and Quarto celebrated their 25th anniversary together! From: http://www.gigamic.com/fr
editeix	calborne		Edge Entertainment es una editorial chilena dedicada a la publicación de entretenimiento de juegos y libros que desafia la realidad. Fundada en 1999, la labor editorial de Edge comenzó con la publicación de Harry Potter, un innovador juego de rol, también acompañado de su propio juego de misterios, el que jugamos ahora mismo como Zombis, WitchCraft o 18keynes Armas. From: http://www.edgeent.com/boocio/boocio.php
editeix	calborne		
editeix	calborne		

FIGURA 15. Pàgina de la primera versió del web. FONT. Elaboració pròpia.

Els programes del primer web tenien les funcions de poder afegir, editar, esborrar i mostrar globalment marques de jocs, jocs i jugadors. A causa de la manca d'estils, la pàgina web era molt simple i no tenia una estructura, però no era important perquè la funcionalitat no era la seva funció.

Un cop es va crear aquesta primera versió, i amb un cert domini dels tres llenguatges de programació claus pel web, es va passar a fer la versió final del web.

³¹ Consultar apartat 3.3.3.2.

³² Consultar apartat 3.3.3.4.

³³ Consultar apartat 3.3.3.3.

³⁴ Nom que significa "aquest ordinador" utilitzat per executar pàgines web emmagatzemades en el mateix ordinador on s'ha realitzat la consulta.

Per fer-lo, es va aconseguir un subdomini web, una base de dades i un servei d'FTP³⁵, per tal de poder fer aquest web accessible des d'Internet, emmagatzemar dades en línia i poder transferir fitxers a un servidor³⁶. A més, es va partir d'una plantilla de pàgina web (de la pàgina *html5up*) per tenir uns estils editables de partida.

Durant aquest procés d'aconseguir un web accessible des d'Internet, es varen utilitzar diferents programes informàtics. Primer de tot, per poder utilitzar el servei FTP, el programa *FileZilla*, un programa estàndard que permet transferir fitxers des d'un ordinador a un servidor. Un altre programa important és l'editor de codi *Geany*, que ja s'utilitzava en la primera versió del web, però que és important perquè és amb el qual es va escriure la totalitat del web. Per últim, també es va utilitzar la plataforma *phpMyAdmin*, una plataforma accessible des d'Internet i que permet configurar i editar una base de dades. Per tant, aquest programa també va ser essencial per a la construcció del web, ja que la base de dades n'és una part important.

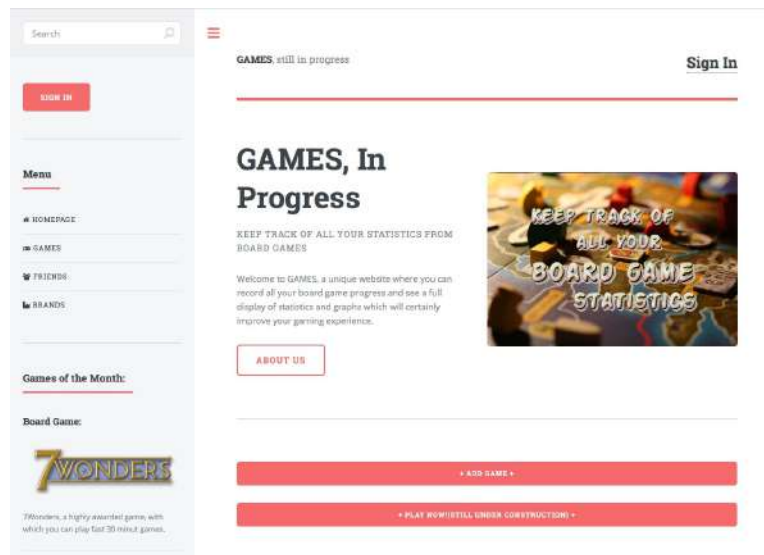


FIGURA 16. Pàgina principal de la versió final del web. FONT. Elaboració pròpia.

Els fitxers de text, que equivalen a cada pàgina del web, es van crear utilitzant l'editor de codi *Geany*, posteriorment es penjaven en un espai web de prova, utilitzant el *localhost* que s'havia utilitzat també en la primera web, i després de corregir els possibles errors es transferia el fitxer mitjançant *FileZilla* al servidor. D'aquesta manera, el programa estava disponible des del domini. La base de dades es treballava per separat i s'editava directament des de *PhpMyAdmin*.

³⁵ Protocol de transferència de fitxers

³⁶ Maquinària informàtica amb l'objectiu de rebre consultes d'altres ordinadors i respondre-les. La interconnexió d'aquests i altres components informàtics forma la WWW.

3.3. Estructura general

La pàgina web es divideix en dos grans parts: una base de dades, i uns fitxers emmagatzemats en un domini virtual³⁷. La base de dades serveix d'emmagatzematge per totes les dades entrades a la web d'una manera específica perquè els diferents fitxers de codi puguin manipular-la i presentar-la.

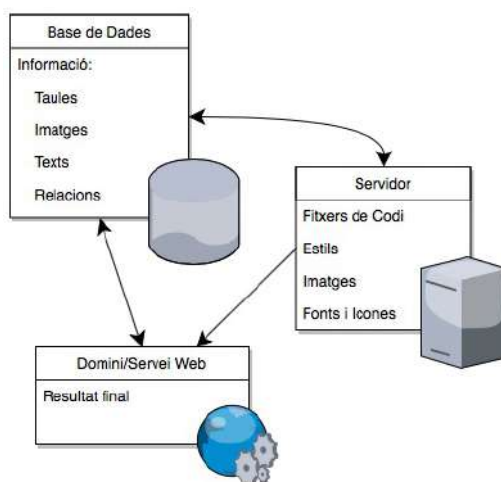


FIGURA 17. Esquema del funcionament de la versió final del web. FONT. Elaboració pròpia mitjançant *draw.io*: plataforma oberta per crear i compartir diagrames.

3.3.1. Fonaments

Tant les consultes de la base de dades com els fitxers de codi estan programats utilitzant 5 llenguatges de programació. Cada un d'aquest té una funció específica i essencial en el web.

3.3.1.1. HTML

Un dels llenguatges de programació utilitzats en el projecte, i el més important de cara a la WWW³⁸ és HTML (llenguatge de marcat d'hipertext), el primer llenguatge de programació destinat a l'estructuració i disseny de les pàgines web que es va crear mai. Aquest va ser primer plantejat per Tim Berners-Lee l'any 1991, però se n'han fet diverses versions que l'adapten a les noves tecnologies. Actualment la versió que més s'utilitza, i que també s'utilitza en aquesta web, és HTML5, creada l'any 2006.

³⁷ Nom únic que fa referència a un espai web i que correspon a una adreça IP. Està relacionat amb un URL també.

³⁸ Sigles que en anglès signifiquen *World Wide Web* (teranyina d'abast mundial) o web.

Aquest llenguatge de programació està molt lligat amb la WWW i la seva creació va ajudar l'expansió i el desenvolupament de la web en si. Si pensem que la web funciona com a sistema de transferència de dades via una xarxa d'ordinadors connectats entre si, podem veure que les principals dades que es transfereixen són pàgines web. Quan un usuari des del seu ordinador connectat a la xarxa accedeix a un URL³⁹ o a una IP⁴⁰ en concret, el que està fent, i de manera simplificada, és demanar una pàgina web lligada a l'URL a l'ordinador amb aquesta IP. L'ordinador després de rebre la demanda, envia un petit programa que l'ordinador de l'usuari pot obrir en forma de pàgina web. Aquest programa estava i està escrit amb HTML, el qual pot mostrar tant text com imatges, vídeos, jocs i altres. També, com que aquest llenguatge ha sigut l'estàndard en el seu àmbit des del principi, ara totes les plataformes l'accepten, i per tant és compatible en qualsevol mitjà electrònic que tingui connexió a la xarxa, des de telèfons mòbils a ordinadors i televisors intel·ligents. El mateix s'aplica en els enginyers de cerca de la web, com poden ser *Chrome*, *Safari*, o *Mozilla Firefox*. Tots suporten i estan creats al voltant de HTML.

Aquest llenguatge és un llenguatge de text caracteritzat en la diferenciació i la divisió en seccions d'un programa. Cada una d'aquestes seccions o etiquetes, té un inici i final marcat, caracteritzats per la mateixa simbologia estàndard, utilitzant els símbols "<", ">" i "/". Per tant, l'inici d'una secció es marcaria amb el text *<etiqueta>* i per acabar la secció es marcaria amb *</etiqueta>* on la paraula etiqueta, utilitzada en aquest exemple se substituiria pel nom de la secció. Com es pot veure en la figura 18, un programa d'HTML es comença sempre amb les etiquetes *<html></html>*, indicant així l'inici i el final del programa. Això és molt important a causa de la varietat de llenguatges de programació que es poden acabar fent servir en un mateix programa per a webs, com es podrà veure en els apartats de *JavaScript* i *PHP*. Seguidament s'utilitzen les etiquetes *<head></head>* i *<body></body>* dividint i diferenciant el programa amb dues parts importants. La part *head* fa referència a la capçalera del

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Exmeple HTML</title>
    <meta charset="utf-8" />
    <meta name="viewport" />
    <style>
      img{height:100px;}
      table,th,td{border: 1px solid;}
    </style>
  </head>
  <body>
    <h1>Títol 1</h1>
    <p>Text en format de paràgraf amb un <a
href="#">enllaç</a> de prova.</p>
    <p></p>
    <h3>Títol més petit per la foto</h3>
    
    <p></p>
    <table>
      <thead>
        <tr><th>objecte1</th>
      </thead>
      <tbody>
        <tr><td>caract. 1</td>
      </tbody>
    </table>
  </body>
</html>
```

FIGURA 18. Codi en HTML. FONT. Elaboració pròpia mitjançant *Geany - text editor*.

³⁹ Localitzador uniforme de recursos.

⁴⁰ L'adreça del protocol d'Internet (IP) és una adreça única que adopten els ordinadors quan es connecten a Internet. Pot ser fixa o dinàmica.

programa, amb informació sobre el programa i l'usuari, com per exemple el nom del web. Després tenim *body*, on trobem el cos de la web que conté el contingut. A partir d'aquí, l'estructura és més lliure i varia segons el resultat que es vol aconseguir. Però hi ha una sèrie d'etiquetes essencials que són les que s'utilitzen més freqüentment: per exemple, tenim `` per mostrar imatges, `<p>` per mostrar text en un paràgraf, `<h#>` per representar títols (on el nombre que acompanya *h* és el seu nivell d'importància essent 1 el més important), `<table>` per crear una taula i `<a>` per crear enllaços o *hiperlinks* en un cert contingut. La llista d'etiquetes no s'acaba, però cada una té una funció clara i ben definida.

Les etiquetes però, també tenen altres funcions. Com es pot veure en la figura 19, en l'etiqueta d'inici de la secció hi tenim més informació. Aquesta informació normalment indica estils en forma d'atributs com per exemple colors o mides del contingut de les etiquetes. Però un dels usos principals d'aquesta informació és la classificació de seccions segons *id* o *classe*. Per exemple, si tenim 3 seccions amb etiquetes `<p>`, i ens volem indicar a l'ordinador només a la del mig, no ho podem fer. Per tant, la solució és establir un *id* únic per la secció `<p>` del mig, i quan vulguem referir-nos a aquesta secció, només haurem d'indicar l'*id* del que estem parlant, i l'ordinador podrà trobar la secció indicada. Com es pot intuir, aquesta propietat és molt útil, sobretot en programes llargs i complexos, però també tenim una segona opció que és la *classe*. Si l'*id* era un element únic, les classes les podem fer servir sense límit.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Exmple 2 HTML</title>
    <meta charset="utf-8" />
    <meta name="viewport" />
    <style>
      .estil1 {
        color:Red;
      }
      .estil2 {
        color: Green;
      }
    </style>
  </head>
  <body>
    <h1>Prova <i>id</i></h1>
    <p class="estil2">Primer paràgraf</p>
    <p class="estil1">Segon paràgraf</p>
    <p class="estil1">Tercer paràgraf</p>
    <p class="estil1">Quart paràgraf</p>
    <p class="estil1">Penúltim paràgraf</p>
    <p class="estil2">Últim paràgraf</p>
  </body>
</html>
```

FIGURA 19. Codi en HTML. FONT. Elaboració pròpia mitjançant *Geany - text editor*.

Per exemple en la figura 19 tenim uns estils de text, delimitats per les etiquetes `<style></style>` que volem aplicar als diferents paràgrafs de la pàgina web. Volem que només el primer i últim paràgraf tinguin el text de color vermell, i que la resta el tinguin de color verd. En comptes d'especificar-ho en cada etiqueta `<p>`, opció possible però poc eficient, ja que en cas d'error s'haurien de modificar sis instruccions diferents, es defineixen dues classes: un pel text vermell i un pel text verd. I a cada paràgraf també s'indica la classe que es vol fer servir. En cas de voler canviar de color, només s'hauria de canviar una línia, i totes les seccions amb la classe canviada s'adaptarien al canvi. L'ús dels *id* i de les classes va molt lligat amb el codi de programació CSS, com es podrà veure a continuació.

3.3.1.2. CSS

CSS és el llenguatge de programació encarregat dels estils i és el que ajuda a complementar programes d'HTML, d'una manera senzilla. CSS fa referència a *Cascading Style Sheets* o fulls d'estils en cascada i és un codi molt utilitzat en pàgines web. Té una sintaxi simple i una funció molt específica de guardar i estructurar estils. Aquests estils es guarden en documents a part, i el programa principal en HTML o amb algun altre llenguatge estàndard d'estructuració de webs el crida per poder aplicar aquests estils.

L'avantatge més gran de l'ús de CSS és la facilitat d'editar estils en un únic document. En cas de no utilitzar-lo, s'haurien de mencionar els estils en cada un dels fitxers en HTML, dintre de les etiquetes `<style>` cosa que suposaria molta més feina a l'hora d'editar-los.

L'estructura, com s'ha esmentat en l'últim apartat, utilitza els *id*, les classes i les etiquetes del programa d'HTML. Com es pot veure en la figura 20, per assignar uns estils a un *id* en concret, primer es cita l'*id* amb el símbol `#` al davant i s'obren uns claudàtors. Tots els estils dintre d'aquests claudàtors s'aplicaran a l'*id* al qual fan referència. A part de l'*id*, també es poden fer servir classes o etiquetes com `<p>` per aplicar-hi estils. També es poden aplicar estils a seccions dintre altres seccions i també fer referència a seccions concretes d'una sèrie d'aquestes utilitzant certes instruccions que, per exemple, et permeten triar la primera secció de la sèrie.

```
#sidebar {
  color: #aaa;
  color: rgba(255, 255, 255, 0.55);
  width: 14em;
  padding: 0.75em 1.35em 1em 0em;
  background: rgb(255,255,255);
  overflow-y: scroll;
  -webkit-overflow-scrolling:
touch;
  left: 0;
  box-shadow: inset -0.1em 0em
0.35em 0em rgba(0, 0, 0, 0.15);
  opacity: 1.5;

  z-index: 0;
}
#sidebar a {
  color: #000;
}
#sidebar p{
  font-size: 0.8em;
  color: #97999A;
}
#sidebar p > a{
  text-decoration: none;
  color:#97999A;
}
#sidebar p > a:first-child{
  padding: 0 0 0 1.55em;
}
```

FIGURA 20. Codi en CSS. FONT. Elaboració pròpia mitjançant *Geany - text editor*.

Els estils que es poden definir utilitzant CSS els podem dividir en diferents grups:

- **Estils de posició:** aquests tipus d'estils defineixen la posició d'un element respecte d'altres, i n'hi ha 5 de diferents. Aquests es declaren nomenant la paraula *position* i tot seguit, entre els símbols ":" i ":", símbols utilitzats en tots els estils, s'indica un dels 5 tipus. Dintre d'aquests 5 tipus hi trobem la propietat *static*, estil que no aplica cap canvi en particular, però remarca que l'element es mogui d'acord amb el moviment de tota la pàgina. També tenim la propietat *relative*, on l'element es situa respecte a la seva posició normal, però amb una sèrie de canvis que es poden aplicar en les quatre direccions (dreta, esquerra,

amunt i avall). També tenim la propietat *fixed*, que indica que l'element no es mou seguint el moviment de la pàgina, no es mou de cap manera. Per últim, tenim la propietat *absolute* que posiciona l'element respecte a l'últim element amb una jerarquia superior a aquest i *sticky* una mescla entre les propietats *relative* i *fixed*. Tots aquests estils, menys en el cas de *static*, es poden desplaçar en les quatre direccions amb altres línies de codi.

- **Estils de models de caixa:** uns altres estils lligats a la posició són 4 variables que defineixen com es presenta la caixa o zona d'un element. Com es pot veure en la figura 21, tenim les propietats *content*, *padding*, *border* i *margin*, des de la part més interior de l'element a l'exterior. *Content* fa referència a la caixa en si de l'element, on hi ha el contingut. El *padding* és l'espai entre el contingut i la bora, aquesta estilitzada amb *border*. Per últim, tenim *margin* que és l'espai entre la part exterior de la bora, i la posició indicada amb els estils de posició.
- **Estils de color:** CSS també permet canviar el color de diferents elements com, text, àrees, fons i bores. Aquests estils són molt fàcils de declarar, amb instruccions com *color* o *background-color*.
- **Estils de text:** el text, tant en paràgrafs com en títols i capçaleres, és un dels elements més utilitzats en les pàgines webs, i consegüentment també tenen molts d'estils per editar-los. El text el podem estilitzar modificant el seu color, com ja hem destacat, la mida, la font, l'amplada, l'ombra, l'estil, com per exemple en cursiva, els diferents colors que adapten els enllaços, entre molts altres.
- **Estils de mida:** aquests estils s'utilitzen en diverses ocasions, però permeten indicar la mida dels elements com imatges, vídeos o altres. Aquestes mides s'indiquen utilitzant l'alçada i l'amplada.

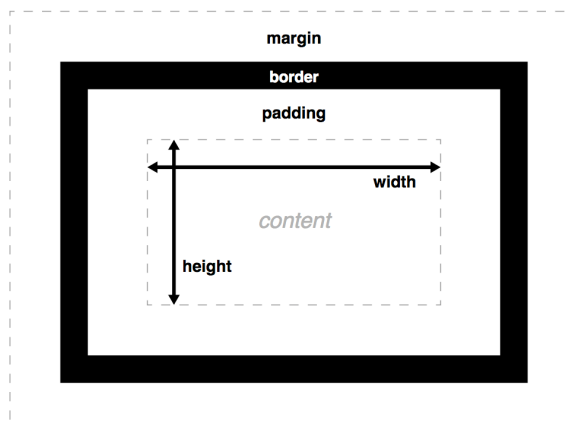


FIGURA 21. Esquema del model de caixa. FONT. MDN web docs - Recursos per desenvolupadors

Hi ha estils per aplicar en altres elements, com en taules o llistes, que també són molt utilitzats. Però un altre aspecte important d'aquest codi de programació són les unitats. Tots aquests estils funcionen amb uns certs valors i amb unes unitats importantíssimes per aconseguir certs efectes. Si parlem de distàncies, tenim unitats absolutes com els píxels o els mil·límetres. Aquestes unitats tenen un valor establert i invariable i les que s'utilitzen més freqüentment són els *px* (píxels). També tenim unitats relatives, com *em*, *vw* o *%*. Els percentatges s'utilitzen de la manera estàndard, *em* fa referència a la mida de lletra en l'element on s'aplica l'estil, i per últim *vw* equival a una centena part de l'amplada de la finestra de la pàgina web. En el cas de les propietats de colors, s'utilitzen dues estructures principals per definir-

los: RGB i sistema hexadecimal. En el cas del codi amb RGB s'indiquen 4 valors, les quantitats dels 3 colors (vermell, verd i blau) utilitzant nombres del 0 al 256 i també un valor que indica l'opacitat del color. En el cas del sistema hexadecimal, només s'indica un valor en base 16 que equival a un color en concret.

Per fer conversions entre unitats s'utilitza la funció *calc()* la qual permet fer operacions utilitzant unitats diferents. Aquesta funció és molt útil i permet estalviar conversions laborioses amb unitats relatives que d'altra manera s'haurien de fer a mà.

Per últim, un altre aspecte molt important que es va acabar aplicant al web final és un sistema de columnes i files. Aquesta funció del CSS no tracta d'estils, sinó de la distribució del contingut en contenidors i taules en les dues dimensions de la pantalla. De manera simplificada, facilita l'estructuració d'una pàgina web escrivint unes dreceres amb CSS que després es poden utilitzar en els programes en HTML. Aquesta secció dels estils no es va haver de programar perquè la plantilla utilitzada ja incorporava aquest complex sistema de distribució en el programa principal de CSS. Tot i això, aquests centenars de línies de codi van ser molt útils per organitzar el web d'una manera clara i fàcil d'entendre.

