

n



Implementació d'una intel·ligència artificial amb diferents personalitats per generar contrincants a un joc de tauler (Dragon Slayer).

Ramon Castells Amat
Grau d'Enginyeria Informàtica
Àrea d'Intel·ligència Artificial

Consultor: David Isern Alarcón
Professor responsable de l'assignatura: Carles Ventura Royo

Data de lliurament: 23/04/2018



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Descriptiu del treball</i>
Nom de l'autor:	<i>Ramon Castells Amat</i>
Nom del consultor/a:	<i>David Isern Alarcón</i>
Nom del PRA:	<i>Nom i dos cognoms</i>
Data de lliurament (mm/aaaa):	<i>04/2018</i>
Titulació o programa:	<i>Grau d'Enginyeria Informàtica</i>
Àrea del Treball Final:	<i>Intel·ligència artificial</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Joc amb adversari, Personalitat, Jocs de taula, Dragon Slayer</i>
<p>Resum del Treball (màxim 250 paraules): <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball</i></p>	
<p>En la majoria de jocs en els que es simula un jugador humà la intel·ligència artificial que controla el rival, intenta maximitzar les seves opcions de guanyar i, quan es permet escollir entre diferents contrincants, la diferència entre ells és el nivell de dificultat que ofereixen al jugador.</p> <p>L'objectiu del projecte és crear un seguit de personalitats (basades en una intel·ligència artificial eficient) però que, a la vegada, també tinguin característiques humanes (més o menys seguretat, capacitat de càlcul, d'assumir un determinat risc, millors en planificació a curt o llarg termini, etc) per dotar de realisme, a més d'intel·ligència, als rivals per poder jugar en solitari contra els diferents rivals del joc.</p> <p>Aquesta intel·ligència artificial amb diferents personalitats s'implementarà en un joc de taula anomenat Dragon Slayer degut a les seves característiques: les mecàniques es comprenen amb facilitat, conté un cert nivell d'atzar i la combinació de mecàniques li dóna certa complexitat a l'hora de jugar-hi.</p> <p>En la implementació dels diferents contrincants es realitzarà un estudi de les</p>	

característiques de la personalitat més destacades per generar cadascun d'ells i que es comportin segons la seva pròpia manera de ser davant davant un problema d'aquestes característiques.

Abstract (in English, 250 words or less):

In most games in which a human player simulates the artificial intelligence that the opponent controls, he tries to maximize his winning options and, when it is allowed to choose between different opponents, the difference between them is the level of difficulty that they offer the player.

The objective of the project is to create a series of personalities (based on efficient artificial intelligence) but which, at the same time, also have human characteristics (more or less security, ability to calculate, to assume a certain risk, better in short or long term planning, etc.) to provide realism, as well as intelligence, to the rivals to be able to play alone against the different rivals of the game.

This artificial integration with different personalities will be implemented in a board game called Dragon Slayer due to its characteristics: the mechanics are easily understood, contains a certain level of chance and the combination of mechanics gives some complexity to the the time to play it.

In the implementation of the different opponents, a study of the most outstanding personality characteristics will be carried out to generate each of them and that they behave according to their own way of being faced with a problem of these characteristics.

Índex

1 - Introducció.....	1
1.1 - Context i justificació del Treball.....	1
1.2 - Objectius del Treball.....	1
1.3 - Enfocament i mètode seguit.....	3
1.4 - Planificació del Treball.....	4
1.5 - Breu sumari de productes obtinguts.....	9
1.6 - Breu descripció dels altres capítols de la memòria.....	10
2 - Software.....	11
3 – Instruccions del joc.....	13
4 - Intel·ligència artificial.....	18
4.1 - Arbres de decisió.....	18
4.2 - Jocs sense adversari.....	18
4.3 - Jocs amb adversari.....	19
4.4 - Jocs amb adversari i elements d'atzar.....	23
4.5 - Dragon Slayer.....	25
5 - Representació del coneixement.....	34
6 - Personalitat.....	37
6.1 – Característiques dels jugadors.....	38
6.2 – Implementació de les característiques.....	38
6.3 – Personalitats implementades.....	41
7 - Disseny de la GUI.....	45
8 - Conclusions.....	48
9 - Glossari.....	51
10 - Bibliografia.....	53

Llista de figures

Figura 1: Principals components del joc (daus i tokens).....	2
Figura 2: Diagrama de Gantt amb la planificació del projecte.....	9
Figura 3: Cares de cada dau de drac amb el cap de cadascun d'ells.....	13
Figura 4: Cares de cada dau de drac amb les ales de cadascun d'ells.....	14
Figura 5: Cares de cada dau de drac amb la cua de cadascun d'ells.....	14
Figura 6: Cares de cada dau de drac amb la representació de les muntanyes.....	14
Figura 7: Cares de cada dau de drac amb la representació de les flames.....	15
Figura 8: Cara del dau de jugador que representa la destralt.....	15
Figura 9: Cara del dau de jugador que representa l'escut.....	15
Figura 10: Cara del dau de jugador que representa les flames.....	15
Figura 11: Token que representa l'ús del desafiament a l'altre jugador.....	17
Figura 12: Exemple d'arbre minmax.....	22
Figura 13: Exemple d'arbre minmax amb poda.....	23
Figura 14: Exemple d'arbre minmax amb elements d'atzar.....	24
Figura 15: Exemple d'arbre minmax amb l'atzar valorat als nodes.....	25
Figura 16: Diagrama de flux d'un torn de jugador a Dragon Slayer.....	30
Figura 17: Estructura per decisió d'elecció de drac o plantar-se.....	31
Figura 18: Estructura millorada per decisió d'elecció de drac o plantar-se.....	31
Figura 19: Part superior de l'arbre de decisió.....	32
Figura 20: Part inferior de l'arbre.....	32
Figura 21: Principals tipus de jugadors segons Richard Bartle.....	37
Figura 22: Gràfic de la personalitat de l'Abel.....	41
Figura 23: Gràfic de la personalitat de la Beatriu.....	42
Figura 24: Gràfic de la personalitat de la Carla.....	43
Figura 25: Gràfic de la personalitat del Daniel.....	44
Figura 26: Disseny de la GUI del Dragon Slayer.....	47
Figura 27: Ús de la GUI en el transcurs d'una partida.....	47

1 - Introducció

1.1 - Context i justificació del Treball

En la majoria de jocs en els que es simula un jugador humà (per exemple, els escacs), la intel·ligència artificial que controla els rivals, intenta maximitzar les seves opcions de guanyar i, quan es permet escollir entre diferents contrincants, la diferència entre ells és el nivell de dificultat que ofereixen al jugador.

Es crearan un seguit de contrincants amb diferent personalitat (més o menys conservadors, calculadors, entusiastes, etc) contra els que poder enfrontar al jugador.

1.2 - Objectius del Treball

L'objectiu del projecte és crear una intel·ligència artificial eficient però que, a la vegada, també tingui característiques humanes.

En el joc de taula que s'implementarà (un joc dels coneguts com a "filler" anomenat Dragon Slayer), a banda de jugar amb intel·ligència, també hi ha un cert nivell d'atzar (daus) que cal tenir en compte però el factor clau del joc és el risc que s'està disposat a assumir per intentar superar la puntuació del rival en cada torn fins al final, per això s'estudiaran els diferents tipus de personalitat que es poden comportar de manera diferent davant un problema d'aquestes característiques i s'implementaran per dotar a cada contrincant d'una personalitat diferent.

A més el joc implementa en les seves mecàniques moments d'interacció entre els jugadors, mitjançant el plantejament d'un desafiament sobre si seguir o plantar-se, que ajuda a afegir un element extra que no té a veure amb el risc o la sort.



Figura 1: Principals components del joc (daus i tokens)

1.2.1 - Objectius generals

L'objectiu del projecte és implementar el joc en un ordinador per poder jugar en solitari contra les diferents intel·ligències del joc o en xarxa contra altres rivals humans.

En la implementació dels diferents contrincants es realitzarà un estudi de les característiques de la personalitat més destacades i se'ls assignaran diferents valors per generar a cadascun d'ells i es comportin segons la seva pròpia manera de ser i, d'aquesta manera, el jugador tindrà la sensació que juga contra un rival humà (amb els diferents graus de les diverses debilitats que tothom té).

1.2.2 - Objectius específics

Segons Richard Bartle, autor de Designin Virtual Worlds existeixen 4 tipus de jugadors (aquesta classificació és la que s'utilitza en projectes de gamificació):

1. Killer: la motivació d'aquest usuari és posicionar-se el primer i és el tipus de jugador que juga per guanyar.
2. Achiever: el seu objectiu és superar els objectius marcats pels joc i es centra en motivacions intrínseques relacionades amb la satisfacció personal i el bé del grup.
3. Socializer: gaudeix compartint el joc amb la resta.
4. Explorer: el motiva l'activitat en sí mateixa i busquen l'autosuperació i el descobriment.

Així doncs s'implementaran aquestes diferents tipus de jugador i, a banda, es contemplaran les diferents formes més característiques en que cadascun d'ells pot afrontar un joc que conté atzar segons la pròpia personalitat humana: aversió al risc, intel·ligència i impulsivitat.

- Amb l'aversió al risc introduïrem un factor extra a la valoració de fins a quin punt val la pena arriscar-se segons el possible benefici o la possible pèrdua segons el jugador sigui més o menys nerviós, competitiu, intel·ligent, etc.
- Amb la intel·ligència gestionarem un tipus de pensament més o menys calculat i de més o menys avaluació de la situació.
- Amb la impulsivitat modificarem el comportament segons diversos criteris: puntuació anterior del jugador humà, ratxa actual, etc, tenint en compte factors que poden modificar aquesta impulsivitat com pot ser el factor de competitivitat del jugador, el nerviosisme davant una situació negativa o d'altres.

D'aquesta manera s'implementaran en total 12 contrincants diferents combinant els 4 tipus existents de jugador amb els 3 factors de personalitat que afecten al joc.

1.3 - Enfocament i mètode seguit

El projecte s'aborda dividint-lo en diverses fases que cal anar completant de forma cronològica, que serien:

1. planificació: tal i com es detalla a l'apartat següent 1.4.
2. documentació i estudi: tant del funcionament del joc i les seves mecàniques com dels diferents mètodes per implementar la intel·ligència

artificial que determinarà les respostes i decisions del jugador controlat per l'ordinador.

3. generació d'entorn gràfic i sonor: tot i que Dragon Slayer sigui un joc de daus i tauler requereix un seguit de treball, sobretot gràfic, per fer-lo atractiu a ulls del jugador i que formarà la part frontend del programa.
4. implementació de regles: en que es programen totes les mecàniques i que es pugui jugar seguint les regles del joc.
5. implementació de la intel·ligència artificial: en base a les regles introduïdes a l'apartat anterior es programarà el sistema que donarà resposta de forma intel·ligent a les accions del jugador humà plantejant-li el repte de superar-les.
6. implementació de les diferents personalitats: aplicació de diferents criteris a l'hora d'avaluar la següent jugada a realitzar per part de l'ordinador, en funció de la intel·ligència implementada prèviament i segons uns paràmetres d'avaluació personal.

Al llarg de cada fase es fan executant les proves pertinents per anar validant cada subtasca en la que es divideix cada fase.

1.4 - Planificació del Treball

1.4.1 - Fites i temporalització

- Fase 1 - Planificació

Estudi i estructura temporal del desenvolupament del projecte per setmanes amb el nombre d'hores indicat per setmana i tasca.

Inici i fi: segona setmana de març

Fita: Document amb el diagrama de Gantt (adjunt al final d'aquest apartat) amb la planificació realitzada.

- Fase 2 - Estudi previ

- Fase 2.1 - Joc

Estudi de les regles del joc.

Inici i fi: tercera setmana de març.

Fita: entendre el funcionament del joc, possibles accions i estratègies a seguir.

- Fase 2.2 - Implementació

Estudi sobre el tipus de llenguatge a utilitzar, estructures de dades, representació del coneixement i sistema d'intel·ligència artificial.

Inici i fi: tercera setmana de març.

Fita: documentar totes les decisions preses i raonar els motius pels quals s'han pres aquestes i no unes altres.

- Fase 2.3 - Personalitats

Estudi de les diferents personalitats aplicades al món lúdic i formes d'encarar les diverses situacions que ens planteja el joc segons cada manera de ser.

Inici i fi: quarta setmana de març.

Fita: exposar a la documentació els tipus de personalitats escollides per adaptar-les al joc, motius i diferències entre elles per generar els possibles rivals.

- Fase 3 - Entorn gràfic

- Fase 3.1 - Disseny elements del joc

Disseny i producció de tots els elements gràfics del joc: tauler, daus, fitxes, marcadors de puntuació, etc.

Inici i fi: de la primera a la segona setmana d'abril.

Fita: realitzar tot l'apartat gràfic del joc per, a partir d'aquesta fase i la següent, poder centrar-nos en la implementació del joc en si i la IA.

- Fase 3.2 - Disseny UI

Disseny i producció de l'entorn gràfic (User Interface) mitjançant el qual l'usuari podrà jugar contra els diferents rivals.

- Inici i fi: tercera setmana d'abril.

- Fita: realitzar tot l'entorn de joc que utilitzarà el jugador humà per, a partir d'aquesta fase i l'anterior, poder centrar-nos en la implementació del joc en si i la IA.

- Fase 4 - Entorn sonor

- Fase 4.1 - Gravació de sons

Gravació dels diversos sons del joc: tirades de daus, efectes sonors, etc.

Inici i fi: segona setmana d'abril.

Fita: deixar enregistrats tots els sons necessaris per donar més ambientació al joc.
- Fase 4.2 - Edició de sons

Edició per adaptar els sons al desenvolupament de la partida.

Fita: obtenció dels sons editats llestos per ser incorporats al programa.

Inici i fi: tercera setmana d'abril.
- Fase 5 - Implementació
 - Fase 5.1 - Implementació entorn

Primerament s'implementarà tot allò que no té a veure amb el joc directament en si: com iniciar partida, tirar els daus, moure fitxes, etc.

Inici i fi: quarta setmana d'abril.

Fita: el programa ha de realitzar totes les tasques que no tenen a veure amb les regles del joc en sí.
 - Fase 5.2 - Implementació regles

La segona fase de la implementació consistirà en introduir al programa les regles del joc perquè tant el jugador humà com la IA les segueixin.

Inici i fi: de la primera a la segona setmana de maig.

Fita: el programa ha de permetre que dos o més jugadors humans puguin jugar entre ells.
 - Fase 5.3 - Implementació IA

En una tercera fase s'implementarà la IA en si, de manera que jugui contra el jugador humà de la forma més òptima possible.

Inici i fi: de la tercera setmana de maig a la primera setmana de juny.

Fita: un o més jugadors humans han de poder jugar contra la IA del joc.
 - Fase 5.4 - Implementació personalitats

Finalment s'estudiarà i es programaran adaptacions de la IA per generar els diferents rivals als que es podrà enfrontar el jugador humà. Com s'ha explicat al punt 2.2 (objectius específics) cal aplicar 3 factors de personalitat que modificaran el comportament del rival escollit entre els 4 tipus de jugador que proposa Bartle i, així doncs, es dedicarà una setmana a cada tipus de jugador.

Inici i fi: de la tercera setmana de maig a la primera setmana de juny.

Fita: els jugadors humans han de poder escollir un rival que tingui una personalitat diferent i que afronti les decisions durant la partida en conseqüència.

- Fase 6 - Proves

- Fase 6.1 - Disseny del joc de proves

Es detallaran tot el tipus de proves necessaris per, un cop superades, confirmar que el programa funciona correctament davant qualsevol situació possible.

Inici i fi: quarta setmana d'abril.

Fita: documentar totes les proves necessàries a realitzar.

- Fase 6.2 - Proves alpha

Seràn totes les proves que es duran a terme mentre s'implementa cada nova funcionalitat a càrrec del programador, que serà el mateix que farà aquestes proves.

Inici i fi: tot el mes de maig (de la primera a la cinquena setmana).

Fita: comprovació de que el programa supera totes les proves detallades a la fase 6.1.

- Fase 6.3 - Proves beta

Un cop superades les proves alpha es passarà el programa a terceres persones (jugadors habituals de Dragon Slayer) perquè el provin i realitzin cadascuna d'elles un joc de proves beta.

Inici i fi: primera setmana de juny.

Fita: comprovació de que el programa supera totes les proves detallades a la fase 6.1.

- Fase 7 - Redacció de la memòria

Redacció del document amb tota la informació del projecte, decisions, tipus d'implementació, de jugadors possibles, etc.

Inici i fi: durant tot el desenvolupament del projecte.

Fita: obtenció del document final a entregar amb la memòria del Treball Final de Grau.

- Fase 8 - Elaboració de la presentació

- Fase 8.1 - Preparació

Preparació tècnica per realitzar la presentació en vídeo del Treball Final de Grau i generació de la presentació amb diapositives.

Inici i fi: primera setmana de juny.

Fita: obtenció de l'entorn i maquinari per poder dur a terme la gravació de la presentació durant la següent fase.

- Fase 8.2 - Gravació

Enregistrament en si de la presentació del treball.

Inici i fi: primera setmana de juny.

Fita: Obtenció d'un vídeo amb la presentació del projecte

- Fase 8.3 - Postproducció

Edició de vídeo i veu en off.

Inici i fi: segona setmana de juny.

Fita: Obtenció del vídeo preparat per ser entregat al campus de la UOC.

- Fase 9 - Defensa pública

Defensa final del Treball Final de Grau.

Inici i fi: tercera setmana de juny.

Fita: superació de la defensa pública.

1.4.2 – Diagrama de Gantt

Es crea el diagrama de Gantt amb la planificació descrita on es calcula aproximadament un total de 270 hores de feina per dur a terme el projecte tenint en compte el següent:

- es deixen unes 30 hores de marge (un 10% per arribar a les 300) per fer front als inevitables problemes, endarreriments i incidències que puguin aparèixer, sobretot en les fases més delicades del treball, com serà la implementació de la intel·ligència artificial i les diferents personalitats.
- es calcula la planificació de manera que cap setmana hi hagi més de 22 hores de dedicació, bàsicament en les setmanes amb més volum de feina un cop les fases menys complicades s'hagin assolit (creació d'entorn gràfic i sonor, anàlisis de mecàniques de joc, etc).

La planificació és realista perquè cada setmana es poden dedicar entre 5 i 6 hores en total comptabilitzant els dies laborables i, a més, el cap de setmana es disposen dels dos dies de festa dels que un es dedicarà íntegrament al projecte (en forma de dos mitjos dies).