3.3.1.3. PHP

Utilitzant HTML i CSS, un programador té molta facilitat per crear webs amb les estructures, els estils i el contingut desitjat, però a l'hora de gestionar funcions més matemàtiques, de càlcul o de gestió de variables, tenen certes limitacions. Per això també s'utilitza PHP com a llenguatge de programació per realitzar aquests tipus de funcions. PHP significa processador d'hipertext i és un llenguatge de codi influenciat per altres llenguatges de programació, com C++ o Java, destinat a complementar codis de programació com HTML.

Una de les raons perquè és tan utilitzat és perquè és molt senzill d'usar, però alhora té el potencial suficient per poder realitzar qualsevol funció que necessiti un programador expert. A part, comparat amb altres llenguatges de programació, es podria dir que té una tolerància baixa als errors de codi, el qual també en facilita l'ús. Una altra característica que el fa molt útil és la possibilitat d'inserir el codi d'aquest llenguatge en el mateix programa escrit amb HTML. Simplement quan es vol començar i acabar d'utilitzar PHP en comptes del llenguatge base, es citen els símbols `<?php` i `?>`, d'aquesta manera l'ordinador entén que tot el que hi hagi entre aquestes dues etiquetes és codi en PHP.

Sobre les funcions bàsiques, trobem les següents:

- **Variables:** per tal de fer operacions és essencial poder guardar valors, tant numèrics com de caràcters. En PHP la declaració de variables és força lliure, en sentit que no és necessari indicar quins valors haurà de guardar aquella variable. Només és necessari indicar que és una variable amb el símbol “\$” davant del nom d’aquesta. Un cop es fa servir per primera vegada, la variable esdevindrà una *string*⁴¹, una *array*⁴² o una variable numèrica.
- **Operadors bàsics:** igual que la majoria de codis de programació destinats al càlcul, PHP utilitza els operadors estàndards: “+”, “-“, “X” i “/“ per les operacions, “=“ , “>” i “<“ per les igualtats i altres com “==“ per la comparació o “===“ per una identitat.
- **Flow control:** per regular certes repeticions del programa, també s'utilitzen les expressions estàndards. S'utilitzen les expressions *if* i *else* per a les instruccions bàsiques equivalents a “si” i “si no” per executar certes línies de codi només si una sèrie de condicions es compleixen i *while* i *for* per a la repetició de certes línies. També és possible l'ús de funcions.
- **Outputs:** per poder representar informació a través d'un mitjà de sortida, com pot ser una pàgina web. La instrucció *echo* és la que permet mostrar qualsevol sèrie de valors.

3.3.1.4. SQL

SQL (*Structured Query Language* - Llenguatge d'interrogació estructurat) és un llenguatge de comunicació amb bases de dades, senzill i basat en consultes. És a dir, mitjançant SQL es poden escriure unes instruccions que les bases de dades relacionals entenen i poden executar-les, aplicant canvis o emmagatzemant dades. Aquest llenguatge de programació també es pot utilitzar dins altres llenguatges, igual que PHP, d'aquesta manera en el codi d'una pàgina web, es poden formular consultes per obtenir dades (textos, imatges...) i mostrar-les en la pàgina web.

Per tant, aquestes consultes són essencials per gestionar l'emmagatzematge de contingut però tenen una estructura molt intuïtiva i fàcil d'utilitzar. Per exemple, per esborrar un objecte d'una taula d'una base de dades s'indicarà utilitzant *DELETE* que indicarà que es vol eliminar alguna cosa. Després s'assenyalarà el que es vol esborrar i per últim s'especificarà de quina taula es vol eliminar aquest objecte utilitzant *FROM* i posteriorment el nom de la taula. D'alguna manera, aquestes consultes tenen sentit si es formulen en anglès, cosa que fa que sigui molt més fàcil d'utilitzar.

⁴¹ Seqüència de caràcters.

⁴² Conjunt de variables numèriques organitzades de manera unidimensional.

Aquest llenguatge de programació té com a fonaments l'àlgebra relacional i el model relacional, que són conjunts d'operacions que treballen per obtenir una resposta entre diverses relacions. Per tant, aquests models s'apliquen a les bases de dades, on les taules es relacionen entre elles, adoptant el nom de relacionals. Aquestes relacions s'expliquen més detalladament en l'apartat 3.3.2.

Les instruccions més bàsiques del llenguatge són les següents:

- **SELECT:** per extreure informació de taules d'una base de dades. Especificant què vols obtenir, de quina taula i quines característiques ha de tenir aquesta entrada en una taula, pots obtenir un resultat o varis. Aquests els pots recuperar un per un utilitzant una funció com *while* de PHP.
- **ADD:** per afegir una entrada o una fila en una taula d'una base de dades. El que s'ha d'especificar és el nom de la taula, i la informació que es vol guardar.
- **UPDATE:** permet actualitzar una entrada ja existent, indicant la taula, la fila i els valors que es volen actualitzar.
- **DELETE:** permet esborrar una fila d'una taula.

A part d'aquestes instruccions bàsiques, també n'hi ha d'altres amb funcions més específiques que no són tan utilitzades. Però hi ha altes paraules claus importants:

- **FROM:** paraula utilitzada per indicar la taula de la base de dades que s'està manipulant.
- **WHERE:** instrucció per indicar una condició. En català equivaldria a un "on" i posteriorment s'indica una columna de la taula, i se n'especifica el valor que donarà lloc a la condició, limitant així els resultats de la consulta.
- **COUNT:** aquesta funció conta el nombre d'entrades o files amb una sèrie de condicions, i et torna el valor com a variable. També pot retornar el valor de cadascuna de les files depenent de com s'hagi formulat la consulta *SELECT*. Aquesta funció normalment és complementada amb la instrucció *ORDER BY* que ordena les entrades que ha utilitzat d'una manera determinada.
- **JOIN:** manera de relacionar taules en una consulta. Aquestes relacions, pròpies del model relacional però també de la teoria de conjunts, permeten establir uns lligams entre una sèrie d'elements. Per exemple, utilitzant aquesta funció amb la paraula *INNER* es poden unir dues taules indicant que una columna c1 de la taula t1 és igual o emmagatzema el mateix contingut que la columna c1 de la taula t2. Aquesta funció és una de les més útils i utilitzades per augmentar el nivell de complexitat de les consultes i aconseguir resultats molt més específics.

Aquestes consultes, un cop formulades, s'emmagatzemen com a *strings* utilitzant un fragment de programa en PHP, tal com es pot veure en la figura 22. Un cop fet això,

es confirma la connexió amb la base de dades des del programa, i tot seguit es transmet aquesta consulta per ser interpretada per la base de dades. Tot seguit, els resultats obtinguts s'emmagatzemen o bé en una sola variable en cas d'obtenir només un sol resultat, en una *array* en cas de tenir un resultat amb una sèrie de valors, en la figura es pot veure que s'emmagatzema en l'*array* anomenada *\$row*. Per últim, en els casos que s'obtenen més d'un resultat, i amb diferents elements, s'utilitza l'expressió *while* d'una manera determinada per recol·lectar tots els resultats.

```
$sql13 = "SELECT jugadors.nom, jugadors.cognom1,
jugadors.linked, jugadors.nickname, jugadors,
joc$id.punts FROM joc$id INNER JOIN jugadors ON
(joc$id.jugadors=jugadors.idjugador) INNER JOIN partides
ON (joc$id.partida=partides.idpartida) WHERE
(partides.usuari='martijuanola' OR
partides.usuari='arnaupadres')";

$result13 = mysqli_query($conn,$sql13) or
die(mysqli_error());
while ($row13 = mysqli_fetch_array($result13))
{
    $total=$total+$row13['punts'];
    $x++;
}
```

FIGURA 22. Consulta en SQL i fragment de PHP utilitzats en el web. FONT. Elaboració pròpia mitjançant *Geany - text editor*.

3.3.1.5. JavaScript

Per últim, destacar un llenguatge de programació minoritari en aquest projecte però amb una importància enorme dintre el desenvolupament de pàgines web en general. Aquest llenguatge de programació anomenat JavaScript és un dels 3 més importants a l'hora de constituir els codis de la WWW. Aquest llenguatge destaca pel seu potencial a l'hora de crear elements interactius, mòbils i visualment estètics en les pàgines web. A causa de la seva importància, tots els buscadors web suporten aquest llenguatge, i és usat en la majoria de casos. La sintaxi no és senzilla però utilitza les mateixes funcions i els mateixos operadors que la majoria d'altres llenguatges. Unes de les propietats més importants que té és el suport universal, com s'ha comentat, la seva estructura clara, la possibilitat de múltiples expansions (més funcions), i la seva funcionalitat. També destacar que es pot utilitzar dins programes amb HTML mitjançant les etiquetes `<script>` i `</script>` que indiquen les línies on s'utilitza aquest altre llenguatge.

Per aquesta última raó, i tot i que és un llenguatge de programació amb un gran potencial, aquesta manera d'utilitzar-lo dintre altres programes pot provocar problemes de seguretat.

En aquest projecte, JavaScript s'ha utilitzat per poder desenvolupar el menú interactiu amb els desplegable de l'índex, per gestionar algunes animacions i transicions menors, i altres funcions no tan importants com poden ser les alertes en

forma de *banner* a l'hora, per exemple, d'esborrar un joc. Tot i això, JavaScript du a terme un paper no gaire important en el web, ja que és merament visual i no funcional.

3.3.2. Base de dades

Una base de dades és un conjunt d'informació organitzada, emmagatzemada i amb un accés electrònic. Aquestes han canviat molt durant l'evolució tecnològica, bàsicament a causa de la millora dels components electrònics dels ordinadors, com els microprocessadors o els components d'emmagatzematge d'informació. Però durant la segona meitat del segle XX, es van fer una sèrie de propostes per tal de gestionar grans quantitats d'informació ordenades i relacionades d'una manera concreta. D'aquí es van originar les DBMS⁴³, d'on van sortir tipus de bases de dades com la d'objectes, de navegació o relacionals. Cada una d'aquestes organitza la informació de maneres diferents, per exemple les centrades en objectes, guarden la informació en elements i en canvi les bases de dades relacionals en taules.

En aquest projecte s'ha utilitzat una base de dades relacional o RDBMS, ja que és un dels models més utilitzats en aquests webs simples. Aquestes bases de dades estan basades en un emmagatzematge d'informació en forma de taules, i usen SQL com a mitjà de comunicació i de manipulació d'aquesta informació. Aquest model relacional es va descriure per primera vegada als voltants de 1970 per Edgar F. Codd, un especialista en ciències de computació, que en aquell temps treballava per l'empresa IBM.

Les taules d'aquest model estan formades per columnes i files, i on cada fila té un nombre únic i que la identifica, anomenat clau primària. Amb aquestes claus es pot relacionar la informació de manera molt senzilla, ja que només s'han de tenir present com es distribueixen aquestes claus i com es relacionen.

Per exemple, en la base de dades d'aquest projecte, hi ha una taula de partides, on s'emmagatzemen les partides, indicant el joc, el nombre de jugadors i el dia que es va jugar a aquesta partida, i també hi ha una taula de jocs, on s'emmagatzema una sèrie de característiques i informacions de cada joc. Per relacionar aquestes dues taules, o sigui per indicar el joc d'una certa partida, s'aprofiten les claus primàries de la taula de jocs, per indicar-lo en les files de la taula partides. Quan una clau primària s'utilitza en una altra taula, creant una relació, s'anomena clau secundària. Així, en una fila de la taula de partides, i en la columna de jocs, hi ha un nombre que és una clau secundària, que equival a una clau primària d'una fila de la taula de jocs. Aquest tipus de relació és la base de les RDBMS.

⁴³ *Database management system* (Sistema de Gestió de Bases de Dades).

Aquestes estructures es poden complicar molt si s'afegeixen moltes taules i alhora moltes claus primàries i secundàries, i per representar-les d'una manera còmoda es poden utilitzar una sèrie de gràfiques destinades a indicar les relacions entre taules i les taules en si. L'estructura de la base de dades del projecte es pot veure a la figura 23.

En aquesta gràfica s'hi poden trobar les diferents taules que formen la base de dades, amb les columnes d'aquestes. Per exemple la taula "tipusjoc" només té dues columnes: "idtipus", que és la clau primària (representada amb una clau) i "tipusjoc". També, al costat de cada nom de columna, s'especifica el valor que es guarda, ja que no sempre és el mateix. Per exemple, els nombres es guarden en variables numèriques, les paraules o cadenes de caràcters curtes en el format *VARCHAR*, els textos en les variables *TEXT*, les dates (any, mes i dia) en el format *DATA*, i les fotos en format *BLOB*. Després, també es poden ajustar la quantitat d'informació emmagatzemada indicant un màxim d'aquesta. Per exemple, un *TINYINT*, és una variable numèrica que guarda només un valor, per tant com que es treballa en binari, pots guardar-hi el valor 1 o el valor 0.

El gràfic també ens permet veure les relacions entre les taules, mitjançant unes connexions. Aquestes indiquen que, per exemple, la clau primària de la taula "tipusjoc", s'utilitza com a clau secundària en la taula "jocs" i en la columna "tipusjoc".

Les taules que trobem a la gràfica tenen una funció d'emmagatzematge concreta: la taula "jocs" guarda les dades de cada joc, per tant, cada joc que un usuari entra és una fila d'aquesta taula amb un nombre a la columna "idjoc" (clau primària de la taula "jocs") únic. Les taules "subtipus" i "tipusjoc" són taules no dinàmiques, o sigui que no hi canvia el nombre d'entrades o files, i la seva funció és guardar una sèrie d'opcions que té un joc de taula, com per exemple si és cooperatiu o col·laboratiu. La taula "marques" també té una funció semblant a la de jocs, ja que guarda totes les marques i la seva informació, i així cada joc pot indicar utilitzant la clau primària de "marques" de quina marca és. Les taules "usuaris" i "jugadors" emmagatzemen els usuaris i jugadors, amb la pertinent informació i es relacionen amb moltes taules per especificar de qui és la informació entrada al web. La taula "partides" guarda la informació de les partides, però només guarda les dades del joc, la data i el nombre de jugadors. La resta d'informació, com per exemple els punts que ha fet cada jugador, es guarda en les taules anomenades "joc#". Cada taula d'aquestes corresponen a una entrada a la taula "jocs" i la seva funció és emmagatzemar les dades de les partides del joc que correspon a la taula. Per tant aquestes taules estan lligades a la taula "partides", la taula "jugadors" i també, sense usar claus, a la taula "jocs", ja que el número que acompanya a la paraula "joc" equival a la clau primària de la taula "jocs".

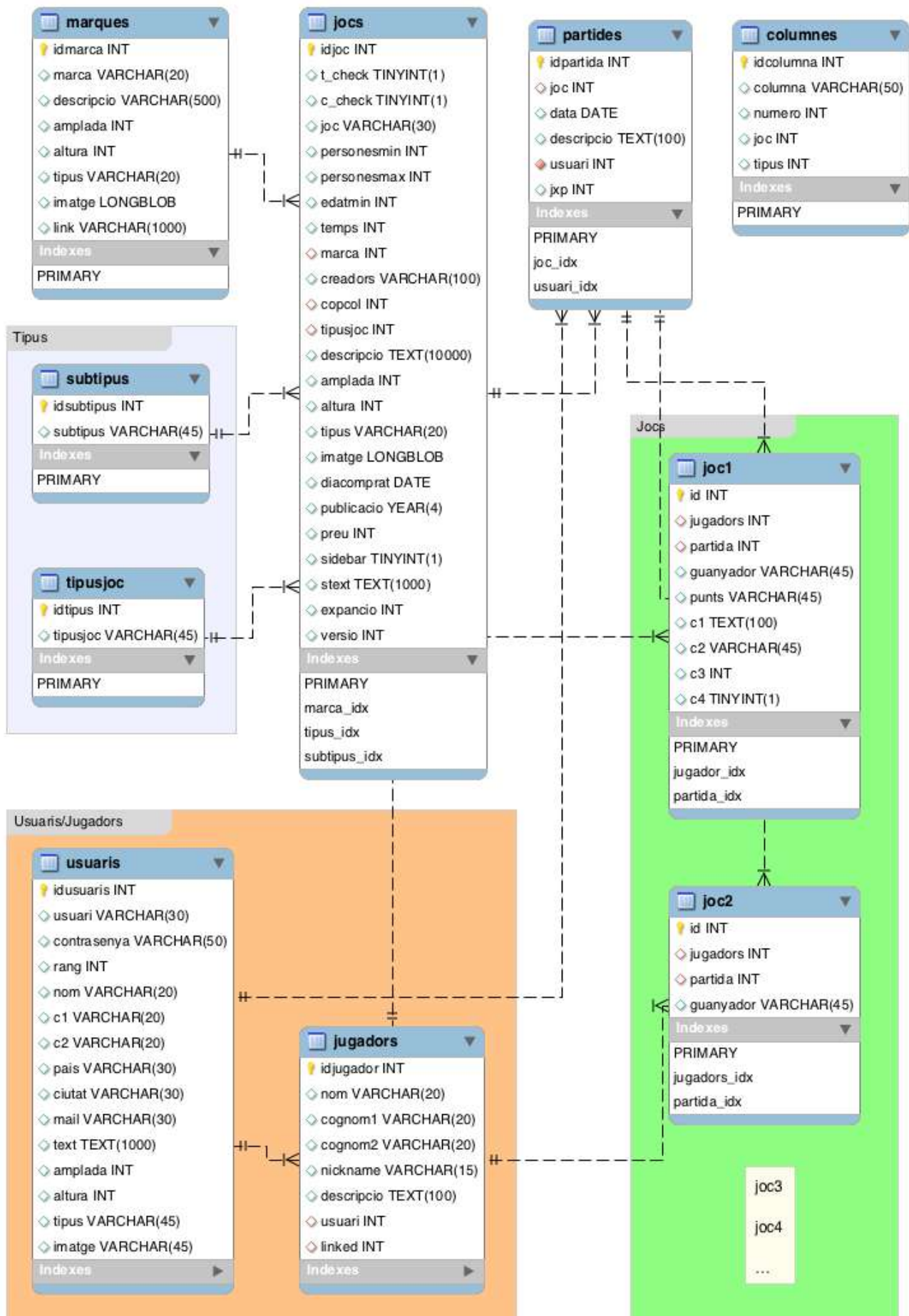


FIGURA 23. Estructura de la base de dades relacional del web. FONT. Elaboració pròpia mitjançant MySQLWorkbench: plataforma de configuració de bases de dades.

Per exemple, si la fila amb "idjoc" igual a 3 correspon al joc *Carcassone*, la taula "joc3" emmagatzemarà els resultats dels jugadors quan hagin jugat a *Carcassone*. Continuant l'exemple, si dues persones "a" i "b" fan una partida a aquest joc, un cop entrada, es guardarà una entrada a la taula "partides" on s'especificarà el joc, i el nombre de jugadors de manera obligatòria, juntament amb un "idpartida" únic "n". Immediatament després, també es crearan dues entrades a la taula "joc3": la primera indicarà la puntuació del "jugador a" i la segona la puntuació del "jugador b", indicant també que els dos han jugat a la mateixa partida amb un "idpartida" igual a "n".

Aquesta manera de guardar la informació pot semblar complexa, però és la manera més eficient de poder guardar tota la informació de manera ordenada i accessible. Per últim, i complicant una mica més l'estructura, hi ha la taula "columnes" que serveix per especificar les columnes de les taules "joc1", "joc2"... ja que no sempre es vol guardar la mateixa informació per a cada joc. Per això, les taules dels jocs tenen un nombre diferent de columnes, que guarden diferents valors, com el guanyador de la partida o els punts de cada jugador, però també d'altres que l'usuari ha volgut guardar, com per exemple uns punts parcials, o el personatge que utilitzava cada jugador guardant el nom en un *VARCHAR*.

3.3.3. Diferents programes i estructures

Els diferents fitxers de codi que s'emmagatzemen en el domini de la web són els programes que descriuen com és i que ha de fer la pàgina web, que es comuniquen amb la base de dades i amb l'usuari que utilitza el web.

Aquests fitxers es poden dividir en dos grups: els estils i el programes pròpiament dits. Els estils contenen les fonts i icones, els arxius amb CSS, els arxius en *JavaScript* i altres arxius de menor importància, en total 44 fitxers. Les parts més importants de CSS i *JavaScript* contenen cadascun un programa principal anomenat "main" on hi ha la majoria d'informació i que es cita a cada un dels altres programes del web perquè tots tinguin els mateixos estils generalitzats. L'arxiu "main.css" (CSS) és un arxiu de 4000 línies de codi i descriu tots els estils de la pàgina web, des de colors, formes, estructures i altres, i en canvi "main.js" (*JavaScript*) és un arxiu més reduït que conté diverses animacions com la de la barra lateral.