A la figura 2 es mostra el diagrama de Gantt corregit amb les 30 hores utilitzades de marge a les seves tasques corresponents.

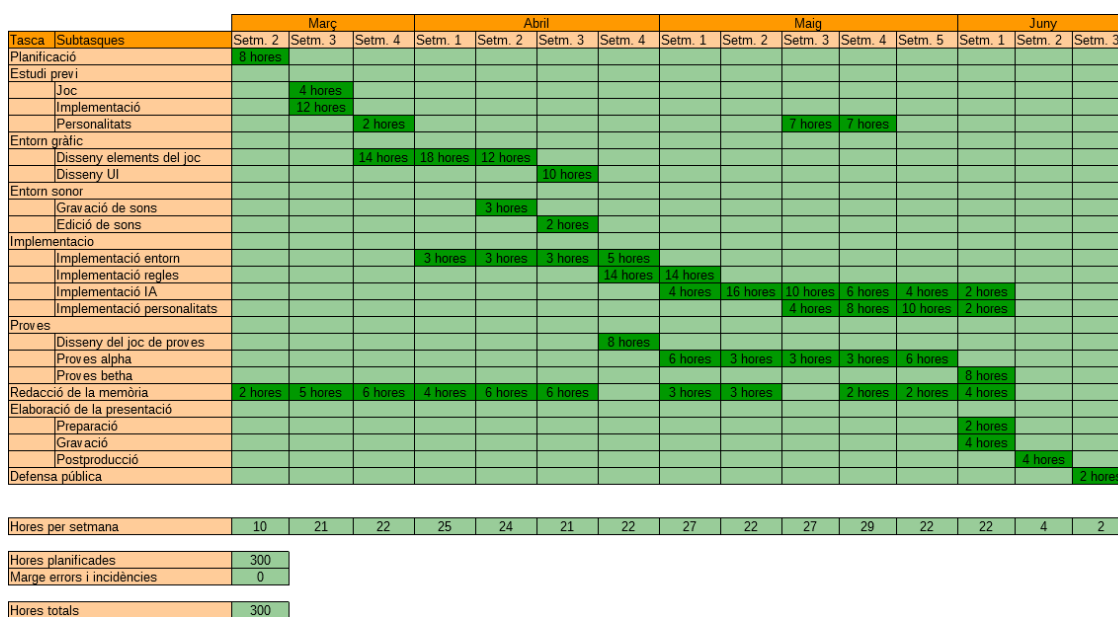


Figura 2: Diagrama de Gantt amb la planificació del projecte

1.5 - Breu sumari de productes obtinguts

L'objectiu del projecte és implementar un joc de tauler, concretament el Dragon Slayer, un joc de daus amb mecànica de «push your luck», és a dir, de temptar

la sort on, el rival al que s'haurà d'enfrontar el jugador humà, no només es plantegi la partida d'una forma intel·ligent sinó que mostri, a més, certs trets de la personalitat humana com poden ser la por a perdre, la precipitació en alguns moments, errors de càlcul, etc i crear un seguit de diferents adversaris amb característiques diferenciades.

1.6 - Breu descripció dels altres capítols de la memòria

A banda del present capítol d'introducció, la present memòria consta dels següents capítols:

Capítol 2 - Software: relació de programari de software lliure utilitzat de forma directa o indirecta pel desenvolupament del projecte.

Capítol 3 – Instruccions del joc: exposició completa de les mecàniques del Dragon Slayer.

Capítol 4 - Intel·ligència artificial: enumeració i explicació dels sistemes utilitzats en la resolució de problemes, jocs amb adversari, elements d'atzar i estudi sobre com adaptar aquestes tècniques al projecte.

Capítol 5 - Representació del coneixement: exposició de com es s'emmagatzemarà la informació dels estats del joc i com es valoraran els mateixos.

Capítol 6 - Personalitat: resum dels 4 tipus de jugador segons Richard Bartle, com s'han implementat les característiques que defineixen als jugadors i resum de les quatre personalitats implementades.

Capítol 7 - Disseny de la GUI: exposició dels criteris més adequats per dissenyar una interfície d'usuari i com s'han aplicat al projecte actual

Capítol 8 - Conclusions: Resum crític dels objectius aconseguits, procés d'aprenentatge i possibles futures millores.

Capítol 9 - Glossari: resum de termes utilitzats a la present documentació relacionats amb el projecte.

Capítol 10 - Bibliografia: relació de llibres, articles, documents i pàgines web consultades durant la realització del projecte.

2 - Software

Ens caldrà determinar per a quin sistema operatiu dissenyarem el joc i, posteriorment, trobar l'entorn i llenguatge de programació, eines de disseny gràfic i de so més adients per crear i dissenyar l'aspecte visual del joc i l'apartat sonor utilitzant dos requisits fonamentals:

- cal que sigui programari lliure o amb llicència gratuïta per estudiants.
- ha de cobrir per si sol totes les necessitats de l'aspecte que vol cobrir, per evitar utilitzar dos o tres programes diferents per una mateixa tasca.

Així doncs a cada apartat es pren la següent decisió:

1. Sistema operatiu: es plantegen dues possibilitats

- Windows: es descarta al ser un sistema operatiu amb llicència de pagament
- Ubuntu: sistema operatiu basat en programari lliure

Per motius de llicència s'escolleix Ubuntu (versió 16,04 LTS)



2. Entorn i llenguatge de programació: ens cal un llenguatge potent per desenvolupar la intel·ligència artificial i, a la vegada, que ens permeti integrar-hi fàcilment l'apartat gràfic del joc.

Un dels entorns més còmodes per realitzar tasques que integrin fàcilment gràfics i programació en Windows és la suite Visual Basic, de la que en disposem un programari similar i basat en programari lliure per Ubuntu anomenat Gambas, que actualment es troba en la seva versió 3.8.4.

S'estudia aquesta opció, valora com funciona tant a nivell de programació com de disseny gràfic d'interfícies amb l'usuari i es decideix utilitzar aquest programari.



A l'apartat de valoració de sistema operatiu es té en compte també la possibilitat d'utilització d'un entorn web per obtenir un programa independent del sistema operatiu utilitzat, però es descarta en favor d'obtenir un aspecte visual més modern i atractiu utilitzant les eines que proporciona Gamas 3.

3. Disseny gràfic: s'utilitza la versió 2,8,22 del conegut entorn de disseny gràfic GIMP per Ubuntu, basat en programari lliure (amb llicència GPL i LGPL) i que cobreix totes les necessitats d'edició d'imatges necessàries per un projecte d'aquestes característiques.



4. Edició de so: degut a que només ens cal enregistrar i editar un nombre baix de talls d'àudio s'utilitza un programa relativament senzill anomenat Audacity, amb llicència GNU GPL v2+



3 – Instruccions del joc

Dragon Slayer és un joc de daus que va veure la llum gràcies al seu mecenatge mitjançant KS (Kickstarter), en el que cada jugador encarna un cavaller que es dedica a buscar i derrotar poderosos dracs i competeix amb la resta per ser considerat el millor.

Per fer-ho cal ser el primer que passi de 40 punts i també ser el que en té més quan hagi finalitzat una ronda de joc, és a dir, tots els jugadors hagin jugat el mateix nombre de torns. En cas d'empat, una ronda extra determinarà el guanyador.

Els punts s'aconsegueixen derrotant dracs:

- Drac blau (el més senzill) : 2 punts
- Drac verd (de dificultat mitja) : 4 punts
- Drac vermell (el més difícil) : 6 punts

En el joc tenim 3 daus que representen al drac blau, 3 al drac verd, 3 al drac vermell i 3 al jugador.

Com caçar un drac: al seu torn cada jugador agafa els seus daus i els 3 que corresponguin al drac que vol derrotar.

Els daus dels dracs tenen 5 símbols:

Cap – El jugador ha trobat on caça el drac.



Figura 3: Cares de cada dau de drac amb el cap de cadascun d'ells

Ales – El jugador ha vist al drac volant.



Figura 4: Cares de cada dau de drac amb les ales de cadascun d'ells

Cua – El jugador ha trobat el refugi del drac.



Figura 5: Cares de cada dau de drac amb la cua de cadascun d'ells

Muntanyes – El jugador no ha trobat res destacable.

(el drac vermell no té muntanyes entre els seus daus)



Figura 6: Cares de cada dau de drac amb la representació de les muntanyes

Alè de foc – El drac ha atacat al jugador.

(el drac blau no ataca amb els seus daus)



Figura 7: Cares de cada dau de drac amb la representació de les flames

Els daus del jugador tenen 3 símbols:

Destral – El jugador està a punt de derrotar al drac.



Figura 8: Cara del dau de jugador que representa la destral

Escut – El jugador es defensa contra els atacs del drac.



Figura 9: Cara del dau de jugador que representa l'escut

Alè de foc – El drac ha atacat al jugador.



Figura 10: Cara del dau de jugador que representa les flames

Un cop llençats els daus del drac i els del jugador:

a) apartem totes les cares que mostrin el cap, les ales o la cua del drac.

b) comparem el nombre de "Alè de foc" amb el nombre de "Escuts".