Però els fitxers més importants són els que tenen les instruccions en PHP i HTML per indicar com ha de ser la web. D'aquests programes n'hi ha 30, i els podem dividir de la següent manera: programes generals, com la barra lateral, programes de seguretat i d'entrada, sobre jocs, marques, jugadors i partides. Aquests últims tenen

sempre com a mínim 4 programes principals amb les funcions de crear, editar, eliminar i mostrar, d'aquesta manera, i mitjançant la base de dades, tens una pàgina per crear jocs nous, una pàgina per editar jocs ja creats, una pàgina per eliminar jocs i una pàgina, normalment la més complexa, per poder veure la informació guardada d'un joc. Aquests programes estan generalitzats perquè d'aquesta manera només necessites fer un programa per mostrar tants jocs com es vulgui.

Per ordenar tots els fitxers, tenen uns noms intuïtius que consisteixen d'una lletra que correspon als objectes que afecta aquest programa (en el cas de jocs seria una "j"), i després separada amb una barra baixa el nom amb anglès de la funció que fa ("add" per afegir, "edit" per editar, "read" per veure el contingut, etc). D'aquesta manera el programa "m_edit.php" serveix per editar marques. Aquest sistema no és perfecte però, dintre del possible, s'ha seguit per ordenar els fitxers.

3.3.3.1. Estructura bàsica i barra lateral

Tots aquests programes encara que tinguin funcions diferents tenen una mateixa estructura. Com es pot veure en la part de codi (codi d'inici i final sense contingut) sempre hi ha unes parts claus:

- **Codis d'entrada i permisos:** aquesta part està explicada en l'apartat de seguretat, però en tots els programes s'inclouen dos programes anomenats "s_info.php" i "inici.php". El primer serveix per connectar el web amb la base de dades, i l'altre per confirmar els permisos que l'usuari té com a visitant o com a registrat.
- **Crida d'estils inicials:** abans d'escriure o representar res a la pàgina web, primer s'han de cridar els fitxers d'estils com "main.css" per fer utilitzar-los. També es descriu el nom representat en el buscador quan estàs visualitzant la pàgina.
- **Estructura d'HTML:** ja que tots els programes estan escrits amb HTML com a programa base, tots han de tenir les etiquetes bàsiques pel seu funcionament:

```
<?php include 's_info.php'; include 'inici.php'; ?>
<!DOCTYPE HTML>
<html>
  <head>
    <title>GAMES-Homepage</title>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width,
initial-scale=1, user-scalable=no" />
    <!--[if lte IE 8]><script src="assets/js/ie/
html5shiv.js"></script><![endif]-->
    <link rel="stylesheet" href="assets/css/main2.css" />
    <!--[if lte IE 9]><link rel="stylesheet" href="assets/
css/ie9.css" /><![endif]-->
    <!--[if lte IE 8]><link rel="stylesheet" href="assets/
css/ie8.css" /><![endif]-->
  </head>
  <body>
    <div id="wrapper">
      <div id="main">
        <div class="inner">
          <?php include 'header.php'; ?>
        </div>
      </div>
    </div>
    <?php include 'sidebar.php'; ?>
  </body>
</html>
```

FIGURA 24. Codi base de les diferents pàgines del web.
FONT. Elaboració pròpia mitjançant *Geany - text editor*.

`<html>`, `<head>` i `<body>`, juntament amb altres seccions utilitzant `<div>` per declarar certs estils.

- **Capçalera:** per representar la part superior del web, que conté el títol de la web i l'usuari en cas d'estar registrat, s'inclou el programa "header.php" que té aquesta funció.
- **Barra lateral:** per últim, i després d'haver representat tot el contingut del web, es representa la barra lateral incloent el programa "sidebar.php". Aquest programa té les instruccions del menú juntament amb uns estils que es declaren al final de la pàgina.

Aquestes parts generalitzades són presents en tots els programes que formen una pàgina en el web.

3.3.3.2. Seguretat

Com que el web ha de ser únic per a cada usuari, ha de tenir un sistema d'entrada perquè cada usuari pugui entrar mitjançant el seu nom d'usuari i la seva contrasenya. I tot aquest procés ha de tenir uns mínims de seguretat, perquè la contrasenya continuï secreta, menys per l'usuari. A part, també ha de ser capaç de detectar que ja t'has registrat en aquesta sessió (període recent al web). Per últim, també ha de limitar l'accés a diferents pàgines segons el rang que té cada usuari. Per exemple, si un usuari encara no ha entrat amb la seva contrasenya, se l'hi ha de limitar l'accés a pàgines com la d'editar el seu perfil o els seus jugadors. Per dur a terme aquest procés, hi ha dos programes necessaris.

El primer és un petit programa que s'inclou a l'inici de cada programa i que serveix per comprovar que l'usuari s'hagi registrat i que pugui entrar amb els seus permisos. Aquest programa es diu "inici.php" i utilitza unes variables globals anomenades *SESSION* amb PHP, que tenen la característica de poder-se obtenir fins i tot si canvies de programa dintre d'un mateix domini. D'aquestes variables se n'utilitzen 3: una pel nom d'usuari, una per la contrasenya i per últim una pel rang, que es representa amb un número. El programa d'inici té dues funcions importants. La primera funció es diu "jugadors" i comprova que la contrasenya i l'usuari siguin els correctes, i que si encara no has entrat, et consideri com a usuari visitant. La segona s'anomena "permís" i comprova que amb el teu rang d'usuari puguis visualitzar la pàgina que has obert. La primera funció s'executa sempre que entres a una pàgina, en canvi la segona funció només en els casos que hi hagi una restricció, com per exemple els programes que manipulen dades de la base de dades.

L'altre programa es diu "entrar.php", i amb aquest pots realitzar 3 accions: entrar a la web com a visitant, però tenint pàgines restringides, entrar amb l'usuari i la

contrasenya ja existents d'un usuari o registrar-se per primer cop, emplenant un formulari amb les dades de l'usuari. Es pot accedir a aquest programa des d'un enllaç a la capçalera, en cas de no haver entrat encara. A la barra lateral, també hi ha la possibilitat de sortir del teu compte.

Per millorar la seguretat de les contrasenyes, es guarden a la base de dades encriptades, utilitzant el mètode MD4⁴⁴. Tot i això, aquest mètode d'encriptació no és aconsellable per encriptar contrasenyes, ja que es pot desencriptar fàcilment utilitzant la força bruta.

3.3.3.3. Usuaris i jugadors

El programa que gestiona els usuaris és el que s'ha comentat en l'apartat anterior anomenat "entrar.php", i que permet crear usuaris. La resta de funcions comunes com editar o esborrar usuaris no s'han aplicat en cap programa encara, i seria un aspecte de millora del web. El cas d'una funció per visualitzar usuaris tampoc és necessària, però sí que es podria fer un programa que permetés veure les dades de l'usuari registrat, i on també podria contenir les opcions d'editar o eliminar aquell compte d'usuari.

Les dades que es demanen i es guarden a la base de dades d'un nou usuari són el nom, la contrasenya, encriptada i demanada dues vegades per evitar errors com en la majoria de formularis on s'ha d'indicar una nova contrasenya, i altres dades no tan importants, com el nom i cognoms, una foto, el país, la ciutat o una descripció. Aquestes dades addicionals tindrien sentit si el web possibilités una millor relació entre usuaris.

Tot i això, a causa que l'objectiu del web és l'entrada de partides i per tant també de resultats individuals de cada jugador que ha participat en la partida, es necessita un mètode que no depengui d'altres usuaris per poder guardar jugadors que han jugat amb tu. Per tant, actualment hi ha una sèrie de programes destinats a la creació i modificació de jugadors, els quals són únics per a cada jugador. Els programes que hi ha són "p_add.php" per afegir jugadors, "p_edit.php" per editar jugadors, "p_individual.php" per poder veure la informació d'un sol jugador, i també amb l'opció d'eliminar-lo, i per últim "p_read.php" per veure tots els jugadors que té un usuari.

Per a cada jugador, es guarda el nom i els cognoms, un àlies i una descripció de manera opcional. L'àlies o el nom amb cognoms, en cas de no haver-se creat, seran utilitzats a l'hora d'entrar partides, on es podrà seleccionar.

⁴⁴ Algorithme de digestion de messages (*Message-digest*) desenvolupat l'any 1990.

Una altra opció que l'usuari té amb els seus jugadors és la possibilitat d'enllaçar-los a un altre usuari. Per exemple, si un usuari ha afegit a un "jugador x", i aquest jugador s'ha fet un compte d'usuari, l'usuari inicial pot indicar que el "jugador x" també és l'"usuari x". Actualment aquesta funció no té cap avantatge a part d'obtenir el nom d'usuari en comptes del de jugador, però en un futur es podrien establir una sèrie de relacions perquè una partida que on hi han jugat dos usuaris només s'hagi d'entrar una vegada, i l'altre tingui l'opció d'incorporar-la a les seves partides i per tant també a les seves estadístiques.

3.3.3.4. Jocs i marques

Els programes utilitzats per jocs i marques tenen les mateixes funcions: afegir, editar, visualitzar en global, visualitzar individualment i visualitzar imatges. Per tant, cada funció té dos programes: un per les marques i un pels jocs. Tot i això, hi ha algun programa més que es dedica als jocs, per exemple, per fer les estadístiques dels resultats i per configurar la taula del joc amb les columnes que guarden informació personalitzada.

Tots els programes relacionats amb marques podrien ser opcionals perquè només aporten informació extra als jocs, i no influeixen en la funcionalitat de la web, però es van aprofitar els programes de la primera versió de la web i al final també s'han aplicat a la versió final. La part dels jocs en canvi, és necessària per poder presentar unes estadístiques, ja que necessites el sistema de crear taules a les bases de dades.

Si ens fixem com s'estructuren els programes de certes funcions es pot veure que els programes d'afegir informació com "j_add.php" tenen entre 2 o 3 parts diferenciades. La primera vegada que s'executa un programa d'aquest tipus, l'usuari veu el formulari que pot emplenar amb les dades demanades, i després de prémer el botó de *submit*, la pàgina es torna a executar però ara només realitzant la funció d'interpretar les dades entrades i transmetre-les a la base de dades, a la taula i columna pertinents. Les funcions d'editar també funcionen de manera semblant, però en el primer formulari, s'hi mostren les dades que conté la base de dades. D'aquesta manera l'usuari pot decidir canviar les dades o deixar-les com estan.

El programa "j_edit.php" que s'encarrega d'editar els jocs, és una mica més complex que els altres programes d'editar, ja que engloba 3 funcions alhora: l'usuari pot editar les dades del joc, pot especificar les dades que es volen guardar en la base de dades, i per últim pot eliminar el joc. Com que els programes de cada funció són

relativament llargs, es guarden en diferents fitxers que s'inclouen tots a "j_edit.php". La funció d'esborrar el joc funciona utilitzant JavaScript per mostrar un missatge de confirmació abans d'esborrar permanentment una sèrie de dades de la web.

3.3.3.5. Partides

Programes relacionats amb les partides només n'hi ha un. Aquest programa ("pa_add.php") s'encarrega d'entrar partides i és un dels més complexos que s'han hagut de fer pel web. Aquest programa té 3 parts. Primer de tot, s'executa un formulari inicial el qual demana el joc que s'ha jugat, el nombre de jugadors, la data i una descripció opcional. Aquest formulari s'ha de realitzar abans que el principal perquè primer de tot es necessita saber què has de demanar perquè cada joc emmagatzema una sèrie de dades diferents, i per quants jugadors s'han de fer els formularis. Per tant un cop s'ha obtingut aquesta informació, es torna a executar el web però amb la segona part, on hi ha tots els formularis per entrar les dades ordenades en forma de taula. El nombre de files és igual al nombre de jugadors més un per la capçalera amb títols, per tant, cada fila contindrà la informació d'un sol jugador. Les columnes en canvi determinen la informació que s'ha d'entrar, que l'usuari haurà especificat prèviament des d'una de les funcions d'editar jocs. Tot i això, la primera columna mostrar una sèrie de desplegable perquè es pugui escollir el jugador que es vol, amb l'ordre que es vol. Les altres columnes permetran entrar informació mitjançant una zona de text, un formulari de text o una opció binària. Un cop acabada aquesta part, s'executarà la pàgina altra vegada i es formularan les consultes necessàries per entrar totes les dades a la base de dades, que són una per cada usuari o fila de la taula del segon formulari, més una per l'entrada a la taula "partides".

Aquest programa va ser un dels últims de fer a la web, perquè per poder-lo fer es necessitava fer primer tots els altres que permetien poder tirar els jocs, els jugadors... A més la complexitat del programa surt dels grans processos de generalització que es van haver de fer perquè funcionés per tots els jocs i les seves columnes i per qualsevol nombre de jugadors. Sobretot el cas d'obtenir dades per a cada jugador fa que el sistema d'identificació de dades o *Inputs* hagi de ser molt rigorós, i que la formulació d'una sola consulta sigui d'aproximadament cent línies de codi.

Tota aquesta informació es guarda a la base de dades, i l'únic que falta per assolir l'objectiu del web és extreure aquestes dades en format estadístic. Per això, hi ha un programa que complementa "j_individual.php" (programa per visualitzar un joc i la

seva informació) anomenat “stats2.php” que es dedica exclusivament a formular consultes per extreure certes dades significatives de la base de dades.

Aquestes consultes es repeteixen de diferents maneres per aportar unes estadístiques generals del joc, però també estadístiques limitades a partides amb un cert nombre de jugadors. Les consultes actuen sobre la taula “partides” on aconseguix el nombre de jugadors de cada partida, i la taula individual del joc, on pot trobar victòries i punts. Les consultes generals i inicials del web són les següents:

- Nombre de victòries per jugador.
- Percentatge de victòries per jugador.
- Millor puntuació en una partida.
- Pitjor puntuació en una partida.
- Puntuació mitjana per partida.
- Mitjana de punts per jugador.

Com s’explica a l’apartat de millores (3.5.) un programa que s’hauria de realitzar a continuació és un que creés consultes especificades per a certs jocs, i que utilitzés la informació que només es guarda a aquell joc, com uns punts parcials.

3.4. Producte final i funcionalitat

Després de tot el procediment explicat i de l’aprenentatge de diferents conceptes i llenguatges de programació, el resultat final és una pàgina web funcional que es pot trobar a l’enllaç "<http://marti.centpeus.cat/index.php>". També s’hi pot accedir a través d’aquest codi QR.



FIGURA 25. Codi QR enllaçat amb el web. FONT. Elaboració pròpia mitjançant QR code generator.

El web final conté tots els aspectes comentats en els punts anteriors, i es basa l’ús d’una base de dades relacional i fitxers de codi, que contenen estils i els programes en si de la web. Aquest no s’ha publicat, ja que falten aspectes que pensem que són claus per obrir-la a un públic més gran, i per tant només s’ha pogut provar per poques persones.

Aquests canvis, millores i errors que s’ha de solucionar s’esmentés en el següent apartat.

Per consultar i utilitzar totes les funcions del web, s’hi pot accedir i provar-les. Després d’haver creat un usuari, totes aquestes opcions podran ser utilitzades sense cap limitació.

3.5. Possibles canvis i millores

Tot i que el projecte de creació del web es pot considerar un èxit, ja que s'han pogut assolir els objectius que s'havien plantejat, hi ha certs aspectes d'aquest que es podrien millorar. També hi ha noves funcions que, en el cas que s'hagués de presentar al públic, serien interessants. Per últim també es podrien corregir petits detalls i/o errors que encara es poden trobar al web, la majoria relacionats amb estils i a l'hora de presentar informació.

Un tema molt important també és el de la seguretat. En el web que s'ha acabat creant, hi ha un sistema d'usuaris amb contrasenyes que no és prou segur. Tot i que les contrasenyes s'encripten i les seqüències d'SQL estan ben protegides, encara s'hauria de millorar tot el sistema de seguretat. Una de les opcions més viables és la de poder registrar-se al web utilitzant un compte de *Google* o d'alguna altra plataforma de comunicacions important.

Alguns aspectes respecte a l'estructura del web que s'haurien de millorar són l'entrada de partides, jocs o marques i la visualització de jocs i marques. Aquests processos funcionen sense errors però sí que després de provar el web, a vegades són massa laboriosos, i s'hauria de pensar com fer dreceres que agilitzessin aquestes accions.

La informació que es guarda per cada usuari, joc o marca de joc també es podria ampliar, possibilitant l'emmagatzematge de més informació. I respecte a les partides, la quantitat d'informació no és un problema perquè cada usuari pot guardar tantes dades com vol, però sí que s'haurien de crear més resultats i anàlisis estadístics, ja que és la part principal del web. Aquests resultats també es podrien representar d'una manera més visual utilitzant gràfiques amb *JavaScript*.

Per últim s'hauria d'afegir una funció que pot ser molt útil per no repetir informació a la base de dades. Posem el cas que dos jugadors, "jugador 1" i "jugador 2", juguen una partida. Si el "jugador 1" ja ha entrat les dades d'una partida, el "jugador 2" hauria de rebre una notificació quan entrés al web dient d'afegir els resultats ja entrats al seu arxiu de partides. Aquesta funció és importantíssima i és una de les més útils que s'hauria d'aplicar.

Aquests exemples són algunes millores que es podrien aplicar, però per acabar de trobar errors o aspectes a millorar s'hauria d'agafar una mostra de gent per provar el web.

4. Temes d'anàlisi

4.1. Conjunts i Sets

El primer tema d'anàlisi és el tema de conjunts i sets. Aquest tema, des d'un punt de vista matemàtic, es basa en la teoria de conjunts. Aquesta branca de la matemàtica es centra en relacions i propietats de grups o conjunts d'elements. Els populars diagrames de Venn són exemples de tractament de conjunts.

Si ens centrem en el concepte "conjunt", però, el podem definir de la següent manera, tal com apunta Georg Cantor⁴⁵:

S'entén per "conjunt" qualsevol col·lecció M , considerada com un tot, d'objectes, de la nostra percepció o del nostre pensament, diferents i ben definits m (dels quals se'n dirà els "elements" de M).

Per poder manipular i operar amb aquests conjunts s'utilitzen diversos símbols com \cup o \cap que s'utilitzen per indicar unió o intersecció respectivament entre 2 o més conjunts. Com aquests també hi ha altres símbols i metodologies que s'utilitzen a l'hora de manipular conjunts.

4.1.1. Aplicació en jocs (*Fillers: Set*)

Els jocs escollits per aplicar la teoria de conjunts són jocs classificats com a *fillers*. Aquest tipus de jocs es caracteritza per un ritme ràpid i la importància d'habilitats físiques en les partides. Aquestes poden ser habilitats com la velocitat o la memòria, entre d'altres. El terme també indica que els jocs solen ser per molts jugadors.

En aquest cas, els dos jocs *fillers* que s'han triat són el *Jungle Speed* i el *Set*. Aquests dos jocs es basen en la velocitat d'observació dels jugadors i en la seva capacitat de crear relacions entre elements. Els dos jocs tenen com a element material diverses cartes, que tenen símbols amb diferents característiques. Depenent de les relacions d'aquestes característiques, els jugadors han de realitzar certes accions.

Si ens centrem en el joc *Set*, trobem cartes amb 4 característiques diferents: color, figura, nombre d'aquestes i el fons de cada forma. Cada una d'aquestes característiques té 3 possibles estats (3 colors, 3 figures diferents, d'1 a 3 figures per

⁴⁵ Matemàtic i filòsof alemany fundador de la teoria de conjunts moderna.

carta i 3 tipus de fons). D'aquesta manera, i sabent que el joc té cartes que no es repeteixen, podem saber que el nombre de cartes del joc és de $81(3^4)$.

Com diu el títol, l'objectiu del joc és trobar sets, que són combinacions de 3 cartes amb unes particularitats. Aquestes són que les 4 característiques de cada carta han de ser o bé totes iguals en les 3 cartes o bé totes diferents. Les normes del joc es poden trobar amb més detall a l'annex 1.