- Si hi ha més escuts que ales de foc (o el mateix nombre) el jugador s'ha defensat amb èxit dels atacs.
- Si hi ha més "Alè de foc" que "Escuts" el jugador perd un dau de guerrer per cada "Alè de foc" que no s'hagi pogut aturar i, si perd els tres, passa el torn al següent jugador i no puntua.

Continuar la caça: Tornem a tirar els daus que ens quedin (després d'apartar els de drac de l'apartat a i els que s'hagin pogut perdre de jugador a l'apartat b) fins que tinguem apartats els tres de drac i, com a mínim, obtinguem a la tirada actual una destrala. Quan això passa hem aconseguit matar al drac i, si volem, podem passar a intentar-ne caçar un altre o aturar-nos i puntuar. Un cop es comença a intentar caçar un drac cal continuar fins que se'l derrota o el guerrer perd el torn.

Caçar un altre drac: si el jugador decideix intentar caçar un altre drac ha de seleccionar un dels que quedin, agafant els 3 daus corresponents del drac i els que li quedin de guerrer repetint el procés de caça de nou fins que el drac sigui derrotat o el guerrer perdi el torn. Cal derrotar als tres dracs per poder tornar a enfrontar-se per segon cop a un drac (el mateix pel tercer cop, etc).

Aturar-se i puntuar: després de derrotar a un drac el jugador es pot aturar i puntuar per tots els dracs que hagi vençut aquell torn. Just abans de puntuar la resta de jugadors tenen l'oportunitat de desafiar al jugador actual a continuar.

Desafiaments: Qualsevol jugador pot gastar la seva fitxa de desafiament (veure el token que la representa a la figura 11) per forçar al jugador que va a puntuar a seguir la cacera i aquest pot escollir entre:

- No acceptar el desafiament: el seu torn finalitza i només aconsegueix la meitat dels punts que hauria aconseguit. A més, el jugador que ha proposat el desafiament guanya 5 punts.
- Continuar amb la cacera: el jugador ha de derrotar, com a mínim, un altre drac per poder-se plantar de nou i puntuar però aconseguirà el doble de punts per cada drac que derroti a partir d'aquell moment. Si continua i és vençut el jugador actual no aconseguirà cap punt i el jugador que l'ha desafiat n'aconseguirà 5.

Només es pot desafiar a un jugador un cop per torn.



Figura 11: Token que representa l'ús del desafiament a l'altre jugador

Guanyador: es considera vencedor de la partida aquell qui, al finalitzar una ronda (és a dir, tots els jugadors han tingut el mateix nombre de torns), passi de 40 punts i en tingui més.

4 - Intel·ligència artificial

Anem a veure com s'implementa un sistema d'intel·ligència artificial en la resolució de problemes i, posteriorment, en jocs amb adversari i amb elements d'atzar com és el cas que ens ocupa per, finalment, realitzar un estudi en profunditat sobre com adaptar aquestes eines existents i el càlcul de probabilitat necessari per implementar-ho al Dragon Slayer.

4.1 - Arbres de decisió

En el camp de la intel·ligència artificial s'utilitzen arbres i grafs per arribar a una solució, ja sigui en problemes que la màquina hagi de resoldre (com podria ser resoldre un sudoku) o jocs amb un altre adversari (com els escacs o les dames).

Els elements que componen un arbre de decisió són:

- Els nodes: que conformen cadascun dels estats possibles en que es pot trobar el joc en un moment donat.
- Valor dels nodes: valor numèric que indica en quina mesura és més propici per arribar a la solució un node (estat) que un altre.
- Fletxes: s'utilitzen per unir els nodes als que podem accedir des d'un estat previ mitjançant sempre una jugada vàlida.

4.2 - Jocs sense adversari

Per trobar una solució el més òptima possible en termes de temps i moviments en jocs sense adversari (resolució d'un sudoku, laberints, etc) es poden utilitzar tant algorismes de força bruta (si disposem dels recursos necessaris en termes de memòria i temps de computació) o algun tipus d'algorisme heurístic, sent el més conegut l'algorisme A* (Hart, 1968).

Algorisme A*

Aquest algorisme implementa una cerca best-first (primer el millor), és a dir, a partir d'un estat inicial intenta trobar quins moviments vàlids són necessaris per arribar a l'estat final (solució) mitjançant aquells moviments que siguin més favorables per minimitzar el nombre de moviments a realitzar o el temps per fer-los.

Per conèixer quin moviment és millor s'utilitza una funció d'avaluació estàtica f formada per:

$$f=g+h$$

on

- f serà el valor numèric associat a un estat concret del joc
- g és la funció del cost d'arribar des de l'estat inicial a l'actual
- h és l'estimació del cost d'arribar des de l'estat actual al final

Un algoritme A* (simplificat) seria el següent:

Algoritme A*

```
Estats_pendants.insertar(Estat_inicial);
Actual = Estats_pendants.primer();
Mentre (no_es_final(Actual) y (no Estats_pendants.buit))
    Estats_pendants.esborrar_primer();
    Estats_ja_estudiats.insertar(Actual);
    Seg_moviments = generar_sucesors(Actual);
    Seg_moviments =eliminar_repetits(Seg_moviments,Estat_ja_estudiats,
Estats_pendants);
    Estats_pendants.insertar(Seg_moviments);
    Actual = Estats_pendants.primer();
fMentre
fAlgoritme
```

4.3 - Jocs amb adversari

En el cas dels jocs en que cal enfrontar-se a un altre rival s'utilitzen els arbres de decisió, un model de predicció per representar els estats dels jocs, categoritzar-ne els més avantatjosos per la màquina i obtenir el màxim de possibilitats de victòria.

Per crear l'arbre es genera l'estat inicial del joc com a node superior (serà l'únic estat no apuntat per cap fletxa), es generen tots els possibles estats als que podem arribar mitjançant una única jugada i formaran l'immediat nivell inferior de l'arbre respecte al node pare.

Per cada node generat es segueix generant ara els següents possibles nous estats (ara utilitzant a cada estat totes les possibles jugades possibles permeses per l'altre jugador) i així de forma successiva fins a obtenir la profunditat desitjada.

Com més vegades repetim aquest procés més profunditat tindran les cerques i més acurades seran les decisions preses mitjançant aquest arbre, en el que es valoren mitjançant la funció d'utilitat els nodes del nivell inferior i, a

partir d'aquests valors, s'obtenen els dels nodes superiors fins a l'arrel mitjançant un algoritme per cercar la millor jugada anomenat Minimax.

4.3.1 - Algoritme Minimax

El teorema Minimax creat per John von Neumann (1903-1957) diu que en els jocs bipersonals de suma zero, és a dir amb equilibri dels jugadors entre pèrdues i guanys, on cada jugador coneix prèviament l'estratègia de l'oponent i les conseqüències que té, existeix una estratègia que permet als dos jugadors minimitzar la pèrdua màxima esperada.

Això ens indica que si un jugador, abans de prendre una decisió, avalua totes les possibles jugades amb les que pot respondre el jugador rival i la pèrdua màxima que pot patir amb aquesta resposta, pot jugar de manera que es minimitzi aquesta pèrdua màxima.

Per fer-ho alternarem cada nivell de l'arbre representant els possibles estats als que es pot arribar després d'efectuar el seu moviment/torn cadascun dels dos jugadors, identificant-los com jugador MAX i jugador MIN.

MAX serà qui inicia el joc (suposarem que som nosaltres) i el seu objectiu serà trobar el conjunt de moviments que li proporcionin la victòria independentment del que faci el jugador MIN.

D'aquesta manera, com més elevat sigui el valor d'un estat més favorable serà pel jugador MAX i com més petit, millor pel jugador MIN, etiquetant amb el valor més alt possible la victòria de MAX i amb el més baix la victòria de MIN (i per tant, derrota de MAX).

De la mateixa manera que amb les cerques sense adversari, ens cal una funció d'avaluació heurística que assigni un valor a cada estat del joc per determinar quin és més favorable respecte a la resta.

Cal indicar que, òbviament, no podem generar tot l'arbre per temes d'espai d'emmagatzemament i temps de computació i, per tant, ens haurem de

limitar a una profunditat determinada en la generació i posterior cerca a l'estructura, tenint present que com més alçada tingui més qualitat tindrà la jugada a fer que en puguem extreure.

Un algoritme Minimax seria el següent:

Funció MiniMax(estat) **retorna** moviment

```
millor_mov: moviment;
max, max_actual: enter;
max = -infinít;
per cada mov a moviments_posibles(estat) fer
    max_actual = valorMin(aplicar(mov, estat);
    si max_actual > max llavors
        max = max_actual;
        millor_mov = mov;
    fsi
fper
retorna millor_mov;
```

fFunció

Funció valorMax(estat) **retorna** enter

```
valor_max:enter;
si estat_terminal(estat) llavors
    retorna evaluació(estat);
sinó
    valor_max := -infinít;
    per cada mov a moviments_posibles(estat) fer
        valor_max := max(valor_max, valorMin(aplicar(mov, estat)));
    fper
    retorna valor_max;
```

fsi

fFunció

Funció valorMin(estat) **retorna** enter

```
valor_min:enter;
si estat_terminal(estat) llavors
    retorna avalua(estat);
sinó
    valor_min := +infinít;
    per cada mov a moviments_posibles(estat) fer
        valor_min := min(valor_min, valorMax(aplicar(mov, estat)));
    fper
    retorna valor_min;
```

fsi

fFunció

tenint en compte que:

- sempre es considera que el rival realitzarà la millor jugada disponible (tot i que no sempre serà així)
- la funció estat_terminal determina si l'estat actual és solució (victòria) o estem al nivell màxim de profunditat.