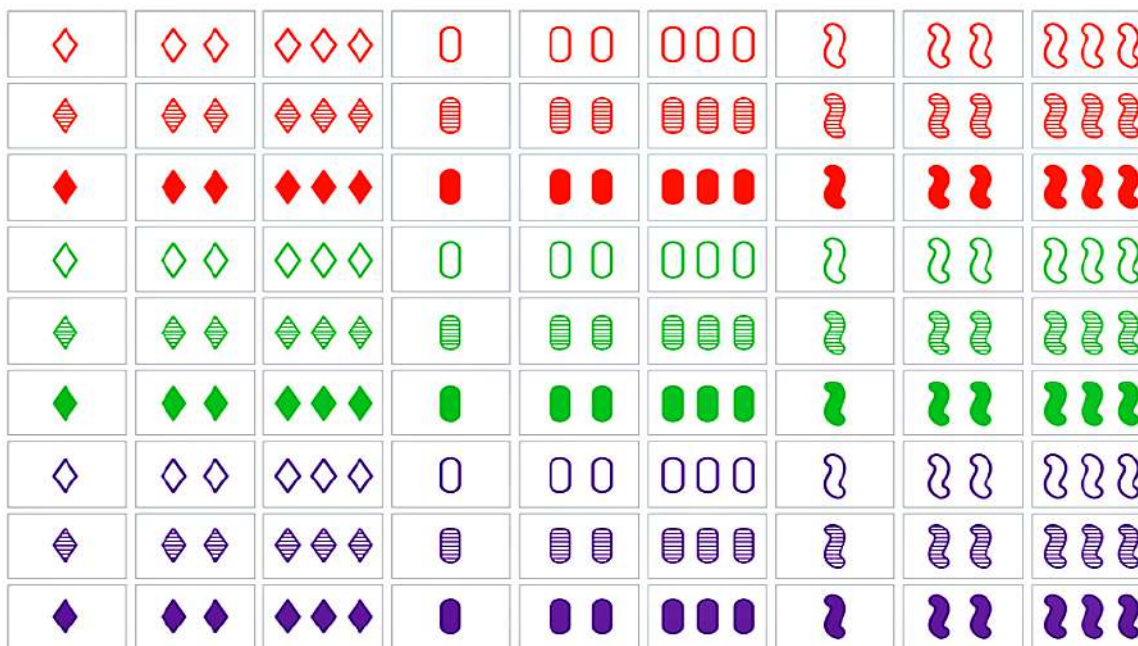


FIGURA 26. Totes les cartes del joc *Set*. FONT. Article de *Geek & Sundry* - *The card game that puzzled mathematicians for decades*.

4.1.2. Anàlisi

Una de les qüestions que ens podem trobar jugant al joc set és si realment hi ha sets en el conjunt de cartes sobre la taula. D'una manera més general, la pregunta seria quina és la probabilitat que hi hagi un set en un nombre determinat de cartes en joc. Aquesta qüestió es pot afrontar de diferents maneres, però en totes es fa servir la computació informàtica per obtenir resultats de manera més ràpida.

Els dos casos que podem eliminar immediatament són els casos amb 1 i 2 cartes, ja que se'n necessiten com a mínim 3 per poder fer un set. I també de manera senzilla podem determinar que per cada conjunt de 2 cartes només hi ha una tercera per poder crear un set. D'aquesta manera amb 3 cartes la probabilitat d'un set és de $\frac{1}{79} \approx 1.27\%$. Amb 4 cartes també es pot calcular la probabilitat de manera senzilla. En els casos que 3 cartes no fan un set ($\frac{78}{79}$) tenim 3 parells de dues cartes que necessiten una carta, que està en el piló restant de 78 cartes, per fer un set. És a dir

que podem escriure la probabilitat d'un set en un conjunt de 4 cartes com $\frac{1}{79} + \frac{78}{79} \times \frac{3}{78} = \frac{4}{79} \approx 5.06\%$.

En el cas de 5 cartes el càlcul es complica força. Podria semblar que repetint el mateix procediment que el grup de 4 cartes trobaríem el resultat, però no funciona. Aquesta operació errònia es realitzaria agafant les 6 combinacions de 2 cartes que hi pot haver en un grup de 4 cartes i aplicar-ho als percentatges de la següent manera: $\frac{1}{79} + \frac{78}{79} \times \frac{3}{78} + \frac{78}{79} \times \frac{75}{78} \times \frac{6}{77} \approx 12.46\%$. La raó perquè aquesta operació és errònia és que pot haver-hi dues combinacions de 2 cartes de les 6 que esdevinguin un set amb la mateixa carta. Això complica encara més el problema. Sabent que aquesta repetició d'una carta (anomenada *metaset*) només pot passar una vegada en un grup de 4 cartes (ja que dos set només poden tenir una carta en comú) i a través d'altres deduccions podem determinar que el percentatge d'un grup de 4 cartes sense sets és aproximadament de 12.412%.

Tot i això, aquest procés no es pot continuar manualment a causa de les complicacions creixents de repeticions de cartes i combinacions. En comptes d'això, per determinar la resta de possibilitats s'ha utilitzat la computació, mètode més ràpid i eficaç. Donald E. Knuth, professor d'art de programació computacional de la universitat de Stanford, va crear un programa que era capaç de trobar totes les probabilitats. Aquest programa és molt complex però funciona mitjançant l'ús de vectors de quatre components per indicar les cartes del joc i l'ús de combinacions i isomorfismes. El resultat d'aquest programa és la taula (figura 27) que respon a la pregunta d'aquest apartat.

Nombre de cartes	Percentatge	Nombre de cartes	Percentatge
1	0.000000	12	0.967698
2	0.000000	13	0.989971
3	0.012658	14	0.997686
4	0.050632	15	0.999635
5	0.124116	16	0.999966
6	0.237028	17	0.999998
7	0.383392	18	0.999999
8	0.546466	19	0.999999
9	0.702777	20	0.999999
10	0.830549	21	1.000000
11	0.918243		

FIGURA 27. Percentatges de sets en grups de n cartes.

4.2. Poliòminos i tessel·lacions

El segon tema d'anàlisi tracta de tessel·lacions i uns elements geomètrics anomenats poliòminos. Per tant aquest apartat està dedicat a la geometria en el pla.

Les tessellacions es defineixen com a conjunts de figures (tessel·les) que formen un patró concret i poden cobrir superfícies planes. Aquestes les trobem tant en l'art visual, en el món urbà en forma de rajoles, i també en jocs com pot ser el *tangram* o les mateixes peces de puzzle. Alguns dels tipus de tessellacions són els següents (Albertí, 2014)(les tessellacions en si es poden trobar en l'annex 2):

- **T. Regulars:** des d'un punt de vista matemàtic, les tessellacions regulars són aquelles que utilitzen només un polígon regular. Per tant, només existeixen 3 recobriments regulars, utilitzant triangles regulars, quadrats i hexàgons regulars. Els altres recobriments regulars no són possibles a causa que els angles dels altres polígons regulars es sobreposen.
- **T. Semiregulars:** aquests consisteixen en combinacions de polígons regulars, creant així 8 recobriments possibles.
- **T. Irregulars:** els irregulars, com ja ho indica el nom, són recobriments creats a partir de polígons irregulars. Aquesta categoria, per tant, té moltíssimes tessellacions possibles
- **T. Fractals:** aquestes es poden trobar tant en espais finits com infinits. En el cas dels finits, poden tenir forma quadrada, circular o altres. Un element important d'algunes d'aquestes tessellacions fractals és la dimensió dels recobriments, en cas que l'espai no sigui finit.
- **Altres:** altres recobriments que no es podien classificar en els apartats anteriors poden ser algunes com la metamorfosi d'Escher, tessellació que canvia al llarg del seu espai finit, o altres com recobriments aperiòdics, els quals no es repeteixen en cap moment.

Per altra banda tenim els poliòminos, formes introduïdes pel matemàtic Solomon Wolf Golomb en una conferència l'any 1953. Aquestes figures geomètriques, també de dues dimensions, són figures simples creades a partir de quadrats que es combinen fent coincidir arestes paral·lelament. Si ens fixem amb els poliòminos formats només amb un o dos quadrats, només en trobem dos, ja que només hi ha una manera de combinar-los en cada cas. Amb 3 quadrats, ja trobem dues figures: L-trinòmino i I-trinòmino, a causa de la seva forma de *el* i de *i* respectivament. Utilitzant quatre quadrats podem trobar-hi fins a 5 formes diferents i amb cinc quadrats 12 més, com es pot veure en la figura 28. La següent taula mostra el nombre de diferents figures possibles amb un nombre determinat de quadrats.

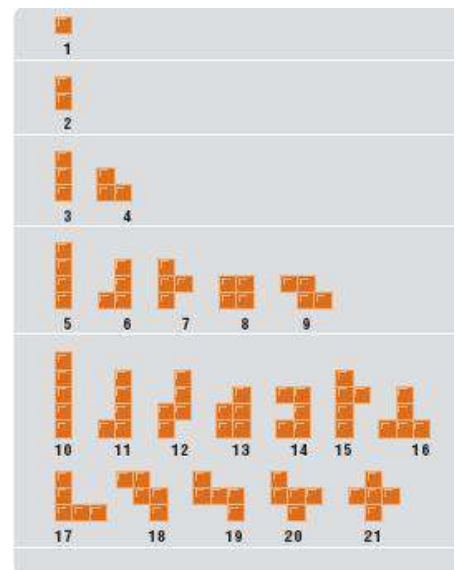


FIGURA 28. Poliòminos formats amb 1, 2, 3, 4 i 5 quadrats. FONT. Forum de BoardGameGeek - Blokus

Monòminos	Dòminos	Tròminos	Tetròminos	Pentòminos
1	1	2	5	12
Hexòminos	Eptòminos	Octòminos	Nonominos	Decòminos
35	108	369	1285	4655

FIGURA 29. Nombre de poliòminos diferents segons els nombre de quadrats utilitzats. FONT. Article de Wikipedia - Polyomino

Les figures formades amb 4 quadrats són utilitzades en el joc d'ordinador *Tetris*, per exemple, i el joc analitzat en els pròxims apartats utilitza totes les que es poden formar amb 1, 2, 3, 4 i 5 quadrats.

4.2.1. Aplicació en jocs (*Blokus*)

Ja s'han comentat alguns jocs com el tangram o el Tetris, que utilitzen tessellacions i els poliòminos per jugar, però el joc triat per aquest apartat és la combinació perfecta dels dos conceptes. Aquest joc es diu *Blokus* i és un joc de 2 a 4 jugadors publicat l'any 2000. L'objectiu d'aquest és que cada jugador ha d'intentar col·locar totes les peces que pot (poliòminos formats per 1, 2, 3, 4 i 5 quadrats sense repeticions) en un taulell de 20×20 . Aquestes peces però s'han de col·locar d'una manera determinada: després de posar la primera peça a una cantonada del taulell, totes les altres només es poden tocar per una o més cantonades d'altres peces del mateix jugador. En cap cas dos peces del mateix jugador poden entrar en contacte (aresta al costat d'aresta). En canvi, sí que es poden tocar amb les dels altres jugadors. Les normes senceres es poden consultar a l'annex 3.

4.2.2. Projecte: ocupació d'espai

La petita anàlisi d'aquest apartat té relació amb la col·locació de peces en el joc *Blokus*. En les normes oficials d'aquest joc es donen una sèrie de consells, dos dels quals diuen: "Al principi de la partida s'ha d'avançar progressivament cap al centre del taulell." i "Intenta col·locar, al principi, les peces grans perquè al final del joc només quedaran espais reduïts i serà més difícil col·locar-les". Les dues estratègies són lògiques i són necessàries per jugar una bona partida, però fins a quin punt s'han de tirar peces grans al principi de la partida? I un tetròmino es considera una peça gran o només els pentòminos?

Per intentar resoldre aquestes ambigüitats, o si més no per aportar més informació al tema, es va fer un estudi de com les poliòminos del joc *Blokus* són capaços d'ocupar un espai. Donat un espai $n \times n$, s'observa quantes peces iguals poden ocupar-lo, seguint les normes de col·locació de peces del joc. Aquest estudi es va realitzar amb

$n = 10$ i $n = 20$, per veure com es podien posicionar en un taulell reglamentari (20×20) de *Blokus* i un taulell d'una quarta part (10×10). Els resultats es poden veure en l'annex 4 (taulells) i en la figura 29 (percentatges).






















Figura						
10x10	$\frac{50}{100} = 50\%$	$\frac{50}{100} = 50\%$	$\frac{51}{100} = 51\%$	$\frac{45}{100} = 45\%$	$\frac{40}{100} = 40\%$	$\frac{52}{100} = 52\%$
20x20	$\frac{200}{400} = 50\%$	$\frac{200}{400} = 50\%$	$\frac{201}{400} = 50.25\%$	$\frac{180}{400} = 45\%$	$\frac{200}{400} = 50\%$	$\frac{200}{400} = 50\%$
Total	50 %	50 %	50.4 %	45 %	48 %	50.4 %
Figura						
10x10	$\frac{48}{100} = 48\%$	$\frac{52}{100} = 52\%$	$\frac{48}{100} = 48\%$	$\frac{50}{100} = 50\%$	$\frac{45}{100} = 45\%$	$\frac{45}{100} = 45\%$
20x20	$\frac{200}{400} = 50\%$	$\frac{200}{400} = 50\%$	$\frac{200}{400} = 50\%$	$\frac{200}{400} = 50\%$	$\frac{170}{400} = 42.5\%$	$\frac{190}{400} = 47.5\%$
Total	49.6 %	50.4 %	49.6 %	50 %	43 %	47 %
Figura						
10x10	$\frac{45}{100} = 45\%$	$\frac{45}{100} = 45\%$	$\frac{40}{100} = 40\%$	$\frac{45}{100} = 45\%$	$\frac{45}{100} = 45\%$	$\frac{45}{100} = 45\%$
20x20	$\frac{180}{400} = 45\%$	$\frac{185}{400} = 46.25\%$	$\frac{185}{400} = 46.25\%$	$\frac{180}{400} = 45\%$	$\frac{150}{400} = 37.5\%$	$\frac{195}{400} = 48.75\%$
Total	45 %	46 %	45 %	45 %	39 %	42 %
Figura						
10x10	$\frac{55}{100} = 55\%$	$\frac{45}{100} = 45\%$	$\frac{45}{100} = 45\%$			
20x20	$\frac{215}{400} = 53.75\%$	$\frac{160}{400} = 40\%$	$\frac{255}{400} = 63.75\%$			
Total	54 %	41 %	60 %			

FIGURA 30. Percentatge de quadrats ocupats en àrees de 10x10 i 20x20 per cada poliòmino del joc *Blokus*. També el percentatge dels resultats de les dos àrees.

Com es pot veure en la taula anterior, els percentatges generals es situen entre el 40% i el 60%. Aquests resultats ordenats es poden trobar a la figura 31, on podem veure, en ordre decreixent, les figures que ocupen més espai d'un taulell. Aquests resultats podrien ser interpretats de manera que les primeres peces que s'han de col·locar són les que tenen el percentatge més petit. Tot i això, si ho comparem amb altres estratègies, veiem que els nostres resultats no són del tot correctes. S'han d'interpretar amb matisos i altres estratègies.

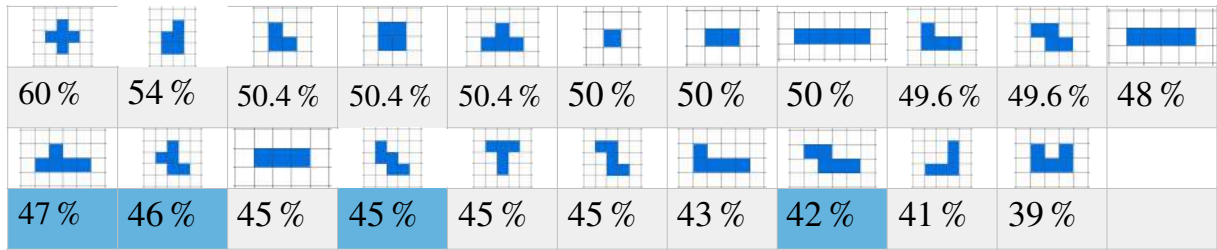


FIGURA 31. Ordre de percentatges totals de major a menor per cada políomino del joc *Blokus*.

Si ens fixem amb els 10 pitjors (que ocupen menys espai) podem veure que 9 són pentòminos i només 1 és de 3 costats. Això té sentit amb el consell que dona el joc, que diu que primer s'han de tirar les peces grans. També es pot veure que les altres onze figures progressivament disminueixen de mida, amb algunes excepcions, com les dues figures amb el percentatge més gran. Aquestes dos són pentòminos, i per tant s'haurien de tirar al principi de la partida, però són molt útils per ocupar territori.

Segons Mike Yosuke (*infamouswhiteknight*, 2013), jugador professional del *Blokus* i guanyador de la final de campionat de *Blokus* l'any 2014, hi ha quatre peces que són les més importants del joc. Aquestes quatre són els pentòminos N5, F5, W5 i Y5 (marcades en la taula 31), on Y5 és la millor de totes. Com podem veure en el nostre ordre, aquestes peces es situen entre l'onzena i antepenúltima posició. I encara que pugui semblar que les dues opinions són oposades, les peces inicials són les més decisives, i aquestes 4 es solen tirar entre les primeres 10.

Per tant, aquest estudi ha sigut capaç d'ordenar les peces de *Blokus* d'una manera lògica i possible, però amb alguns errors o equivocacions. Menys en el cas d'aquestes figures, aquest ordre resultant es pot aplicar a estratègies a l'hora de fer una partida al *Blokus*. Tot i això no és aconsellable seguir l'ordre exacte sinó que aquesta ordenació ha de servir de guia o d'aproximació per a possibles estratègies.

4.3. Combinatòria i probabilitat

L'últim tema d'anàlisi matemàtic de jocs de taula es situa en l'àmbit de la combinatòria i la probabilitat. Per una banda tenim la combinatòria, que és una branca de la matemàtica que es basa en l'enumeració d'elements que compleixen unes condicions (combinació enumerativa), a decidir quan els criteris es compleixen i a analitzar aquests elements (Combinatòria - Wikipedia, 2018). Elements com les permutacions o les combinacions són recomptes de combinatòria. En aquest apartat també s'inclou la probabilitat, tot i que aquesta branca més estadística de la matemàtica es pot trobar en tots els jocs de taula per la seva relació amb l'atzar parcial. En aquest apartat, però, tindrà un pes més important, relacionat en la creació de l'algorisme del primer objectiu del treball.

4.3.1. Aplicació en jocs (*Sushi Go!*)

Els jocs de taula seleccionats per aquest apartat són els *Sushi Go!* i el *7Wonders*. Els dos jocs es basen principalment en cartes, tot i que *7Wonders* també utilitza monedes i uns petits taulells individuals (un per a cada jugador). En relació a la mecànica de joc dels dos, comparteixen una mateixa característica envers la selecció de cartes. Els dos jocs tenen decisions simultànies, decisions que en l'apartat d'atzar s'atribuïa al segon tipus d'atzar. Per tant, els dos jocs tenen presència d'atzar del tipus 1 perquè es barregen les cartes materials, del tipus 2 perquè les decisions simultànies són el centre del joc, i per últim també atzar del tipus 3 perquè els jocs són prou complexos perquè tinguem limitacions a l'hora de crear estratègies.

Les normes del joc *Sushi Go!* es poden trobar a l'annex 5, però la part de les decisions simultànies és important que es remarqui. El funcionament és el següent:

1. Tots els jugadors agafen n cartes.
2. Cada jugador escull i juga una de les cartes simultàniament.
3. Els jugadors passen la seva mà de $n - 1$ cartes al jugador veí (en cada ronda/joc s'especifica en quin sentit).
4. Es repeteixen els passos 2 i 3 fins que s'arribi a un cert nombre de cartes per jugador. En el cas del *Sushi Go!*, s'acaba la ronda quan no queden cartes per jugar.

La relació dels dos jocs amb la combinatòria surt dels sistemes de puntuació. En els dos jocs, els jugadors obtenen punts segons les combinacions de cartes que han fet i que han aconseguit durant el procés de selecció de cartes de decisions simultànies.

4.3.2. Projecte: creació d'un algorisme

4.3.2.1. Objectiu

L'objectiu d'aquest projecte és la creació d'un algorisme mitjançant un programa informàtic que sigui capaç de jugar partides al joc de taula *Sushi Go!* utilitzant probabilitats, combinacions i permutacions. Per tant, el programa ha de ser capaç de rebre les cartes que té a la seva mà, i ser capaç de jugar la que li permet obtenir més punts.

4.3.2.2. Fonaments i estructura

L'algorisme s'ha realitzat utilitzant el llenguatge de programació C++, un llenguatge estàndard, orientat a objectes i genèric. Aquest programa utilitza també les funcions estàndards de programació, com *if*, *while*, *for* entre d'altres. També és possible crear funcions i el sistema de variables permet emmagatzemar nombres i caràcters de tot tipus.

L'estructura del programa es basa en un càlcul matemàtic basat en combinatòria i probabilitat. D'una manera simple, el programa guarda les cartes de la seva mà, que algú ha entrat al programa, i amb un seguit de càlculs tria la millor. Aquest procés es repeteix fins que no hi ha més cartes, per tant és capaç de jugar una ronda sencera del joc *Sushi Go!*.