- avaula(estat) representa l'execució de la funció heurística.
- valorMax i valorMin són dues funcions recursives que determinen el valor de les jugades segons sigui el torn de MAX o de MIN.
- es considera (com a convenció) que el valor de l'estat quan guanya MAX és infinit i quan guanya MIN és menys infinit.

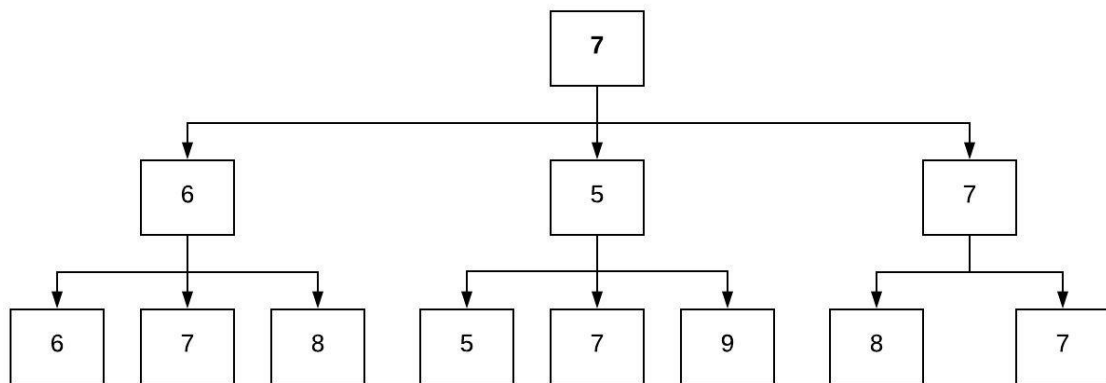


Figura 12: Exemple d'arbre minmax

Com veiem a la figura 2 es valora cada estat del nivell inferior mitjançant una funció heurística, al nivell immediatament superior es valora segons el valor mínim del seu subarbre (6, 5 i 7) i el seu nivell superior amb el valor màxim del seu subarbre, és a dir, 7.

4.3.1.1 - Poda

En jocs com els escacs, on de cada node en poden aparèixer vora 50 nodes fills (tots els possibles taulers als que podem arribar mitjançant una jugada vàlida), un arbre de decisió amb una profunditat important pot arribar a ser enorme i requerir un temps de computació i un espai d'emmagatzemament inviablès, pel que és interessant utilitzar una tècnica de branch and bound (ramificació i poda) que ens permetrà no haver de generar rames que siguin pitjors que altres generades prèviament.

Aquest algorisme, anomenat poda alfa-beta, utilitza el MiniMax vist anteriorment i guarda la informació dels valors de les millors jugades de cada jugador en dos valors anomenats alfa (cota inferior del valor que pot assignar-se com a última opció a un node de màxims) i beta (cota superior del valor que pot assignar-se com a última opció a un node de mínims).

Amb aquests valors acotem el valor d'exploració i valorar si interessa desenvolupar una branca nova o no.

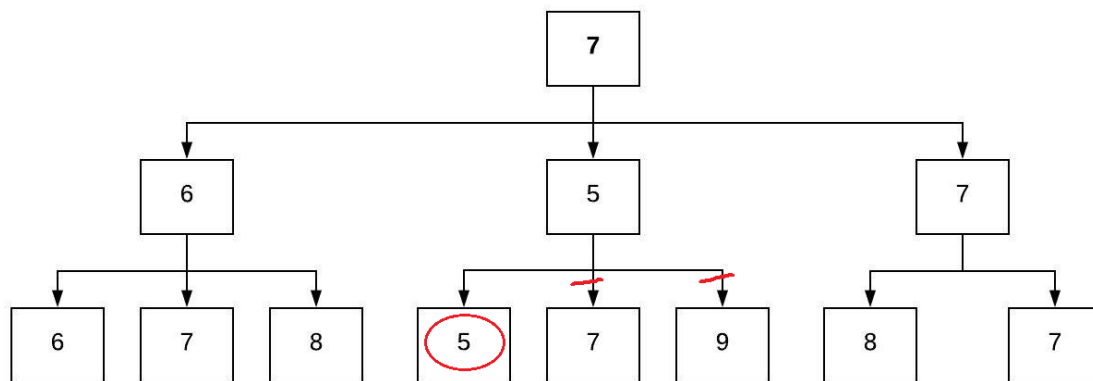


Figura 13: Exemple d'arbre minmax amb poda

Per exemple si observem la figura 3, quan veiem que obtenim un valor inferior al del pare de l'arbre prèviament valorat (quan obtenim un 5 marcat amb un cercle quan al subarbre anterior el mínim ha sigut un 6) ja sabem que el subarbre actual serà pitjor i ens estalviem desenvolupar-lo i, per tant, ja no calcularíem el 7 i el 9.

L'estalvi en temps de càlcul i espai és significatiu quan l'arbre té molts nivells inferiors respecte als nodes superiors als que apliquem la poda, és a dir, si els nodes que no calculem amb el 7 i 9 tinguessin subarbres amb més nivells, que tampoc ja hauríem de recórrer ni valorar.

4.4 - Jocs amb adversari i elements d'atzar

Fins ara hem vist arbres de decisió en jocs sense elements d'atzar, com podrien ser les dames o els escacs, on cada jugador pren una decisió i l'executa però, en jocs que inclouen elements que introdueixen aleatorietat (cartes, daus, etc) cal tenir en compte aquest factor a l'arbre de decisió.

Així doncs, en generar l'arbre hem de tenir en compte la probabilitat de que el jugador pugui realitzar aquella acció i arribar fins al node fill i , cada vegada que li toca moure a un jugador, haurem de considerar primer els diferents possibles valors de l'element aleatori i , per cadascun d'ells, desenvolupar l'arbre de possibles jugades que se'n puguin desprendre.

Posteriorment, per valorar cada node, ens caldrà afegir a la funció heurística la probabilitat d'arribar-hi, ja que podem trobar un valor molt profitós en un node però que tingui una probabilitat molt baixa (seria arriscar pensar que podem arribar-hi) o un altre node menys profitós però més probable (és a dir, més fàcil de que succeeixi).

Per exemple, si implementem un sistema per jugar al parxís totes les jugades que requereixin un valor determinat al dau només es podran realitzar en $1/6$ dels casos. Així doncs, per cada node primer generem els possibles estats si al dau surt un 1, seguit dels possibles estats si surt un 2 i així successivament fins al 6. Un cop generats els estats, calculem el valor de cadascun d'ells considerant que hi haurà una probabilitat de $1/6$ de que siguin una jugada vàlida.

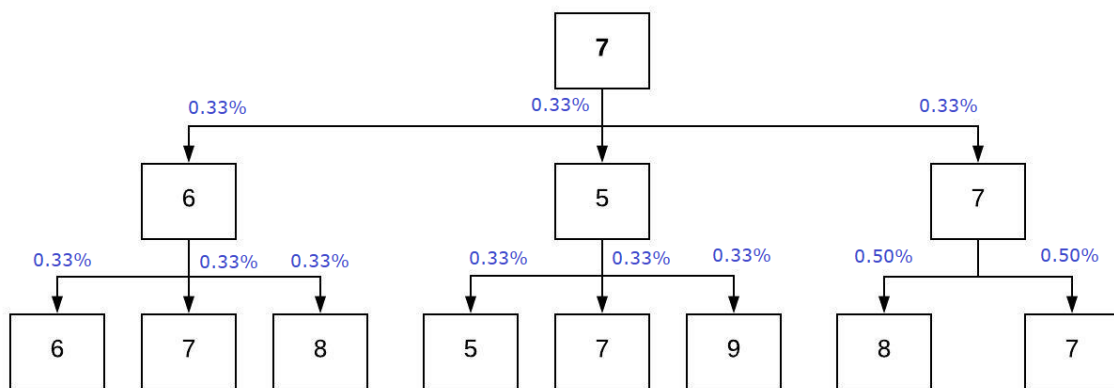


Figura 14: Exemple d'arbre minmax amb elements d'atzar

Si observem la figura 4, veiem l'arbre utilitzat en exemples anteriors però ara cada node valdrà el seu valor multiplicat per la probabilitat de que es produeixi, calcul que se veu ja realitzat a la figura 5.

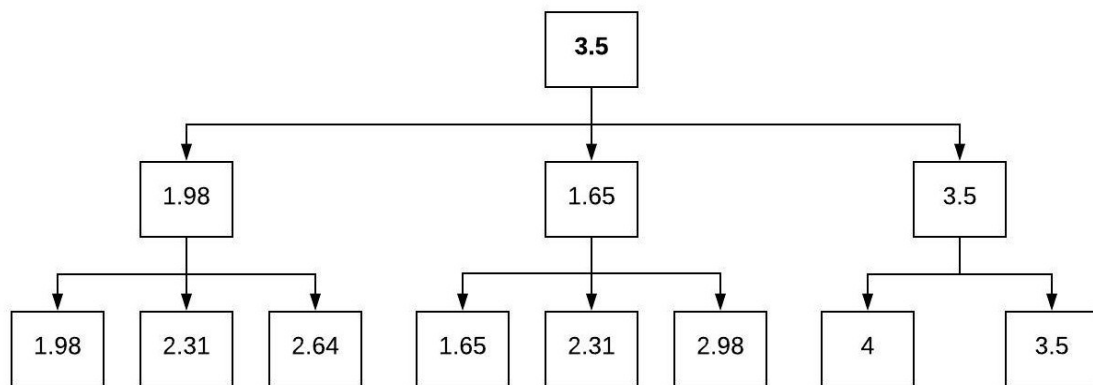


Figura 15: Exemple d'arbre minmax amb l'atzar valorat als nodes

És a dir, ara ja no podem escollir la branca sinó que serà l'atzar qui determinarà a quina situació ens trobarem i, per això, valorem el benefici de cada node tenim en compte la probabilitat d'obtenir-lo.