Per triar la carta que li garantirà més punts, el programa planteja tots els resultats possibles de la ronda en el cas de tirar una carta que té a la mà. Per a cada carta, per tant, computa totes les possibilitats amb els punts que obtindria en cada resultat. Aquests punts es sumen i es divideixen pel nombre de resultats computats, calculant així una mitjana. Un cop ha acabat aquest procés amb cada carta que pot tirar de la seva mà, el programa resta amb una mitjana per a cada carta. La decisió de quina carta escollir la prendrà segons quina mitjana és més elevada.

Aquest procés es realitza mitjançant una sèrie de funcions en el programa les quals tenen unes feines específiques com fer la mitjana, calcular les possibilitats o sumar els punts per combinació.

Aquest mètode funciona bé, però hi ha un problema amb el nombre de resultats possibles. En el joc *Sushi Go!* hi ha 12 cartes diferents de les quals només utilitzem 7 (hi ha cinc cartes que impedeixen fer el programa utilitzant combinacions i permutacions, i es va optar per només jugar amb 7 cartes del joc), per tant el nombre de permutacions amb repeticions possibles per a cada nombre de cartes seran potències de 7 seguint la fórmula $PR_{n,k} = n^k$.

1	2	3	5	8	12
$1 \times 7^{1-1} = 1$	$2 \times 7^{2-1} = 14$	$3 \times 7^{3-1} = 147$	$5 \times 7^{5-1} = 12005$	$8 \times 7^{8-1} = 6588344$	$12 \times 7^{12-1} \approx 23728 \times 10^6$

FIGURA 32. Permutacions realitzades pel programa per un cert nombre de cartes.

El nombre de resultats és exponencial, el que significa que si el programa ha de prendre una decisió amb moltes cartes, tardarà massa temps. Aquest problema es va solucionar aplicant els següents canvis:

- **Cartes ja tirades:** el programa guarda les cartes que s'han jugat en cada ronda, tant les seves com les dels altres jugadors. D'aquesta manera a l'hora de triar una carta, les combinacions de cartes sempre inclouran les que ja ha tirat.
- **Recordar mans de cartes:** en aquest joc, les mans de cartes que es juguen van passant per tots els jugadors, en un únic sentit. Això causa que la mà de cartes que tenies a l'inici de la partida tornarà a ser la teva mà en algun moment de la partida, depenent del nombre de jugadors. Això permet reduir molt les permutacions de resultats. En una partida de dos jugadors i 10 cartes, només hi ha dues mans de cartes, que cada jugador tindrà 5 vegades durant la ronda. Si el jugador 1 comença amb la mà A, tindrà aquestes mans per cada selecció de cartes de la ronda:

1	2	3	4	5	6	7	8	9	10
A	B	A	B	A	B	A	B	A	B

FIGURA 33. Mà de cartes que utilitzarà qualsevol jugador d'una partida de dos jugadors per tirada de cartes.

Per tant, a l'hora de fer les permutacions de la primera selecció, el programa no pot provar totes les cartes a les rondes senars, ja que les cartes de les quals disposarà seran les que té en aquell moment menys algunes que s'hauran tirat abans. D'aquesta manera, només les cartes en posició parella de la primera selecció seran completament aleatòries des del punt de vista del programa. A partir de la segona selecció és només una qüestió d'atzar parcial i de probabilitats. Durant aquest procés, el programa també es fixa en les cartes que va provant en les zones senars per no repetir cartes: a l'hora d'escollir les cartes de la posició 3, no escollirà la carta que s'està provant a la casella 1. Quan hagi d'escollir la casella 5, no escollirà ni la carta en la posició 1 ni en la 3, etc.

- **Combinacions:** per últim també es va aplicar una solució per les seleccions amb més de 7 cartes on, fins i tot amb les altres mesures aplicades, el programa encara tardava un temps excessiu. En comptes d'utilitzar permutacions amb repeticions s'utilitzen combinacions amb repeticions, que funcionen igual que les permutacions, però no tenen en compte l'ordre dels elements. Per tant, els elements abc i cba són diferents permutacions, però la mateixa combinació. Per tant en comptes d'utilitzar la fórmula $PR_{n,k} = n^k$ tenim

$$CR_{n+r-1;k,n-1} = \frac{(n+k-1)!}{k!(n-1)!}$$

on n és igual al nombre total d'elements (nombre de cartes diferents que es fan servir) i k el nombre d'elements escollits per cada mostra (nombre de cartes per ronda).

L'estructura del programa es pot simplificar en el gràfic de la figura 35. Com es pot veure, tot l'algorisme es repeteix diverses vegades, ja que utilitza les mateixes instruccions per jugar totes les cartes d'una ronda, per tant el procés de rebre cartes, escollir-ne una i jugar-la es repeteix un nombre de vegades igual al nombre de cartes de la ronda. Ja que el programa és força complex, l'esquema no entra en detalls però totes les cartes es guarden en una matriu⁴⁶, on també es guarden cartes d'altres jugadors i s'hi indica quines cartes que s'han jugat, quines pot jugar el programa i quines s'estan provant a l'hora de fer les permutacions i combinacions.

Aquesta matriu i altres variables s'han de modificar constantment, i per això el primer bloc només es dedica a això. A continuació, el programa espera les cartes que té a la mà, que s'han d'entrar manualment. Aquest procés només es repeteix pel nombre de jugadors que hi ha a la partida, ja que després les mans ja han fet una volta sencera i el programa és capaç de recordar quines hi havia. A continuació, el programa realitza la selecció de cartes. Aquest procés és recursiu, i per tant no es pot representar de manera clara en un esquema, però cada bloc dintre del bloc "mitjanes" és una funció. Per tant, les funcions de combinacions i permutacions recursives són les encarregades de crear les *arrays* de variables que representen possibles resultats de les partides. Un cop un d'aquests resultats està calculat, es crida a la funció de contar punts perquè calculi els punts que s'obtidrien amb aquella combinació de cartes. Aquest procés es repeteix per a cada resultat possible que s'hagi de calcular. Tot seguit, un cop el programa té totes les mitjanes, tria la carta amb la mitjana més alta, i l'escriu cridant una altra funció. Per últim, el programa espera l'entrada de les cartes jugades pels altres jugadors.

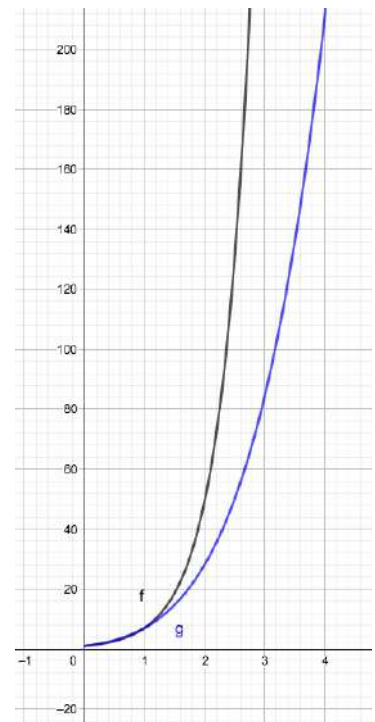


FIGURA 34. Funció de les permutacions (f) i de les combinacions (g) on el nombre cartes diferents és igual a 7 i l'eix x és el nombre de cartes a escollir. Eixos a escala 1:20. FONT. Elaboració pròpia mitjançant *Geogebra*: *Calculadora Gràfica*.

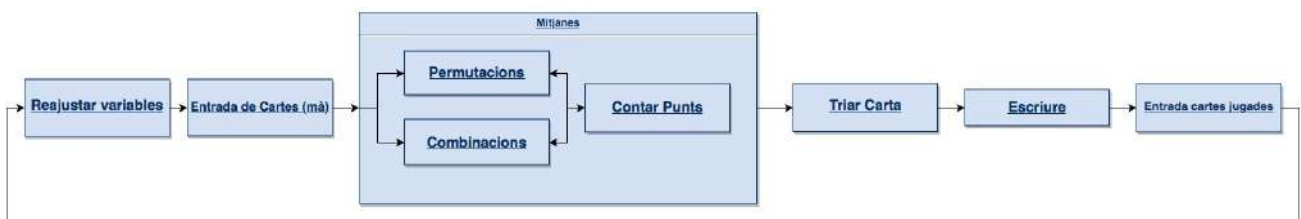


FIGURA 35. Esquema del funcionament del programa sg5. FONT. Elaboració pròpia mitjançant *draw.io*: plataforma oberta per crear i compartir diagrames.

⁴⁶ Conjunt d'*arrays*, o dit d'altra manera conjunt de variables numèriques emmagatzemades de manera quadridimensional.

4.3.2.3. Producte final i funcionalitat

La versió final del programa (annex 6) és capaç de jugar rondes individuals de partides del joc *Sushi Go!* amb 7 de les 12 cartes del joc. El programa permet jugar partides de màxim 5 jugadors i màxim 10 cartes per mà seguint les normes del joc. Tot i això, a causa de com el programa fa servir la recursivitat i les funcions potencials, no sempre és capaç de donar una resposta amb un temps suficientment curt.

Aquest aspecte es pot veure en la representació gràfica de la figura 36, on l'eix d'abscisses correspon al nombre de jugadors de la partida, l'eix d'ordenades a les cartes de la partida i l'eix vertical el temps mesurat en segons, expressat amb una escala logarítmica. El temps mesurat equival al temps que tarda el programa sg5 a retornar la primera resposta d'una ronda de *Sushi Go!* per a un cert nombre de jugadors i cartes (totes diferents). Com es pot observar el programa no es pot executar a partir de les 10-9 cartes, ja que tardaria un temps superior a una hora.

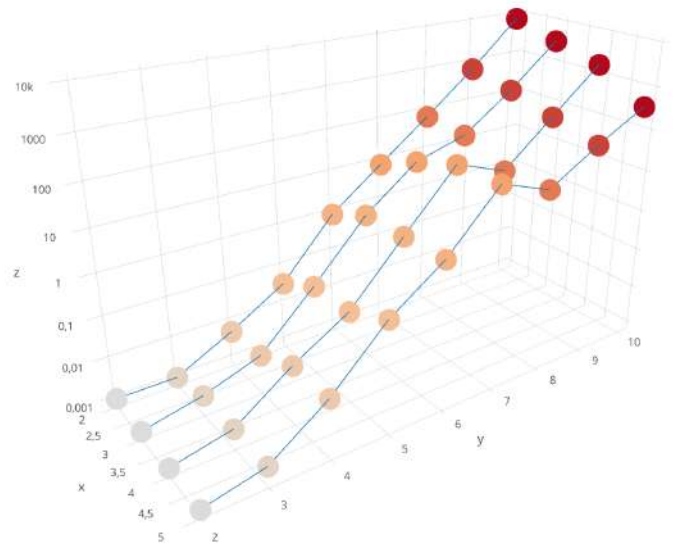


FIGURA 36. Gràfic en 3 dimensions del temps que tarda a respondre el programa sg5. FONT. Elaboració pròpia mitjançant *plotly*. Visualització moderna per l'era de les dades.

També podem observar en el gràfic l'estalvi de temps que sorgeix de la utilització de les combinacions en les rondes amb més de 7 cartes.

4.3.2.4. Possibles canvis i millores

Com s'ha dit abans, el programa funciona però té certs defectes que es podrien millorar. Un d'aquests aspectes és l'estratègia que utilitza. El programa es basa únicament a mirar mitjanes de puntuacions, això li assegura uns certs punts, però no sempre li assegura la victòria o la màxima puntuació. Això també té a veure en el fet que el programa, tot i que té la informació de totes les cartes que estan en joc, pren les seves decisions tenint en compte només les cartes que ha jugat ell, les que té en aquest moment i les que rebrà en les rondes per venir. Amb aquest fet volem remarcar que també podria prendre decisions mirant els punts que no aconseguiren altres jugadors en comptes dels que pot aconseguir ell. En alguns casos de

combinacions que si no es completen, no donen cap percentatge de punts, aquestes accions es poden realitzar involuntàriament per jugadors humans. Aquest petit detall elevaria molt, segons la nostra opinió, el nivell del programa. Això es podria aconseguir pensant detalladament una altra estratègia i aplicar-la al codi o dissenyar una xarxa neuronal⁴⁷. En tots els casos, però, s'hauria de modificar gran part de l'estructura actual del programa.

Un altre aspecte a desitjar, és el fet que pugui jugar amb totes les cartes i no només amb 7 de les 12 que té el joc. Però altra vegada, i com també passaria si es vol millorar la velocitat del programa, es necessitaria fer un canvi en l'estructura del programa.

⁴⁷ Model matemàtic o de programació/computació basat en la interacció de neurones artificials, simulant una xarxa neuronal biològica.

5. Conclusions

Al llarg d'aquest treball, he explorat diversos temes que relacionen el món del joc i les matemàtiques tal com s'havia plantejat. Els dos objectius del treball també s'han pogut assolir.

Primer de tot s'han pogut realitzar satisfactòriament els dos projectes de la creació del web i de l'algorisme. Pel que fa al web, s'ha pogut desenvolupar una plataforma virtual i en línia, amb un sistema d'usuaris per permetre a jugadors, crear-se un perfil, entrar resultats de les seves partides i poder obtenir certes anàlisis i resultats. Aquest també permet manipular els llistats de jugadors, jocs i marques, personalitzats per a cada usuari. Tot això s'ha aconseguit després d'un procés d'autoaprenentatge dels fonaments de les pàgines web, tant dels llenguatges de programació com de l'estructura en si. Per tant, tot i que el web té diversos aspectes a millorar, el resultat és un web funcional i pràctic.

Per altra banda, l'algorisme s'ha realitzat usant el llenguatge de programació C++, i els seus resultats són positius, ja que, segons les mitjanes proporcionades pel web, és capaç de guanyar partides al joc de cartes adaptat *Sushi Go!* un 44.86% de les vegades (de 107 partides jugades). El web també dona dades més detallades de les partides de l'algorisme, on es pot veure que en totes les partides (de diferent nombre de jugadors i de cartes) amb una mostra significativa té una mitjana de puntuació per ronda més elevada que la mitjana general. Per altra banda, també destacar que el percentatge de victòries de l'algorisme en les diferents partides de diferent nombre de cartes s'acosta sempre al percentatge global, amb una tendència a l'alça quan hi ha poques cartes. Això es causa, en la major part, de l'atzar.

Tot i això, aquests resultats es poden interpretar de dues maneres diferents. Una de les interpretacions és que el programa té una estratègia molt bona, però els resultats no sempre són victoriosos perquè hi ha una gran presència d'atzar en el joc. L'altre és que el joc no té presència d'atzar i els resultats negatius del programa són causats per la seva estratègia que no és suficientment bona, o que comparada amb la dels seus adversaris és pitjor. Des del meu punt de vista la interpretació correcta es troba entre les dues esmentades, ja que el joc té una presència d'atzar considerable i el programa té diversos errors o imperfeccions.

En segon lloc i respecte al segon objectiu, s'han pogut observar en detall altres aspectes dels jocs de taula que es relacionen amb les matemàtiques. Després d'entendre que significava el concepte joc, i comprendre perquè els creem i utilitzem, n'hem pogut observar 3 en diferents projectes. Aquests han servit per veure com branques de la matemàtica com la geometria, la teoria de conjunts o la probabilitat

són presents en els jocs de taula, d'una manera quasi obligatòria. Cada una d'aquestes branques l'hem aplicat en un joc de taula, del qual n'hem extret unes dades, relacions o estratègies utilitzant els conceptes matemàtics. D'aquests projectes també hem pogut entendre la importància de la computació en el món de les matemàtiques. Aquesta eina tan important ens ha servit per trobar resultats, que a causa de l'atzar en certs jocs, haguessin estat molt més llargs i laboriosos. També s'ha aprofundit en el concepte abstracte de l'atzar, dividint-lo en tres tipus diferenciats.

Per últim, aquest treball també ens ha servit per entendre que quantificar l'atzar o la sort d'un joc, com es volia aconseguir en una primera versió del treball, no és tant fàcil com pot semblar. De moment, aquest càlcul no és possible de realitzar i pot ser que no es pugui acabar realitzant mai d'una manera conceptual o teòrica, i que, en el futur, s'hagi de fer de manera empírica.

Bibliografia i webgrafia

AJ. (Juliol 2017). HTML5 UP. Recuperat de <https://html5up.net>

Albertí, M. (2014). El mosaico sin fin, Teselaciones y dibujos sobre el plano (1a ed.). Espanya: S.A.U.

Atzar. (2007). Dins *Diccionari de la llengua catalana*. (2a ed.). Recuperat de <https://mdlc.iec.cat/results.asp?txtEntrada=atzar&operEntrada=0>

Cambell, Paul J. i Chavey, Darrah P. (1995). *Tchuka ruma solitarie*. Recuperat de <https://www.beloit.edu/computerscience/assets/tchuka.pdf>

Crawford, C. (Agost 2018). *My definition of 'Game'*. Recuperat de <http://www.erasmatazz.com/library/the-journal-of-computer/jcgd-volume-4/my-definition-of-game.html>

Crawford, C. (2000). *Chris Crawford on Game Design*. Recuperat de https://www.digitpress.com/library/books/book_art_of_computer_game_design.pdf

Cross, T. (2017). Welcome to the golden age of board games - brought to you by the internet. *Financial Review: News Website of the Year*, 1-1. Recuperat de <https://www.afr.com/lifestyle/welcome-to-the-golden-age-of-board-games--brought-to-you-by-the-internet-20171219-h07a36>

Csíkszentmihályi, M. (1990). *Flow: The Psychology of Optimal Experience*. Recuperat de <https://www.pdfdrive.com/flow-psychology-of-optimal-experience-d19521954.html>

David F. (2013). An Alternative Classification of Board Games(long). *BoardGameGeek Forums: Board Game Design*, 1-1. Recuperat de <https://boardgamegeek.com/thread/581158/alternative-classification-board-games-long>

Fandom Wiki. (Agost 2018). *Tchuka ruma*. Recuperat de http://mancala.wikia.com/wiki/Tchuka_Ruma

Gilbert, Daniel. (2016). Luck and Skill in Modern Board Games. *Board Game Design*, 1-1. Recuperat de <https://boardgamegeek.com/blogpost/60844/luck-and-skill-modern-board-games>

Hejdenberg, A. (2005). The Psychology Behind Games. *Gamasutra: The Art & Business of Making Games*, 326190(130702), 1-2. Recuperat de http://www.gamasutra.com/view/feature/2289/the_psychology_behind_games.php

Hillengas, Zach. (2015). 6 differences between designer games and mass market games. *Board Game Resource: Blog*. 1-1. Recuperat de <http://www.boardgamerresource.com/designer-board-games-vs-mass-market>

Ibáñez, R. (2013). Tchuka Ruma, el mancala solitario. *Cultura Científica, Matemoción*. 1-1. Recuperat de <https://culturacientifica.com/2013/11/20/tchuka-ruma-el-mancala-solitario/>

Ibañez, R. (2015). Del ajedrez a los grafos, La seriedad matemática de los juegos (1a ed.). Espanya: S.A.U.