Això pot fer que la branca millor sigui una altre que sense elements d'atzar ja que un node amb una valoració elevada pot perdre gran part d'aquest valor en funció de la probabilitat que hi hagi per obtenir-lo, és a dir, com menys probabilitat menys probable serà que succeeixi i s'aconsegueixi realment aquell valor.

4.5 - Dragon Slayer

Al contrari del que ens trobem en d'altres jocs amb adversari i que s'han comentat fins ara, sobretot els que tenen elements d'atzar, aquí ens trobem amb una sèrie de diferències que cal remarcar per establir la base amb la que s'implementarà la intel·ligència artificial bàsica (sense personalitat) amb la que s'haurà d'enfrontar el jugador humà.

El programa no haurà de fer sempre el mateix (analitzar la situació i moure una fitxa o tirar un dau) sinó que haurà de prendre tres tipus de decisions totalment diferents segons la situació en la que ens trobem:

- cada vegada que sigui el seu torn:
 - escollir a quin drac ataca i quan el mati decidir

- si continuar amb el risc de perdre tots els punts d'aquell torn
- si plantar-se i puntuar
- quan es planta per puntuar, si el jugador el desafia a continuar
 - escollir si l'accepta
 - escollir si el rebutja
- quan acaba el torn del jugador rival i aquest s'ha plantat per puntuar:
 - decidir si val la pena gastar la única fitxa de desafiament disponible per reptar-lo a continuar o no

Per tant ens trobem amb quatre problemes diferents dintre de la partida que cal afrontar. Els dos primers (si plantar-se o no i, en cas que no sigui així, contra quin drac lluitar en següent lloc) serà la base del joc però segurament sigui amb un ús intel·ligent dels altres dos i la gestió del risc amb el que es decanti la victòria.

A l'hora de generar resposta a cada moment s'utilitzaran les tècniques d'intel·ligència vistes fins ara amb certes modificacions (com es veurà més endavant en aquest mateix apartat) i cada decisió anirà en funció de la situació actual, per exemple, si s'ha plantat per puntuar amb una gran quantitat de punts serà més fàcil que es vegi obligat pel rival a continuar o plantejar-se, quan juga, deixar algun drac fàcil per evitar que el desafiïn o facilitar-li la superació del mateix en cas d'haver de continuar.

Bàsicament, Dragon Slayer és un joc que implementa una mecànica coneguda com "push your luck" (que podríem traduir com: "pressiona la teva sort") en la que a cada torn podem decidir si seguim jugant per acumular més punts però corrent el risc de perdre els que portem en aquell torn i, a més, afegeix el gir dramàtic que (un cop per partida) els jugadors poden desafiar al rival a seguir arriscant-se o no amb les conseqüències que això implica.

Estudiant el joc també observem que, en general, no es tracta tant de "respondre" la jugada de l'adversari sinó de superar els 40 punts quant abans millor per evitar que l'adversari ho faci abans, podent interactuar amb ell només mitjançant els desafiaments i que, segons la puntuació del rival i de qui hagi jugat primer el torn, ens interessarà arriscar-nos més o menys.

De totes maneres, que contingui elements d'atzar no implica que necessàriament hagi de guanyar aquell jugador amb més sort amb els daus sinó que guanya aquell que gestiona millor aquest factor aleatori, minimitzant l'impacte dels resultats negatius i maximitzant els beneficis dels resultats positius.

Així doncs ens cal saber la probabilitat de superar cada drac a cada moment, per tant, calcularem les probabilitats de fer-ho per cada drac i segons el nombre de daus de jugador disponibles (entre 1 i 3) perquè, coneixent prèviament aquestes probabilitats ens estalviem calcular-les a cada situació del joc estalviant-nos temps de computació.

Això ho podem calcular d'aquesta manera degut a que a partir del moment en que iniciem la lluita contra un drac, aquesta continua fins al final sense intervenció del jugador, fins que venç o és derrotat i, per tant, les probabilitats seran fixes per cada drac i per cada nombre de daus de jugador disponibles a l'inici de l'enfrontament.

Degut a la mecànica del joc de tirar daus i retirar tant els que mostren parts del drac com els que perdi el jugador si no es pot defensar d'ells perquè no li han sortit escuts, amb repeticions de tirades fins que o el drac o el jugador siguin derrotats, ens trobem amb un problema matemàtic de càlcul de probabilitat força complexe, pel que es decideix un cop implementat al joc la mecànica d'atac als dracs, executar-les un gran nombre de vegades, acumular els resultats i comprovar en quin % d'ocasions es guanya o es perd en cadascuna de les nou situacions (3 dracs amb 1, 2 i 3 daus de jugador).

S'executen algunes proves amb 100.000 execucions i amb 10.000, es compara i s'observa que la diferencia entre els dos és elevada en temps (entre de 10 i 20 minuts per realitzar les 100.000 execucions) però el càlcul de % de victòria canvia en menys d'un 0'33% i, per tant, es decideix realitzar totes les execucions amb 10.000 casos.

A mode d'exemple es mostra la comparativa entre 100.000 i 10.000 execucions en el cas d'utilitzar 2 daus contra el drac vermell:

Nombre de casos	Temps d'execució	% de victòries calculat
100.000	20 minuts i 14 segons	30.55%
10.000	2 minuts i 2 segons	30.78%

Temps d'execució (en segons)

Daus del jugador	Drac vermell	Drac verd	Drac blau
3	155	137	79
2	122	133	146
1	71	44	83

Índex de Victòries / Derrotes

Daus del jugador	Drac vermell	Drac verd	Drac blau
3	7703/2297	8837/1163	9611/359
2	5893/4107	7320/2680	8823/1177
1	3078/6922	4263/5737	6187/3813

% de victòries

Daus del jugador	Drac vermell	Drac verd	Drac blau
3	77,03%	88,37%	96,11%
2	58,93%	73,20%	88,23%
1	30,78%	42,63%	61,87%

Amb aquests valors calculats sabem les probabilitats de vèncer cadascun dels tres dracs segons el nombre de daus de jugador que hi hagi disponibles.

A banda ens cal obtenir, de les victòries, la probabilitat de que perdem daus de jugadors i quants, així doncs:

Si comencem amb 3 daus:

Daus finals del jugador	Drac vermell	Drac verd	Drac blau
3	39,41%	50,65%	62,27%
2	21,06%	22,89%	23,14%
1	16,56%	14,83%	11,00%
0 (derrota)	22,97%	11,63%	3,59%

Si comencem amb 2 daus:

Daus finals del jugador	Drac vermell	Drac verd	Drac blau
3	0%	0%	0%
2	35,08%	45,68%	59,71%
1	23,85%	27,52%	28,52%
0 (derrota)	41,07%	26,80%	11,77%

Si comencem amb 1 dau:

Daus finals del jugador	Drac vermell	Drac verd	Drac blau
3	0%	0%	0%
2	0%	0%	0%
1	30,78%	42,63%	61,87%
0 (derrota)	69,22%	57,37%	38,13%

A la figura 6 es mostra un diagrama de flux per veure com funciona un torn a Dragon Slayer i identificar els quatre moments en que cal prendre una decisió, concretament:

- a "Escriure drac i lluitar"
- a "El jugador es planta?"
- a "El rival planteja desafiament?"
- a "El jugador accepta el desafiament ?"

Només la de "El rival planteja desafiament" serà presa per l'ordinador al torn del jugador humà mentre que la resta formen part de les decisions que caldrà anar prenent durant el propi torn en si de l'ordinador.

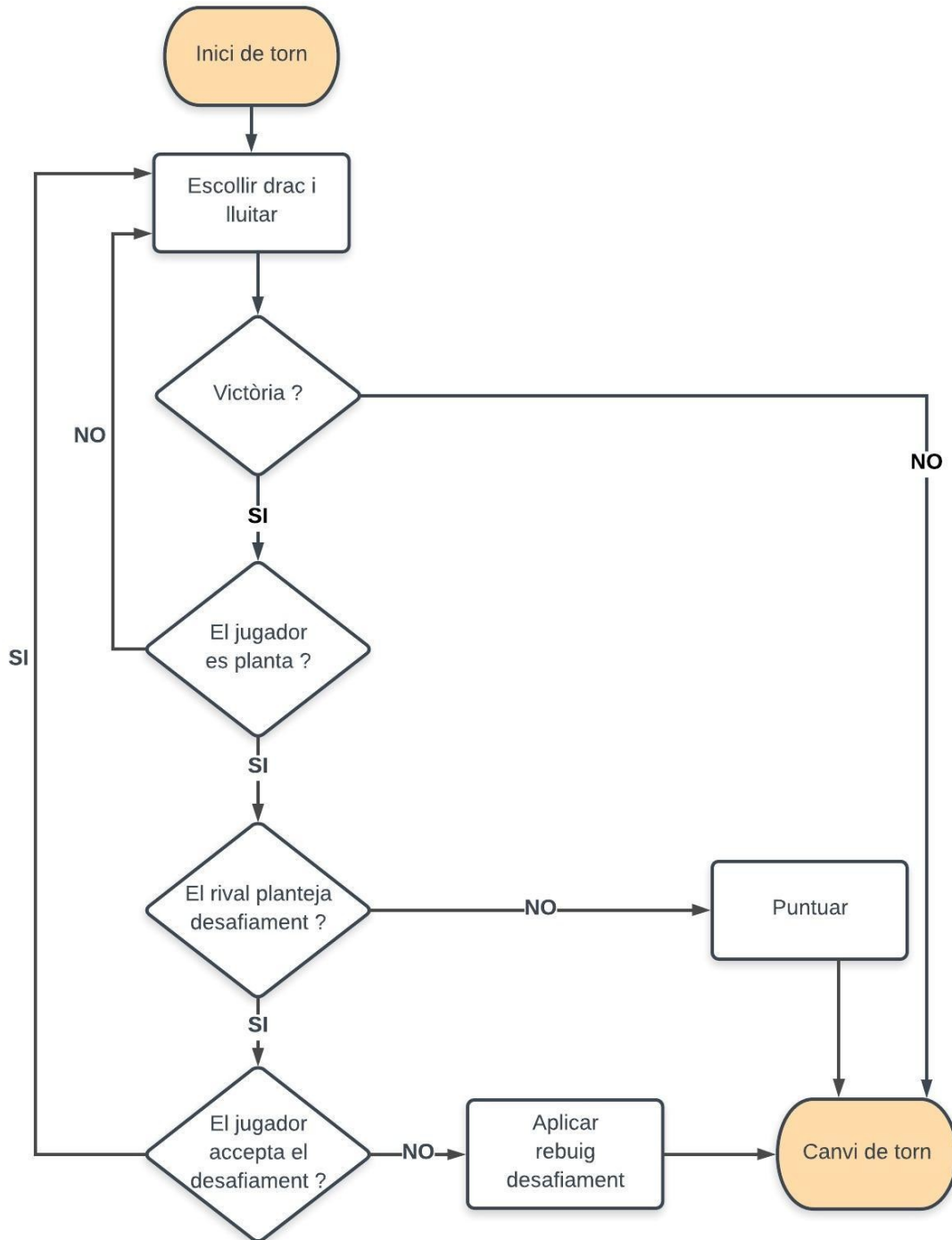


Figura 16: Diagrama de flux d'un torn de jugador a Dragon Slayer

L'arbre de possibles jugades que es va plantejar inicialment va ser el que es mostra a la figura 7, en el que cal escollir un drac per lluitar contra ell o plantar-se (tenint present que no ens podem plantar d'inici sinó després de com a mínim haver derrotat a un drac), i que correspon a la forma en que un jugador humà afrontaria el seu torn.

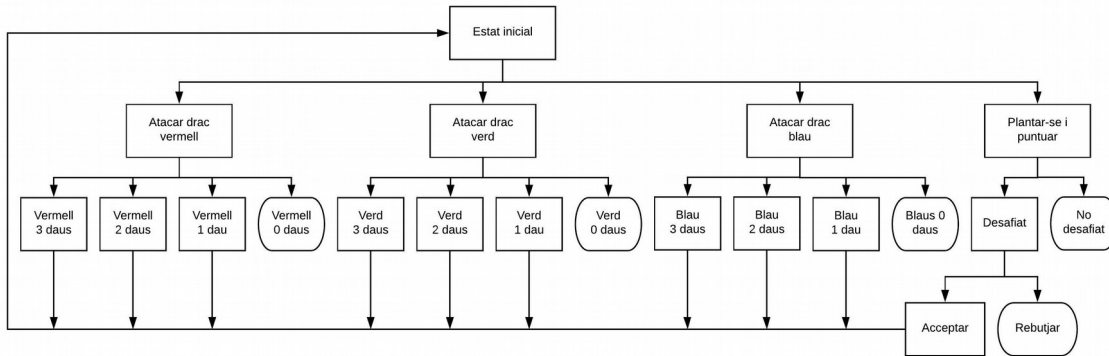


Figura 17: Estructura per decisió d'elecció de drac o plantar-se

Arbre que cal generar cada vegada que l'ordinador ha de decidir si plantar-se o continuar (i amb quin drac fer-ho)

A partir del moment en que el rival ha utilitzat la fitxa de desafiament ens estalviarem generar la última branca de l'arbre i ens quedarà com la figura 8.

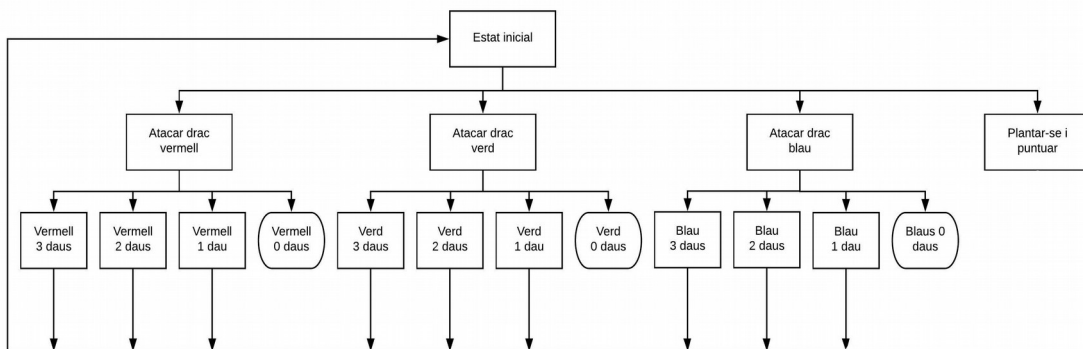


Figura 18: Estructura millorada per decisió d'elecció de drac o plantar-se

Posteriorment es comprova durant el desenvolupament de l'arbre i, en favor d'un millor raonament, que és més adient tot i fer-lo més complexe generar-lo com es mostra a la figura 9.

a) per una banda plantejar-se quin drac atacar (o si rendir-se) i, en cas de lluitar i aconseguir la victòria, si se seguiria jugant o no.

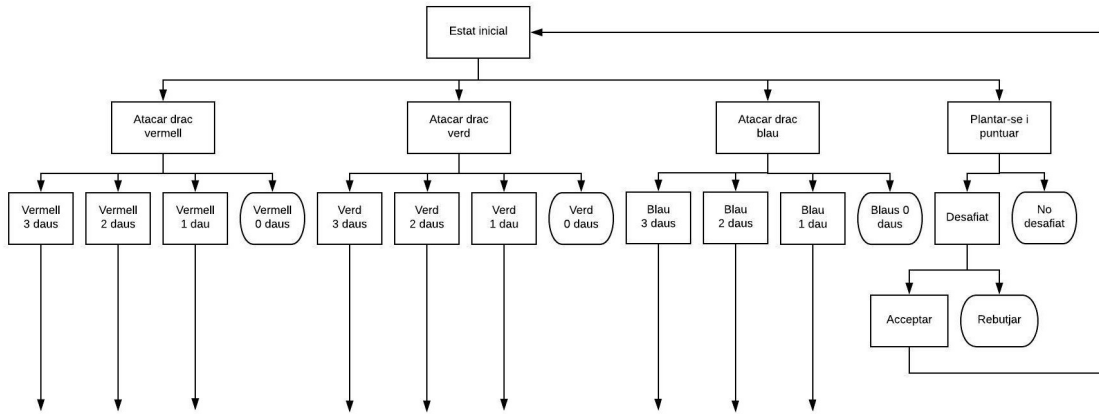


Figura 19: Part superior de l'arbre de decisió

Cadascuna de les 9 possibilitats de victòria ens genera un arbre com el del següent apartat.

b) després de cada victòria i segons els dracs i daus disponibles valorar, en funció de la possibilitat de que l'adversari utilitzés la fitxa de desafiament fent-ne una estimació segons la situació actual de la partida si seguir o no.

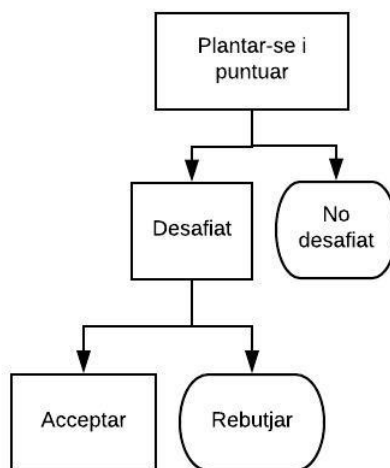


Figura 20: Part inferior de l'arbre

Arbre amb el que valorem cada victòria sobre un drac

De totes maneres, podem aprofitar a generar una petita biblioteca o condicions de jugades prefixades que ens estalviaran temps de càlcul i donaran més eficiència a l'ordinador, com poden ser:

a) quan el jugador humà és el jugador inicial i ha arribat o superat als 40 punts no cal generar cap arbre, ja que l'ordinador disputarà el seu últim torn i ha de jugar fins a superar aquesta puntuació o perdrà la partida.

b) si l'ordinador és el jugador inicial i el jugador humà ha superat els 40 punts però no ha arribat a 45 i es planta, sempre caldrà jugar la fitxa de desafiament si està disponible com a últim recurs per evitar perdre la partida.

c) si es desafia a l'ordinador i rebutjar aquest desafiament dóna la victòria immediata al jugador humà sempre s'acceptarà.