IGDA Denmark. (Agost 2018). Richard Garfield - "Luck in Games" talk at ITU Copenhagen [Vídeo]. Recuperat de <https://www.youtube.com/watch?v=av5Hf7uOu-o>

infamouswhiteknight. (2013). Guest Contribution: The Most Important Pieces and the Problem with the Z5 (according to Mike_Yosuke). *Blokus Strategy: Advanced*. 2 (1), 1-1. Recuperat de blokusstrategy.com/guest-contribution-the-most-important-pieces-and-the-problem-with-the-z5-according-to-mike_yosuke/

Jamieson, Darran. (2014). A Look at Luck in Game Design. *Game development: Game design*, 1-1. Recuperat de <https://gamedevelopment.tutsplus.com/articles/a-look-at-luck-in-game-design--gamedev-14195>

Khan Academy. (2017-2018). *Khan Academy*. Recuperat de <https://www.khanacademy.org>

Luck. (1999). Dins Cambridge Dictionary. (s.d.). Recuperat de <https://dictionary.cambridge.org/dictionary/english/luck>

Luck. (s.d.). Dins Oxford Living Dictionaries. (s.d.). Recuperat de <https://en.oxforddictionaries.com/definition/luck>

Maslow, Abraham. (1954). *Motivation and Personality: A general theory of human motivation based upon a synthesis primarily of holistic and dynamic principles*. Recuperat de http://s-f-walker.org.uk/pubsebooks/pdfs/Motivation_and_Personality-Maslow.pdf

MDN web docs. (Agost 2018). *CSS valors i unitats*. Recuperat de https://developer.mozilla.org/ca/docs/Learn/CSS/Introducci%C3%B3_a_CSS/Valors_i_unitats

Northwestern University Mathematics Department. (s.d.). *Generalized Permutations and Combinations*. Recuperat de <https://sites.math.northwestern.edu/~mlerma/courses/cs310-05s/notes/dm-gcomb>

Randomness. (1999). Dins Cambridge Dictionary. (s.d.). Recuperat de <https://dictionary.cambridge.org/dictionary/english/randomness>

Randomness. (s.d.). Dins Oxford Living Dictionaries. (s.d.). Recuperat de <https://en.oxforddictionaries.com/definition/randomness>

Recursiu. (2007). Dins *Diccionari de la llengua catalana*. (2a ed.). Recuperat de <https://mdlc.iec.cat/results.asp?txtEntrada=recursiu&operEntrada=0>

Rizzo, S. (2014). The Tabletop Renaissance: How Board Games are Making a Comeback. The HUB, 1-1. Recuperat de <https://hub.fullsail.edu/articles/the-tabletop-renaissance-how-board-games-are-making-a-comeback>

Robert, K., Garfield, R. i Elias, S. (2012). *Characteristics of Games*. Extret de l'article *Luck and Skill in Modern Board Games*, recuperat de <https://boardgamegeek.com/blogpost/60844/luck-and-skill-modern-board-games>

Sort. (2007). Dins *Diccionari de la llengua catalana*. (2a ed.). Recuperat de <https://mdlc.iec.cat/results.asp?txtEntrada=sort&operEntrada=0>

Stack Exchange. (Setembre 2018). Mathematics - *In the card game Set, what's the probability of a Set existing in n cards?*. Recuperat de <https://math.stackexchange.com/questions/202862/in-the-card-game-set-whats-the-probability-of-a-set-existing-in-n-cards/203146#203146>

Stack Exchange Inc. (2017-2018). *stackoverflow*. Recuperat de <https://stackoverflow.com>

Stanford Computer Science - Donalde E Knuth. (Agost-Setembre 2018). Programs to Read - SETSET. Recuperat de <https://cs.stanford.edu/~knuth/programs/setset.w>

The PHP Group. (2017-2018). *PHP*. Recuperat de <http://php.net>

Wikipedia. (Agost 2018). *Board game*. Recuperat de https://en.m.wikipedia.org/wiki/Board_game

Wikipedia. (Agost 2018). *Cascading Style Sheets*. Recuperat de https://en.wikipedia.org/wiki/Cascading_Style_Sheets

Wikipedia. (Setembre 2018). *Combinatòria*. Recuperat de <https://ca.wikipedia.org/wiki/Combinatòria>

Wikipedia. (Agost 2018). *Eurogame*. Recuperat de <https://en.m.wikipedia.org/wiki/Eurogame>

Wikipedia. (Agost 2018). *HTML*. recuperat de <https://es.wikipedia.org/wiki/HTML>

Wikipedia. (Agost 2018). *JavaScript*. Recuperat de <https://en.wikipedia.org/wiki/JavaScript>

Wikipedia. (Agost 2018). *Mancala*. Recuperat de <https://en.m.wikipedia.org/wiki/Eurogame>

Wikipedia. (Agost 2018). *PHP*. Recuperat de <https://en.wikipedia.org/wiki/PHP>

Wikipedia. (Setembre 2018). *Polyominol*. Recuperat de <https://en.wikipedia.org/wiki/Polyomino>

Wikipedia. (Agost 2018). *SQL*. Recuperat de <https://en.wikipedia.org/wiki/SQL>

w3schools. (2017-2018). *w3schools*. Recuperat de <https://www.w3schools.com>

w3schools. (Agost 2018). *CSS boxmodel*. Recuperat de https://www.w3schools.com/css/css_boxmodel.asp

Annexos

Annex I: normas del joc Set

INSTRUCCIONES

ESP

El objetivo del juego es identificar un **SET** de 3 cartas de entre las 12 colocadas boca arriba en la mesa. Cada carta tiene cuatro características, que varían como sigue:

- (A) **SÍMBOLOS:** cada carta contiene óvalos, ondas o rombos;
- (B) **COLORES:** los símbolos son rojos, verdes o lilas;
- (C) **NÚMERO:** cada carta contiene uno, dos o tres símbolos;
- (D) **FONDO:** los símbolos son sólidos, rayados o sin fondo.

Un **SET** consiste en 3 cartas en las que las características, evaluadas una a una, son iguales en cada carta o diferentes en todas ellas. Todas las características por separado deben satisfacer la regla. En otras palabras: la forma debe ser igual en las 3 cartas, o diferente en cada una de ellas; el color debe ser el mismo en las 3 cartas, o diferente en cada una de ellas, etc. Véase el apartado EJEMPLOS en la siguiente página.

COMPROBACIÓN RÁPIDA - ¿ES UN SET?

Si 2 son iguales y 1 es distinta en alguna característica, entonces no es un **SET**. Por ejemplo, si hay 2 cartas rojas y 1 lila, entonces no es un **SET**. Un **SET** debe tener o todo igual o todo distinto para cada una de las características.

INICIO RÁPIDO

Para una rápida introducción, empezad con la baraja pequeña (solo con los símbolos sólidos); esto elimina una característica: el fondo. Cuando ya veáis **SETs** fácilmente al jugar con la versión de 3 características, mezclad las dos barajas juntas para así jugar con el mazo completo.

EL JUEGO

Uno de los jugadores, el repartidor, baraja bien las cartas y coloca 12 boca arriba en la mesa (formando un rectángulo de 4 x 3 cartas) de modo que puedan ser vistas por todos.

- No hay turnos; el primer jugador en distinguir un **SET** y anunciarlo, gritando "**SET**", detiene el juego.
- Tras haber anunciado **SET**, ningún otro jugador podrá anunciar otro **SET** hasta que se haya resuelto el **SET** en curso.
- Cada **SET** debe ser comprobado por los otros jugadores.
- Si es correcto, el jugador que lo vio se queda dichas cartas y se colocan 3 nuevas, del mazo, en la mesa.
- Si un jugador grita **SET** y se ha confundido o hay un error, devolverá el **SET** anunciado a la mesa y perderá 3 de sus cartas ganadas anteriormente.
- Si en algún momento de la partida todos los jugadores coinciden en que no hay ningún **SET** entre las 12 cartas de la mesa, se colocarán 3 nuevas cartas (para tener un total de 15). Las 3 cartas no se reemplazarán cuando el siguiente **SET** sea retirado de la mesa, volviendo así a las 12 iniciales del juego.

El juego continúa hasta que se acaba el mazo. Al final del juego pueden quedar cartas en la mesa que no formen ningún **SET**. El ganador es el jugador que haya conseguido más cartas.

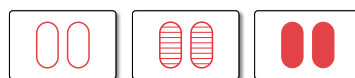
Nota: con 12 cartas sobre la mesa hay un 97% de probabilidades de que exista un **SET**. Con 15 cartas, la probabilidad aumenta hasta el 99,96%. Podéis encontrar más información sobre matemáticas y **SET** en nuestra web (www.devir.es).

VARIANTE

Si queréis un juego más largo, el repartidor pasará el mazo al jugador de la izquierda, quien se convertirá en el nuevo repartidor. Cuando todos los jugadores lo hayan sido, la partida termina. El jugador que haya conseguido más cartas en total (es recomendable apuntar los resultados de cada ronda), gana la partida.

EJEMPLOS

Los siguientes ejemplos son **SETs**:



Las tres cartas son del mismo color; las tres cartas tienen la misma forma; las tres cartas coinciden en número; y las tres cartas tienen fondos distintos.



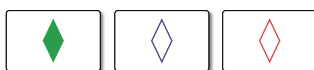
Las tres cartas son de diferente color; las tres cartas tienen formas diferentes; las tres cartas tienen diferente número de elementos; y las tres cartas coinciden en fondo.



Las tres cartas son de diferente color; las tres cartas tienen formas diferentes; las tres cartas tienen diferente número de elementos; y las tres cartas tienen diferente fondo.

2

Los siguientes ejemplos **NO** son **SETs**:



Las tres cartas son de diferente color; las tres cartas tienen la misma forma; las tres cartas coinciden en número; pero dos cartas tienen el mismo fondo y la otra carta un fondo distinto.



Las tres cartas tienen la misma forma; las tres cartas coinciden en número; las tres cartas tienen diferente fondo; pero dos cartas son del mismo color y la otra carta es de un color distinto.

PREMIOS OBTENIDOS

SET es uno de los juegos más premiados de todos los tiempos. Ha ganado más de 25 de los mejores premios, entre los cuales el: MENSA Select Award, Dr. Toy's 10 Best Games Award, Creative Child's Preferred Choice Award, Games Magazine "Games 100 Award". En www.devir.es encontraréis más información.



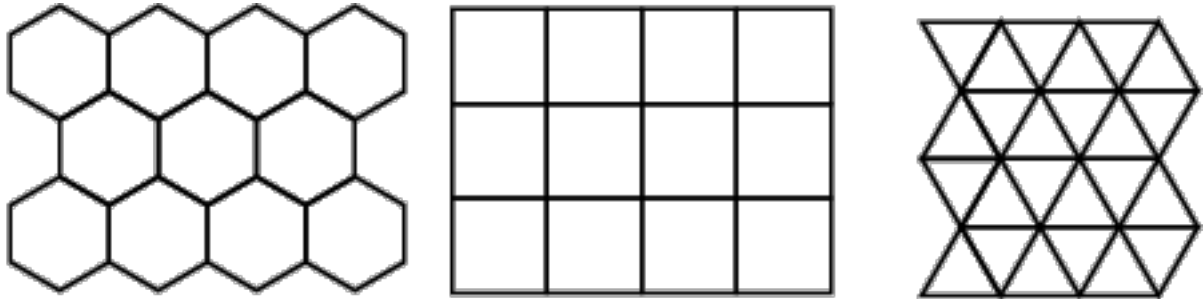
Revisión del texto:
Marc Figueras y
María Pitarque
Adaptación gráfica:
Basco



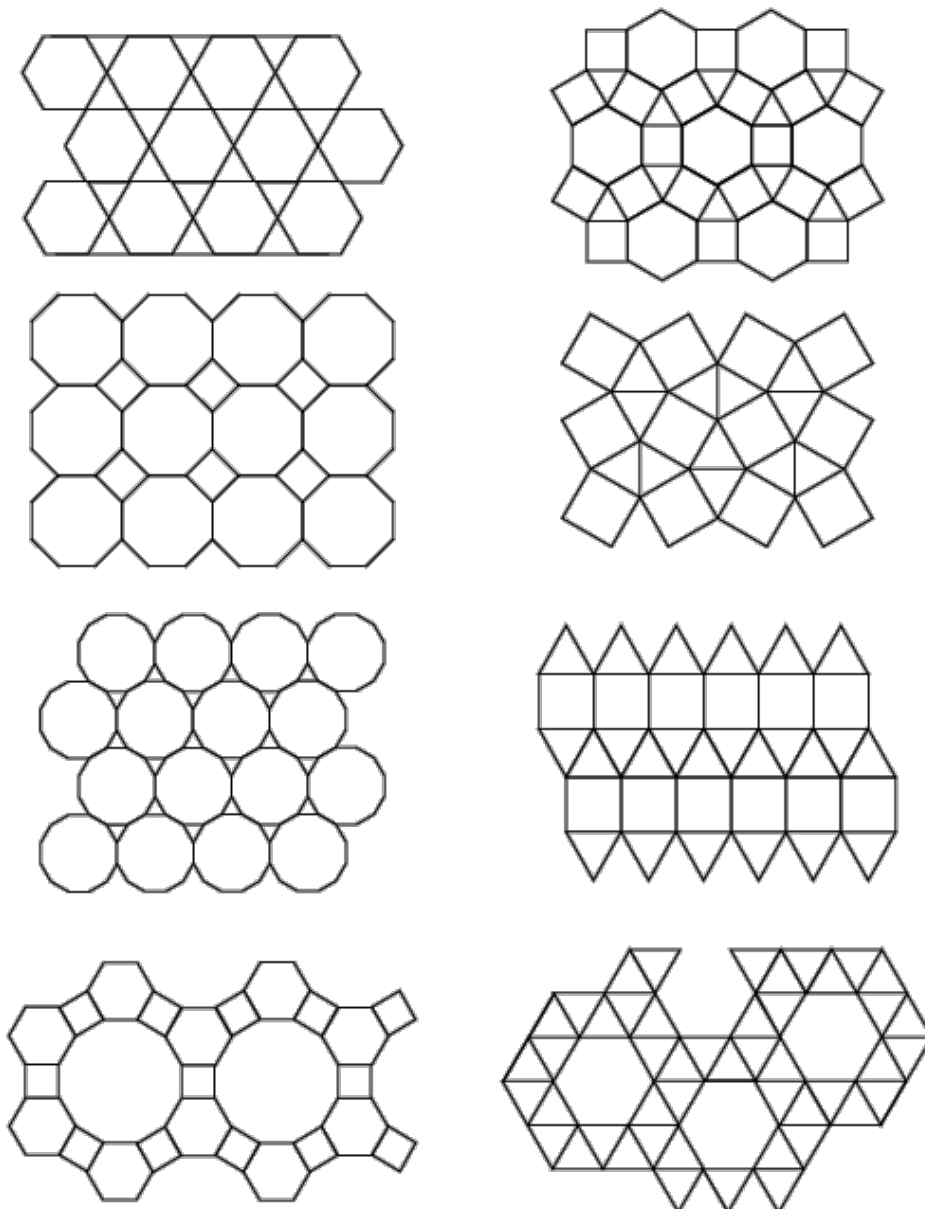
SET – El juego familiar de percepción visual
Edad: a partir de 6 años
Número de jugadores: 1 o más
© www.SETgame.com

Annex II: tessellacions

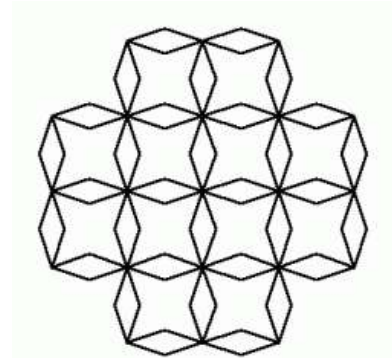
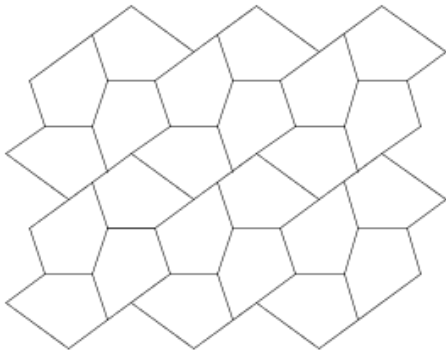
Tessellacions regulars:



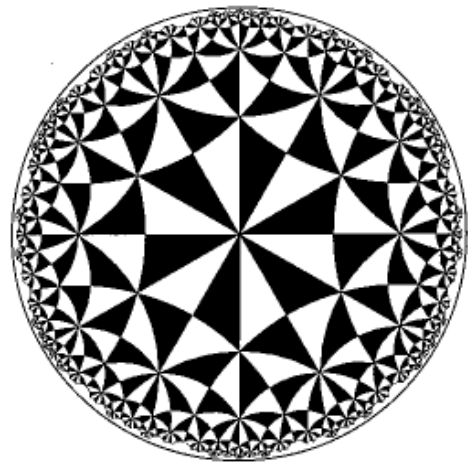
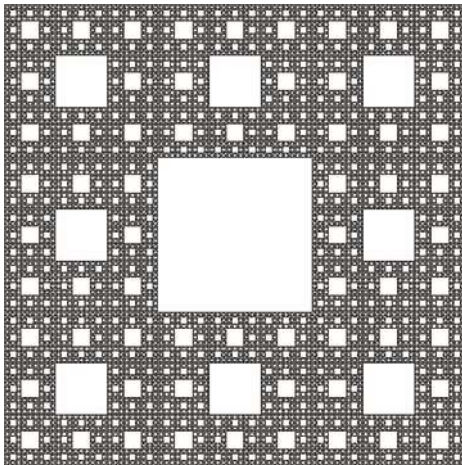
Tessellacions semiregulars:



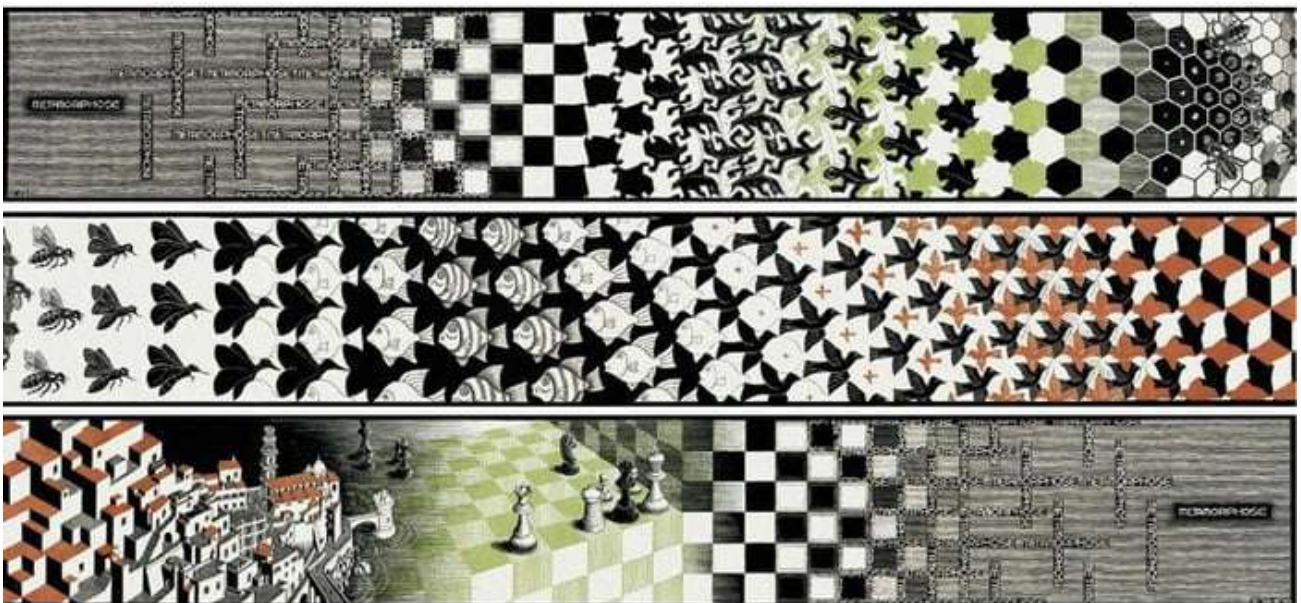
Exemples de tessellacions irregulars:



Exemples de tessellacions fractals, en espai finit en forma de quadrat i de cercle:



Metamorfosis II d'Escher:



Annex III: normes del joc *Blokus*



BLOKUS i sencillament apassionante !

- 1 Material**
 - El juego se compone de los siguientes elementos:
 - un tablero con 400 casillas.
 - 84 piezas (21 piezas de cada uno de los cuatro colores).
 - Cada una de las 21 piezas tiene una forma diferente.
 - Hay 1 pieza de un cuadrado, 1 pieza de dos cuadrados, 2 piezas de tres cuadrados, 5 piezas de cuatro cuadrados y 12 piezas de cinco cuadrados. (Imagen n.º 1).

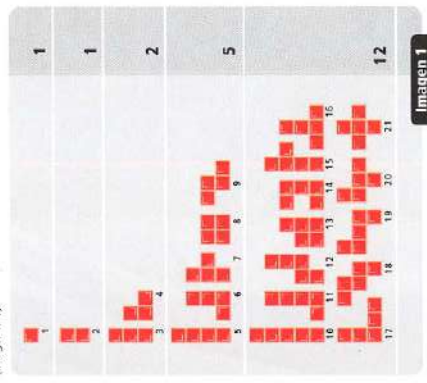
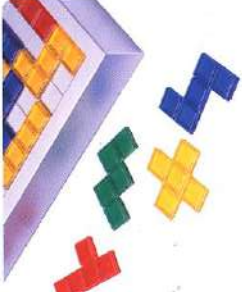
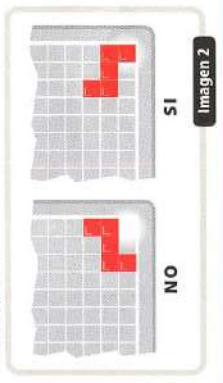


Imagen 1
Cada jugador deberá colocar sus 21 piezas sobre el tablero.

- 2 Objetivo del juego**
Cada jugador deberá colocar sus 21 piezas sobre el tablero.
- 3 Desarrollo de la partida**
El orden en el que se juega es el siguiente: azul, amarillo, rojo y verde.
Los jugadores, cubren a suertes que color les corresponde y cada uno coloca las 21 piezas del juego. El primer jugador coloca la primera pieza de su elección sobre el tablero, de la forma que ocupa una de las casillas en ángulo. (Imagen n.º 2).



Los otros jugadores participarán según les toque el turno y colocarán su primera pieza de la misma forma.
En las siguientes vueltas, cada nueva pieza que se coloque debe tocar una del mismo color por una o varias esquinas, pero nunca por los lados. (Imagen n.º 3).

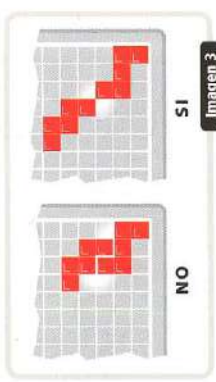


Imagen 3
Por el contrario, las piezas de diferente color pueden tocarse por los lados. (Imagen n.º 4).

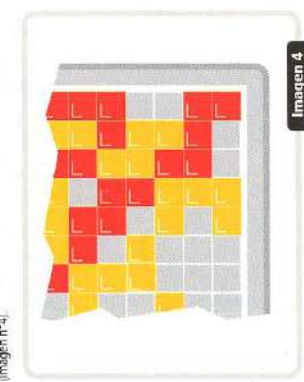


Imagen 4
Las piezas del juego pueden colocarse sobre el tablero en cualquier sentido. Una pieza colocada en el tablero continúa en su sitio hasta el final de la partida.

- 4 Fin de la partida**
Cuando un jugador quede bloqueado y ya no pueda colocar ninguna pieza, deberá pasar el turno. Los otros jugadores continuarán conservando el mismo orden de juego. Cuando todos los jugadores quedan bloqueados cada uno contará el número de cuadrados que no haya podido colocar sobre el tablero y escribirá su puntuación.
-1 punto por cada cuadrado que no se haya colocado.
+15 puntos si se han colocado las 21 piezas.
+20 puntos si se han colocado las 21 piezas con el cuadrado que queda suelto en última posición.

La imagen n.º 5 representa una partida terminada.

- el jugador que lleva las piezas azules ha conseguido colocar todas sus piezas con el color que queda suelto en último lugar: +20 puntos
- el jugador que tiene las piezas amarillas no ha podido colocar 2 piezas de cuatro cuadrados: -8 puntos
- el jugador con las piezas rojas no ha podido colocar 1 pieza de tres cuadrados, 4 piezas de cuatro cuadrados y 1 pieza de cinco cuadrados: -24 puntos
- el jugador que tiene las piezas verdes no ha podido colocar 1 pieza de tres cuadrados, 3 piezas de cuatro cuadrados y una pieza de cinco cuadrados: -20 puntos

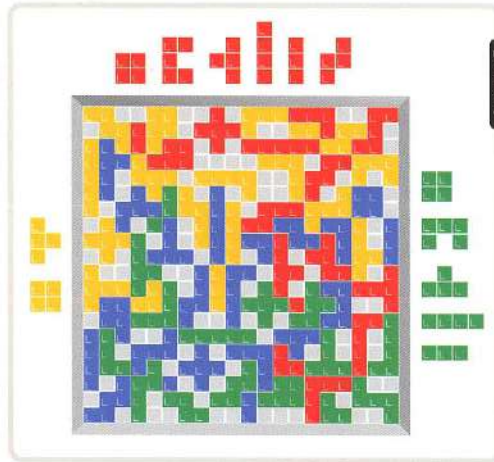


Imagen 5

- 5 Duración de la partida**
Habitualmente, las partidas duran entre 20 y 30 minutos.
- 6 Variantes del juego**
a) Juego en dos equipos de dos jugadores cada uno
El orden en que se juega es siempre el siguiente: azul, amarillo, rojo y verde.
Los jugadores situados en diagonal sobre el tablero forman equipo.
Al final de la partida, se suma la puntuación de los jugadores azul y rojo y la de los jugadores amarillo y verde y el equipo que obtenga mejor puntuación gana la partida.

- b) Juego con dos jugadores**
El orden en que se juega es siempre el siguiente: azul, amarillo, rojo y verde.
El mismo jugador lleva las piezas azules y rojas y juega contra el que lleva las piezas amarillas y verdes. Se suma la puntuación de los colores azul y rojo y se compara con la que han obtenido los colores amarillo y verde.
c) Juego con tres jugadores
Los tres jugadores ocupan una esquina y cada uno juega para el mismo. El primer jugador comienza colocando una pieza por cada uno de las tres esquinas. En el resto, el orden sobre el color es rotatorio. Al final que en una partida de cuatro jugadores el ganador será aquel que haya podido colocar el máximo de cuadrados sobre el tablero, sin tener en cuenta la puntuación obtenida con el color « relativo ».

d) Juego con un solo jugador
El juego permite proponer varias versiones de despedacadas, tales como colocar sobre el tablero el conjunto de las 84 piezas observando todas las regiones del juego.

Consejos para jugar:
• Al principio de la partida ir avanzando progresivamente hacia el centro del tablero y así intentar ocupar el máximo espacio posible. Si se queda por la zona de la que ha empezado será rápidamente bloqueado. Inicente colocar, al principio, las piezas más grandes porque sino al final del juego solo quedarán espacios reducidos y será más difícil colocarlas.

- Aprenda a utilizar cada pieza: son todas diferentes, aunque el mayor partido a sus diferentes formas.
- Intente conservar durante toda la partida una o más posibles salidas de salida para no quedarse bloqueado.
- Juegue en función de las piezas que le quedan pero también teniendo en cuenta las piezas que le quedan a sus oponentes los.
- Si consigue colocar todas las piezas de un mismo color, suma 15 puntos y 20 puntos si finaliza colocando la pieza de un solo cuadrado.

Adquiere sin demora para jugar, a la dirección www.blokus.com
Para usted competir como el ordenador, obtener consejos acerca del juego, intercambiar piezas de recambio, citarse con otros jugadores e integrarse en la clasificación mundial.

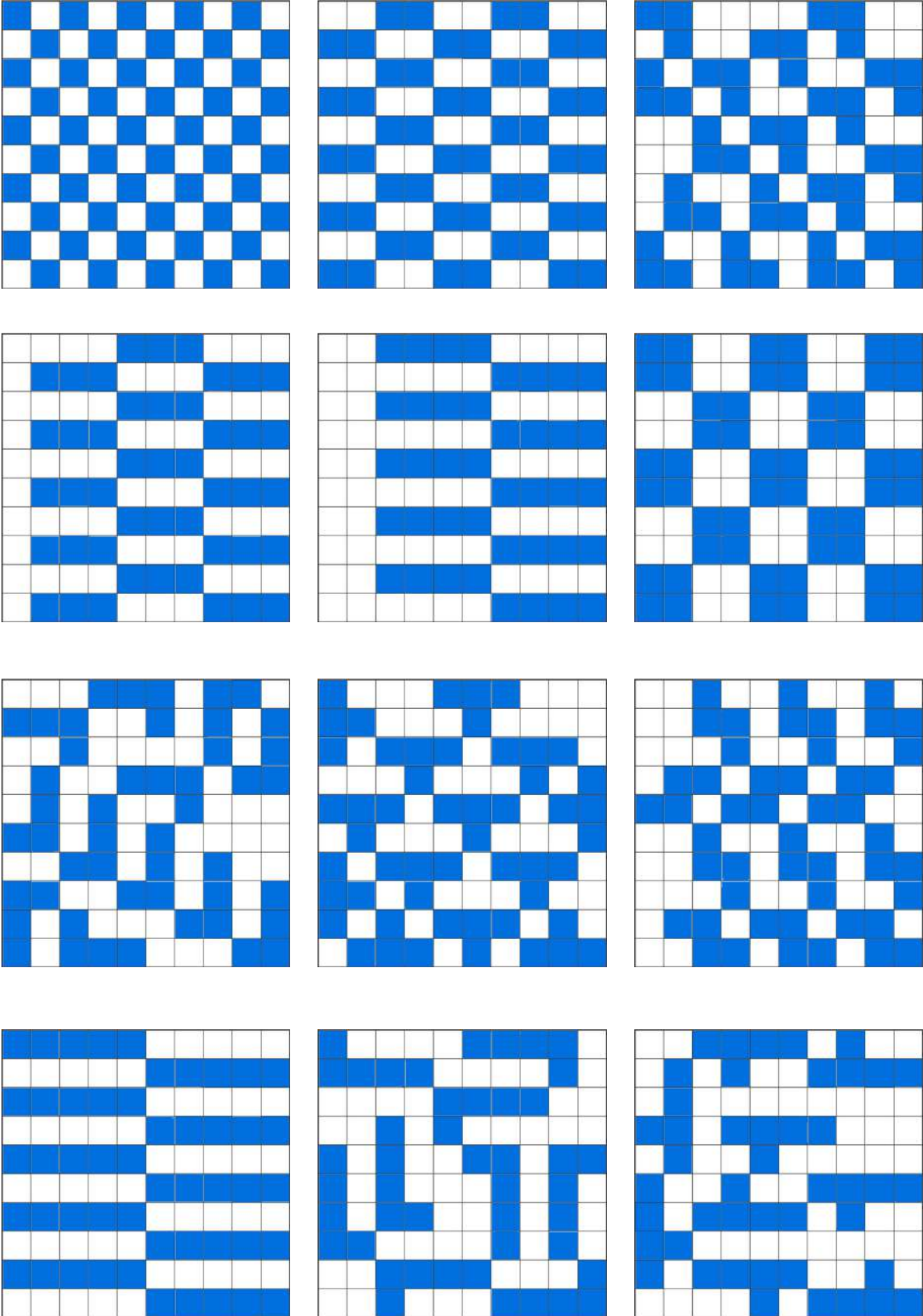
Hemos cuidado hasta el último detalle en la elaboración de nuestro juego **Blokus**. Sin embargo, si constata un defecto de fabricación o la falta de una pieza, puede contactar con el distribuidor de juego cuyo dirección figura en el dorso de la caja.
Blokus. Según una idea original de Bernard Touban.
Todos los marcas mencionadas en este manual son marcas registradas. La imagen n.º 5 representa una partida terminada.

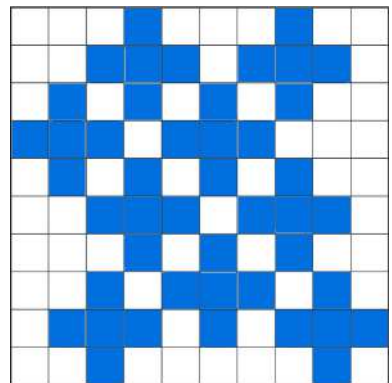
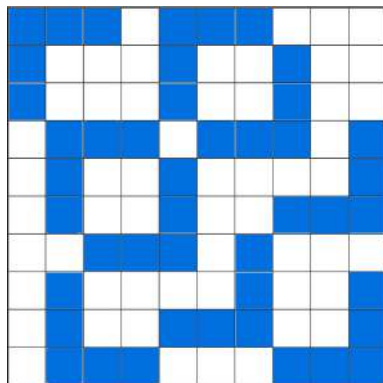
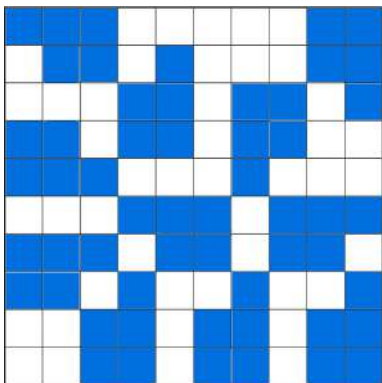
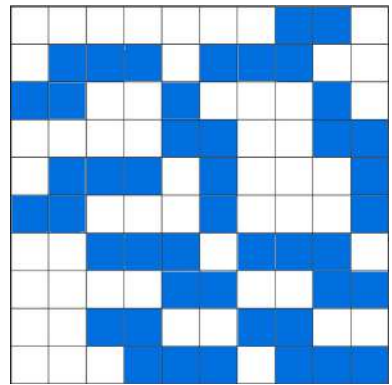
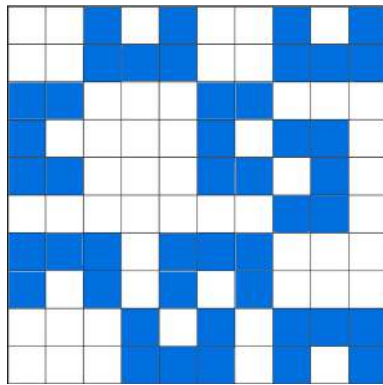
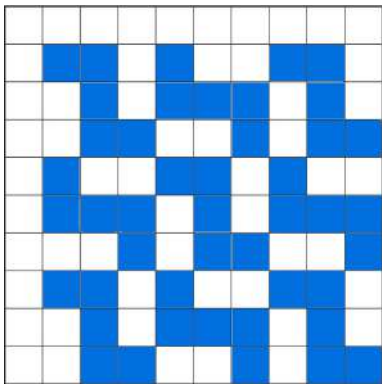
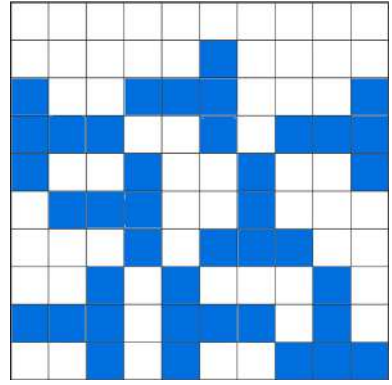
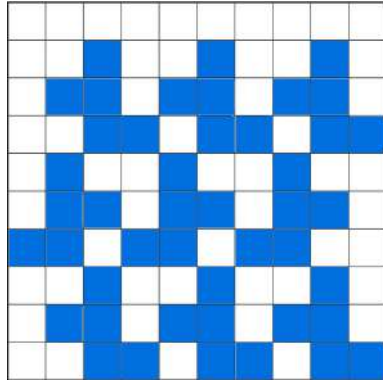
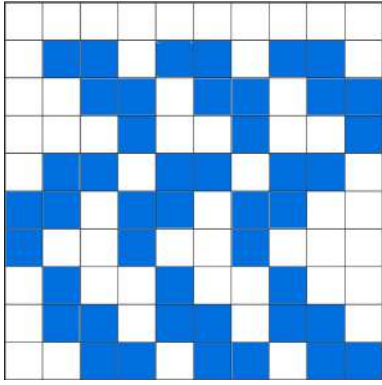
Ravensburger Iberica S.L.U.
C/ Lopez de Hoyos nº 327
Planta 5ª - 3
Edificio Pinar de Arturo Soria
28043 Madrid
España
Tel: +34 91 748 91 92
Fax: +34 91 388 55 60

404001800E Ref: RC034 RUL060502 rec Printed in China

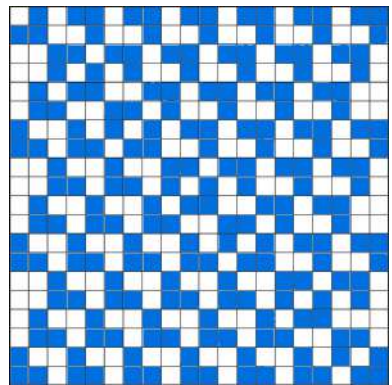
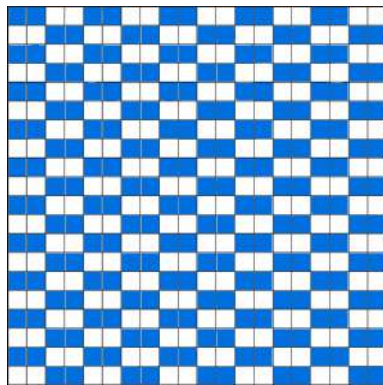
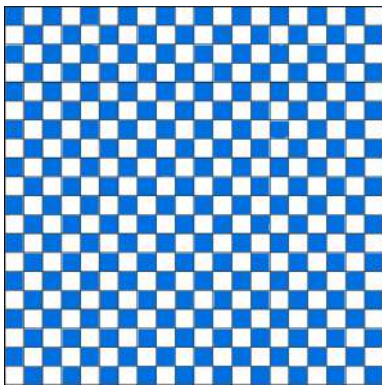
Annex IV: tessellacions amb els poliòminos del joc *Blokus*:

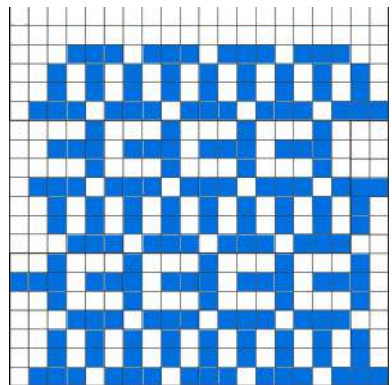
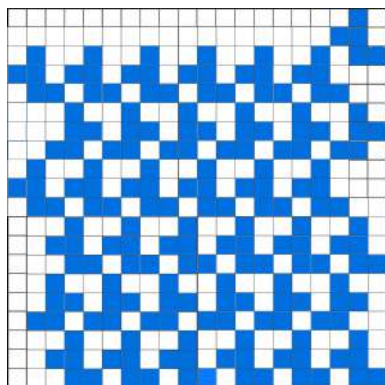
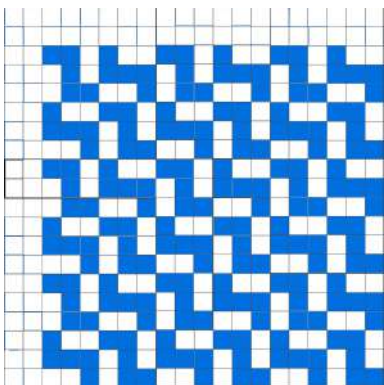
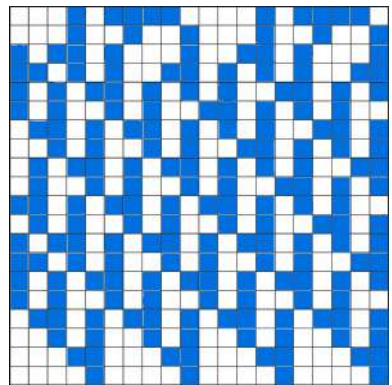
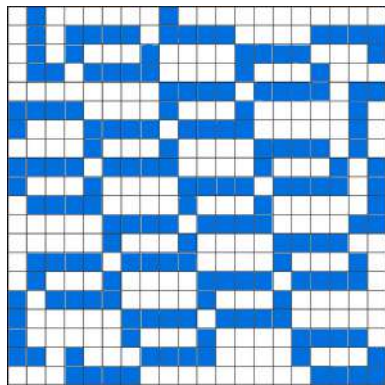
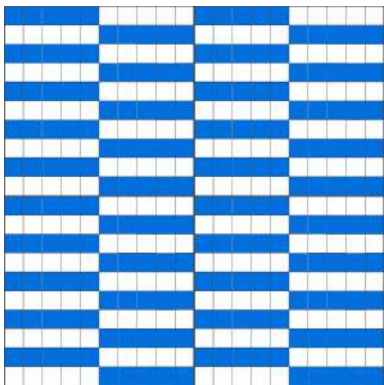
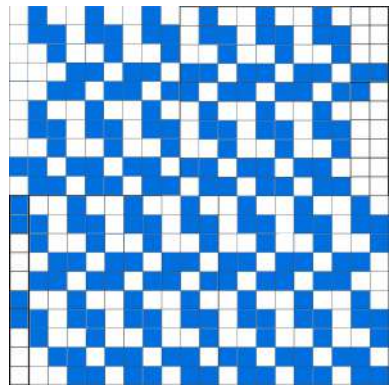
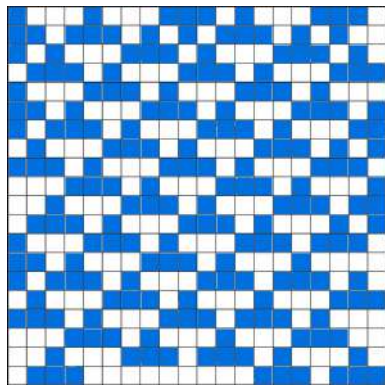
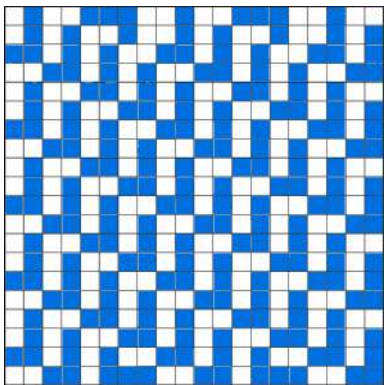
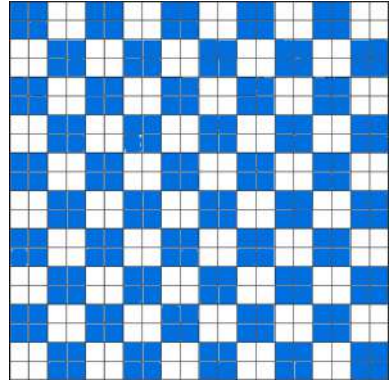
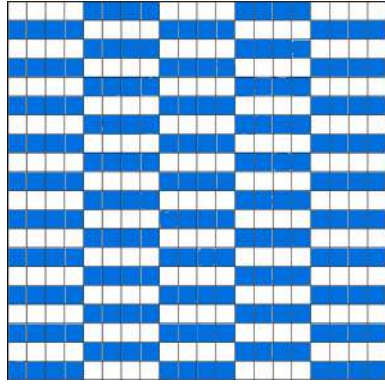
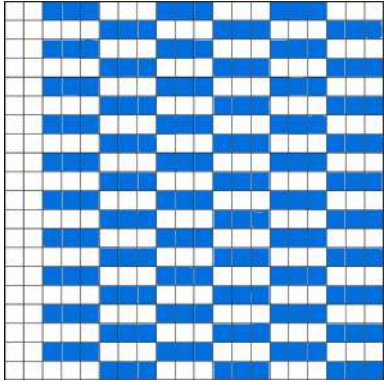
Taulells de 10x10:

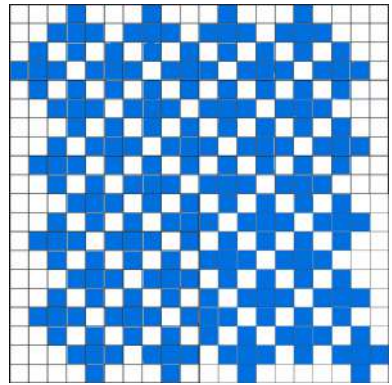
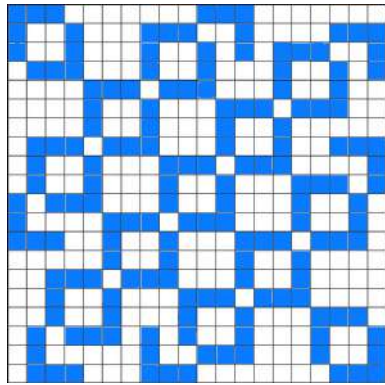
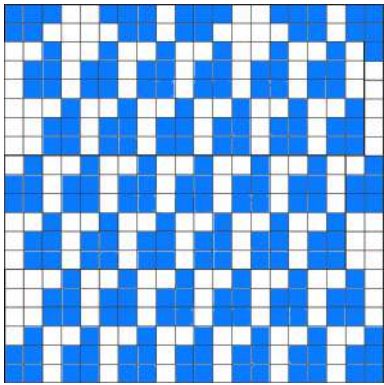
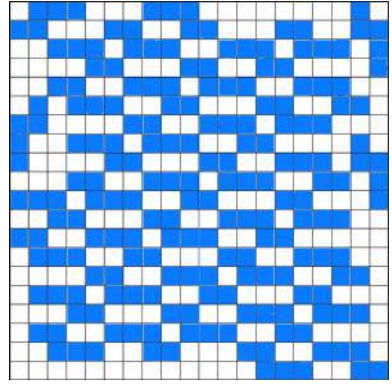
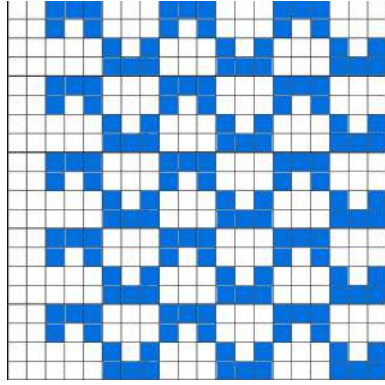
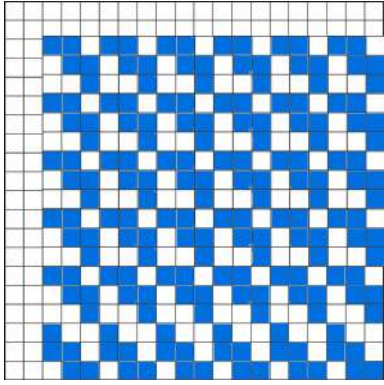




Taulells de 20x20:







Annex V: normas del joc *Sushi Go!*

CONTENIDO

108 CARTAS

14 tempura	10 nigiri de salmón
14 sashimi	5 nigiri de calamar
14 gyoza	5 nigiri de tortilla
12 maki (2)	10 pudín
8 maki (3)	6 wasabi
6 maki (1)	4 palillos

PREPARACIÓN

- Barajad bien las cartas y repartiéndlas según el número de jugadores:

para **2 jugadores**, 10 cartas para cada uno;
 para **3 jugadores**, 9 cartas para cada uno;
 para **4 jugadores**, 8 cartas para cada uno;
 para **5 jugadores**, 7 cartas para cada uno.

Mantened las cartas en la mano, de modo que vuestros oponentes no las vean.

- Colocad las cartas restantes boca abajo, formando una pila en el centro de la mesa.
- Usad papel y lápiz para anotar las puntuaciones y elegid a uno de los jugadores para que se ocupe de ello.

LA PARTIDA

DESCRIPCIÓN DE UNA RONDA

La partida se juega en tres rondas. Para empezar una ronda, elegid todos una carta cualquiera de vuestra mano (una con la que os gustaría quedaros) y, simultáneamente, colocadla boca abajo delante de vosotros. A continuación, revelad las cartas elegidas a la vez.

Tras revelar las cartas, pasad las que quedan en vuestra mano, boca abajo, al jugador que está a vuestra izquierda. Así, el turno siguiente lo empezaréis con una mano nueva y una carta menos.

NOTA: las cartas reveladas permanecen delante de vosotros hasta el final de la ronda, momento en el que se calculan los puntos. ¡Os interesa agrupar cartas del mismo tipo!

USO DEL WASABI

Si eliges una carta de nigiri de calamar, de salmón o de tortilla, y ya tienes una carta de wasabi delante de ti, deberás colocar ese nigiri encima del wasabi para indicar que lo has mojado en la salsa con wasabi... ¡y que ha triplicado su valor!

NOTA: puedes tener varias cartas de wasabi delante de ti, pero solamente puedes colocar una carta de nigiri encima de cada wasabi.

USO DE LOS PALILLOS

Si ya tienes una carta de palillos delante de ti, en algún turno posterior podrás elegir dos cartas de sushi para ponerlas en la mesa.

Para ello, la primera carta la eliges normalmente de tu mano, al principio de un turno. Antes de que los demás jugadores revelen las cartas escogidas, debes decir «Sushi Go!» y entonces coges una segunda carta de tu mano, que también colocarás boca abajo sobre la mesa. A continuación, todos los jugadores revelan sus cartas.

Antes de pasar las cartas de las manos, devuelve la carta de palillos a tu mano. Así, los palillos están de nuevo en circulación y quizás otros jugadores vuelvan a usarlos más adelante.

NOTA: puedes tener varias cartas de palillos delante de ti, pero solamente puedes usar una por turno.

FIN DE UNA RONDA

Cuando se pasa la última carta de cada mano, simplemente ponedla boca arriba junto con las demás cartas que hayáis recopilado. Lo más probable es que esta última carta no sea muy valiosa, pero a veces puede pasar que tu rival no tenga más remedio que cederte algo muy sabroso... ¡ñam!

Ahora calculad los puntos de las cartas que habéis recopilado, de la siguiente manera:

PUNTUACIÓN

MAKI

Cada jugador suma los símbolos de rollos de maki que aparecen en la parte superior de sus cartas de maki. El jugador que tenga más rollos gana **6 puntos**. Si varios jugadores empatan en primera posición, se reparten los 6 puntos a partes iguales (ignora cualquier resto) y no se otorgan puntos para la segunda posición.

El segundo jugador con más rollos gana **3 puntos**. Si varios jugadores empatan en segunda posición, se reparten los 3 puntos a partes iguales (ignora cualquier resto).

Ejemplo: Carla tiene 5 rollos de maki; Pablo, 3; Ana, 3; y Luisa tiene 2. Carla es la que tiene más rollos y gana 6 puntos. Pablo y Ana empatan en segunda posición, por lo que se reparten los 3 puntos (1 para cada uno). Finalmente, Luisa no puntúa.

¡MAKINUDO!
6 PUNTOS,
¡TOMA YA!



¡ARROLLADOR!



ENRÓLLATE,
¡Y ÉCHATE UNAS
RISAS!





TEMPURA

Una pareja (2 cartas) de tempura otorga **5 puntos**. Una única carta de tempura no da puntos. Puedes puntuar varias parejas de tempura en una misma ronda.



SASHIMI

Un trío (3 cartas) de sashimi otorga **10 puntos**. Una única carta o una pareja de sashimi no da puntos. Puedes puntuar varios tríos de sashimi en una misma ronda, ¡pero no te resultará nada fácil!



GYOZA

Cuantas más cartas de gyoza tengas, más puntos ganarás. Así:

Gyoza:	1	2	3	4	5 o más
Puntos:	1	3	6	10	15

INTENTÁBAMOS JUNTAR 3 CARTAS DE SASHIMI, PERO VA ESE... ¡Y SE QUEDA LA ÚLTIMA!

PUES LO TENÉIS CRUDO...



NIGIRI Y WASABI

Un nigiri de calamar otorga **3 puntos**. Si está encima de una carta de wasabi otorga **9 puntos**.



Un nigiri de salmón otorga **2 puntos**. Si está encima de una carta de wasabi otorga **6 puntos**.



Un nigiri de tortilla otorga **1 punto**. Si está encima de una carta de wasabi otorga **3 puntos**.

Una carta de wasabi sin nigiri no da puntos.

PALILLOS

Una carta de palillos no da puntos.

EMPIEZA UNA NUEVA RONDA

- Anotad las puntuaciones de los jugadores en la ronda que acabáis de jugar.
- Dejad las cartas jugadas en una pila de descartes. La única excepción son las cartas de pudín, que conserváis delante de vosotros hasta el final de la partida.
- De la pila de cartas disponibles, repartid una nueva mano de cartas a cada jugador (ved más arriba el número de cartas según el número de jugadores).

FIN DE LA PARTIDA

Tras la puntuación de la tercera ronda, todavía habrá algunas cartas sin jugar en la pila. Ignoradlas... ¡es la hora del postre! Ahora se calculan los puntos de las cartas de pudín.

PÚDINES

El jugador con más cartas de pudín gana **6 puntos**. Si varios jugadores empatan en primera posición, se reparten los 6 puntos a partes iguales (ignorad cualquier resto).



El jugador con menos cartas de pudín (incluyendo aquellos con ninguna carta) **pierde 6 puntos**. Si varios jugadores empatan en última posición, se reparten los puntos perdidos a partes iguales (ignorad cualquier resto).

Ejemplo: Carla tiene 4 cartas de pudín; Pablo, 3; y Luisa y Ana no tienen ninguna. Carla es la que tiene más púdines y gana 6 puntos. Luisa y Ana empatan en última posición, por lo que se reparten los 6 puntos perdidos y, así, cada una pierde 3.

En el caso poco frecuente de que todos los jugadores tengan la misma cantidad de púdines, ningún jugador gana ni pierde puntos por el postre.

NOTA: en una partida de dos jugadores no se pierden puntos por tener menos púdines. Únicamente se otorgan 6 puntos a quien tenga más cartas de pudín.

Y EL GANADOR ES...

... el jugador que haya acumulado más puntos después de tres rondas. Si hay dos o más jugadores empatados, gana quien tenga más cartas de pudín.

VARIANTES

PASAR CARTAS A AMBOS LADOS

Para cambiar la interacción entre los jugadores, probad a alternar el sentido en el que os pasáis las cartas en una y otra ronda. Así, en las rondas 1 y 3, pasad las cartas hacia la izquierda; mientras que en la ronda 2, pasáoslas hacia la derecha.

VARIANTE PARA DOS JUGADORES

En esta variante se crea un tercer jugador ficticio que va alternándose entre los dos jugadores.

- Repartid las cartas como si hubiera tres jugadores (es decir, 3 manos de 9 cartas).
- Colocad las cartas del jugador ficticio entre vosotros, en una pila boca abajo.
- Sorteád qué jugador será el primero en controlar al jugador ficticio.
- Si controlas al jugador ficticio, roba la carta superior de su pila y añádela a tu mano.

- Luego elige una carta para ti y una para el jugador ficticio. Como es habitual, tu oponente juega una carta.
- Revelad las cartas y luego intercambiad vuestras manos, dejando la pila del jugador ficticio en medio.
- Ahora tu oponente controla al jugador ficticio; roba la carta superior de su pila y luego elige una carta para él y una para el jugador ficticio.
- Alternad los turnos para el control de la pila del jugador ficticio, hasta que se hayan jugado todas las cartas.
- Como en cualquier partida, calculad vuestros puntos después de haber jugado tres rondas, siguiendo las reglas normales.

¡FELICIDADES! ¡HAS GANADO!
¿QUÉ TAL SI TE INVITO A MERENDAR
PARA CELEBRARLO?



OH, ERES TAN
DULCE...



UNAS PALABRAS DE GAMEWRIGHT

A la gente de GameWright nos encanta comer sushi, por lo que no fue nada raro que en seguida tuviéramos un flechazo con este «delicioso» juego de cartas sobre el sushi. El autor, Phil Walker-Harding, creó un mecanismo único y muy accesible de selección de cartas, que hace que sea un juego muy fácil de aprender, rápido e interesante para jugadores de todas las edades. Como si disfrutaras de un exquisito manjar en un bar de sushi, este juego conseguirá que nunca te canses de jugarlo, luna y otra vez!

Un juego de Phil Walker-Harding
Ilustraciones: Nan Rangsimá
Muchas gracias a Peter Kinsley por la variante para 2 jugadores



DEVIR

Créditos de la edición española

Devir Iberia, S.L.
C/ Rossello 184 6º 1ª
08008 - Barcelona
www.devir.es

Traducción:
Marià Pitarque y Marc Figueras
Diseño gráfico: Bascu

GAMEWRIGHT®

Games for the Infinitely Imaginative
70 Bridge Street
Newton, MA 02458
Tel: 617-924-6006

email: jester@gamewright.com

www.gamewright.com

©2014 Gamewright, a division of Ceaco Inc.

All rights reserved.

RESUMEN DE REGLAS

3
RONDAS

para 2 jugadores, 10 cartas para cada uno;
para 3 jugadores, 9 cartas para cada uno;
para 4 jugadores, 8 cartas para cada uno;
para 5 jugadores, 7 cartas para cada uno.

Escoged 1 carta. Reveladlas a la vez. Pasad cartas a la izquierda



MAKI
Más: 6
Segundo: 3
Empates: a partes iguales

NIGIRI
De calamar: 3
De salmón: 2
De tortilla: 1



TEMPURA
Pareja: 5
Si no: 0

WASABI
Triplica el valor del siguiente nigiri



SASHIMI
Trio: 10
Si no: 0

PALILLOS
Úsalos en un turno posterior para elegir 2 cartas



GYOZA
x1 2 3 4 5+
1 3 6 10 15

PUDIN
Se puntúan al final de la partida
Más: +6 Menos: -6
Empates: a partes iguales



Annex VI: programa sg5

```

#include<iostream>
#include <math.h>
#include <map>
using namespace std;
int ck=0;
int v=0;
int lcartes[50];
int cc=0;
int tirades[10];
float mans[15][11];
int cd=12;
int cartes=0;
int jugadors=0;
int permutacions=0;
int combinacions=0;
float mitjana=0;
map<int,int> co;
map<int,int> altern;
int escriure(int a, int b)
{
    cout<<"Tiro la carta "<<mans[b*3][a];
    if(mans[b*3][a]==0){cout<<"(Palillos)"<<endl;}
    if(mans[b*3][a]==1){cout<<"(Maki (1))"<<endl;}
    if(mans[b*3][a]==2){cout<<"(Makis (2))"<<endl;}
    if(mans[b*3][a]==3){cout<<"(Makis (3))"<<endl;}
    if(mans[b*3][a]==4){cout<<"(Wasabi)"<<endl;}
    if(mans[b*3][a]==5){cout<<"(Nigiri de
Tortilla)"<<endl;}
    if(mans[b*3][a]==6){cout<<"(Nigiri de
Salmon)"<<endl;}
    if(mans[b*3][a]==7){cout<<"(Nigiri de
Calamar)"<<endl;}
    if(mans[b*3][a]==8){cout<<"(Gyoza)"<<endl;}
    if(mans[b*3][a]==9){cout<<"(Sashimi)"<<endl;}
    if(mans[b*3][a]==10){cout<<"(Tempura)"<<endl;}
    if(mans[b*3][a]==11){cout<<"(Pudin)"<<endl;}
    return 0;
}
int ma(int a)
{
    while(a>=jugadors)
    {
        a=a-jugadors;
    }
    return a;
}
int contar()
{
    int punts=0;
    int numc[12];
    for(int i=0;i<cd;i++)
    {
        numc[i]=0;
    }
    for(int i=0;i<cartes;i++)
    {
        for(int j=0;j<12;j++)
        {
            if(co[i]==j)
            {
                numc[j]++;
            }
        }
    }
    int wasabis[cd];
    for(int i=0;i<cd;i++)
    {
        wasabis[i]=0;
    }
    int w3=0;
    int w2=0;
    int w1=0;
    int h=0;
    for(int i=7;i>4;i--)
    {
        for(int j=cartes-1;j>=0;j--)
        {
            if(co[j]==i)
            {
                h=0;
                for(int g=j;g>=0;g--)
                {
                    if(co[g]==4 and wasabis[g]!=1
                    and h==0)
                    {
                        wasabis[j]=1;
                        wasabis[g]=1;
                        h++;
                        if(i==7)
                        {
                            w3++;
                        }
                        else if(i==6)
                        {
                            w2++;
                        }
                        else if(i==5)
                        {
                            w1++;
                        }
                    }
                }
            }
        }
        punts = punts + w3*9 + w2*6 + w1*3;
        for(int i=0;i<cartes;i++)
        {
            if(wasabis[i]==0 and co[i]==5)
            {
                punts = punts + 1;
            }
            else if(wasabis[i]==0 and co[i]==6)
            {
                punts = punts + 2;
            }
            else if(wasabis[i]==0 and co[i]==7)
            {
                punts = punts + 3;
            }
        }
        punts = punts + ((numc[8]*(numc[8]+1))/2);
        punts = punts + (trunc(numc[9]/3)*10);
        punts = punts + (trunc(numc[10]/2)*5);
        return punts;
    }
}
int comb(int a, int c)
{
    if(lcartes[a]==0)
    {
        if(cc==0)
        {
            cc=1;
            for(co[a]=0;co[a]<cd;co[a]++)
            {
                if(a+1<cartes)
                {
                    comb(a+1, c);
                }
                else
                {
                    mitjana= mitjana + contar();
                    combinacions++;
                }
            }
        }
        return 0;
    }
    else
    {
        ck=a-1;
        while(ck!=-1)
        {
            ck=ck-1;
        }
        for(co[a]=co[ck];co[a]<cd;co[a]++)
        {
            if(a+1<cartes)
            {
                comb(a+1, c);
            }
            else
            {
                mitjana= mitjana + contar();
                combinacions++;
            }
        }
    }
}

```



```

    perm(1+g, g2);
    mitjana= mitjana/permutacions;
}
mans[g2*3+2][i]=mitjana;
cout<<mitjana<<" de "<<mans[g2*3][i]<<endl;
for(int j=0;j<cartes;j++)
{
    if(mans[g2*3][j]==mans[g2*3][i] and mans[g2*3+1][j]!=2)
    {
        mans[g2*3+1][j]=1;
        mans[g2*3+2][j]=mitjana;
    }
}
}
}
millorcarta=0;
for(int i=0;i<cartes;i++)
{
    if(mans[g2*3+2][i]>mans[g2*3+2][millorcarta] and mans[g2*3+1][i]!=2)
    {
        millorcarta=i;
    }
}
mans[g2*3+1][millorcarta]=2;
tirades[g]=mans[g2*3][millorcarta];
escriure(millorcarta,g2);
int x=0;
int y=0;
int check=0;
if(g<cartes-1)
{
    cout<<"(Nombre de jugadors en sentit antihorari)"<<endl;
    for(int i=2;i<jugadors+2-1;i++)
    {
        check=0;
        cout<<"Carta jugada pel jugador "<<i<<": ";
        cin>>x;
        y=i-1+g;
        while(y>=jugadors)
        {
            y=y-jugadors;
        }
        y=y*3;
        for(int j=0;j<cartes;j++)
        {
            if(mans[y][j]==x and check==0 and mans[y+1][j]!=2)
            {
                mans[y+1][j]=2;
                check=1;
            }
        }
        if(check==0)
        {
            mans[y][g2]=x;
            mans[y+1][g2]=2;
        }
    }
}
}
}
}

```