Per aquestes tres situacions límit no ens cal generar cap arbre i, si es produeixen, s'actua en cada cas com s'acaba de definir perquè només hi ha una única opció vàlida per complicat (estadísticament) que sigui evitar-les.

Així doncs, a partir d'aquí, la intel·ligència artificial, a través d'un cert nivell de risc que li associem, anirà jugant temptant la sort fins perdre o plantar-se, deduït en aquest segon cas, segons les condicions de joc si el jugador humà utilitzarà la seva fitxa de desafiament o no i, en cas d'haver-hi forts indicis d'aquest possible ús, es plantarà abans de veure's forçada a seguir i perdre els punts acumulats o renunciar al desafiament i donar excessiva avantatge al rival.

D'aquesta manera per cada decisió (utilització de la fitxa de desafiament contra el jugador humà i, òbviament, per deduir si seguir lluitant i contra quin drac o plantar-se) o suposició (possible ús del desafiament per part del jugador humà en el torn de l'ordinador s'utilitzarà l'estructura d'arbre vista fins ara, tant per part de la màquina com del jugador humà.

5 - Representació del coneixement

Ens cal representar computacionalment la situació del joc en cada moment i mantenir-la emmagatzemada per poder prendre una decisió en funció de la mateixa.

D'entre totes les variables que ens pot caldre tenir en compte per valorar tenim totes les següents:

1. punts actuals de l'ordinador.
2. punts actuals del jugador humà.
3. punts acumulats en el torn en curs pel jugador actual (ja sigui l'ordinador o el jugador humà).
4. identitat del jugador inicial: per saber si se superen els 40 punts si l'altre jugador té un últim torn o no
5. fitxa de desafiament disponible per l'ordinador.
6. fitxa de desafiament disponible pel jugador humà.
7. dracs vius i el seu tipus.
8. nombre de daus de jugador disponibles.

Durant la partida hi ha tres decisions a prendre que ens cal implementar:

1. plantar-se o seguir: es pren després de la mort de cada drac i és una decisió que cal prendre constantment durant tota la partida per part dels dos jugadors al seu torn.
2. desafiar o no al rival: és una decisió que (mentre es tingui la fitxa de desafiament, que només es pot utilitzar una vegada per partida) cal prendre al final de cada torn de l'adversari.
3. acceptar o rebutjar el desafiament del rival: situació que es dona quan l'adversari utilitza la seva fitxa de desafiament i, per tant, és una decisió que només cal prendre (com a molt) un cop per partida.

Coneixent totes les variables en joc anem a classificar-les segons a quina decisió les hem de tenir en compte:

Plantar-se o seguir	Desafiar o no al rival	Acceptar o rebutjar un desafiament del rival
1. punts actuals de l'ordinador. 2. punts actuals del jugador humà. 3. punts acumulats en el torn en curs pel jugador actual. 4. identitat del jugador inicial. 5. fitxa de desafiament disponible per l'ordinador. 6. fitxa de desafiament disponible pel jugador humà. 7. dracs vius i el seu tipus. 8. nombre de daus de jugador disponibles.	1. punts actuals de l'ordinador. 2. punts actuals del jugador humà. 3. punts acumulats en el torn en curs pel jugador actual (humà). 4. identitat del jugador inicial. 5. fitxa de desafiament disponible per l'ordinador. 6. fitxa de desafiament disponible pel jugador humà. 7. dracs vius i el seu tipus. 8. nombre de daus de jugador disponibles.	1. punts actuals de l'ordinador. 2. punts actuals del jugador humà. 3. punts acumulats en el torn en curs pel jugador actual (ordinador). 4. identitat del jugador inicial. 5. fitxa de desafiament disponible per l'ordinador. 6. fitxa de desafiament disponible pel jugador humà. 7. dracs vius i el seu tipus. 8. nombre de daus de jugador disponibles.

(*) en vermell les variables que no es tenen en compte en aquella decisió

Com veiem, tot i que algunes variables es tindran en compte en algunes decisions si i altres no, s'han de valorar gairebé totes per escollir la més encertada possible, tot i tenir en compte que un cop escollit un camí, a partir d'allí intervindrà la sort, amb unes probabilitats que ja tindrem prèviament calculades.

D'aquesta manera es crea una estructura en Gambas anomenada *estat* que contindrà tota aquesta informació necessària per conèixer l'estat del joc i decidir en conseqüència:

Public Struct estat

```

daus_jugador As Integer
puntuacio_jugador As Integer
puntuacio_ordinador As Integer
puntuacio_provisional As Integer
jugador_inicial As Integer
fitxa_desafiament_ordinador As Boolean
fitxa_desafiament_jugador As Boolean
drac_vermell_viu As Boolean
drac_verd_viu As Boolean
drac_blau_viu As Boolean

```

End Struct

Cada node dels arbres vistos fins ara contindrà tota aquesta informació que anirà canviant en funció de les operacions que ens porten d'un node pare a cada node fill corresponent.

La informació sobre quin és el jugador inicial no es guarda als nodes perquè és constant durant la partida i, per tant, no variarà d'un node a un altre en la generació de l'arbre.

6 - Personalitat

Com s'ha comentat a l'apartat 1.2.2 del present document, segons Richard Bartle i tal i com es mostra a la figura 11, existeixen bàsicament 4 tipus de jugadors segons el que més els motiva a l'hora de participar en un joc, ja que a no totes les persones els atreuen els mateixos factors.



Richard Bartle <http://www.mud.co.uk/richard/hclds.htm>

Drawing: Frank Caron <http://frankcaron.com>

Understanding behavioral patterns in player types, and what features support the behavior desired is also useful. Maybe more useful.

Figura 21: Principals tipus de jugadors segons Richard Bartle

Un joc amb les característiques del Dragon Slayer no atraurà als jugadors explorers, perquè el joc no els ofereix els que ells valoren com a interessant en una experiència lúdica (descobrir una història, nous mapes, localitzacions o fases, etc) i tampoc podem implementar a l'ordinador un comportament socializer perquè, tot i que es podria simular que al programa li diverteix el joc i la interacció amb l'usuari, no tindria efecte sobre el seu comportament durant la partida. De totes maneres, un joc com el Dragon Slayer té força més atractiu per jugadors achievers però, sobretot, pels killers, ja que les mecàniques del joc que ens ocupa segueixen el tipus de disseny que atreu a aquest tipus de jugadors, especialment a aquests últims.

6.1 – Característiques dels jugadors

Analitzant doncs les característiques d'aquests perfils de jugadors i les mecàniques i tipus de decisions que cal prendre durant una partida de Dragon Slayer, trobem un seguit de factors que influeixen en la forma de valorar la manera d'afrontar cada nova situació i actuar en conseqüència, sempre tenint en compte l'objectiu final d'aconseguir la victòria. Aquests factors serien els següents:

- Aversió al risc: qualificarem així al jugador conservador, que prefereix assegurar les puntuacions i no intenta aconseguir-ne de més elevades per torn si això implica una probabilitat (més o menys) elevada de perdre els acumulats.
- Anàlisi paràlisi (també conegut com AP): temps excessiu que dedica un jugador a decidir quina jugada executar entre les disponibles. Es produeix un bloqueig en el que s'analitzen les situacions i, davant el dubte, es tornen a analitzar de nou en diverses ocasions sense decidir-se per cap.
- Calculador: un jugador competitiu busca la victòria i juga valorant totes les opcions i triant sempre la més eficient segons el seu criteri. És el tipus de rival més complicat de vèncer.
- Intuïtiu: per ser capaç de preveure, segons la situació actual de la partida, les accions del rival en el seu proper torn i actuar en conseqüència.
- Nervios: un jugador intranquil intentarà, no només guanyar, sinó evitar situar-se massa punts per darrera l'adversari per no trobar-se amb una diferència de punts molt complicada de superar als torns finals.

6.2 – Implementació de les característiques

Implementarem les diferents personalitats basant-nos en les característiques acabades d'enumerar de la següent manera:

- Aversió al risc: com més elevada sigui menys s'arriscarà i jugarà d'una forma més conservadora, valorant més els punts que porti acumulats en aquell torn que els que potencialment pugui obtenir atacant al següent drac.

- Per representar aquesta característica de la personalitat utilitzarem les següents dues variables:
 - *valor_maxim_node_triar_drac*: farà de fita superior per valorar el risc de lluitar contra un drac determinat, així, com més elevat sigui aquest valor més limitarem el risc disposat a córrer i es valoraran els possibles nodes, i per tant, decisions a prendre de forma corresponent.
 - *valor_maxim_node_decidir_acceptar_desafiament*: farà de fita superior quan es valori si acceptar la fitxa de desafiament i tindrà un funcionament idèntic a l'anterior durant la generació de l'arbre però en la situació de l'ús potencial de la fitxa de desafiament per part del jugador humà, és a dir, tenint en compte el valor doble dels enemics vençuts.

- Anàlisis paràlisis: mitjançant aquest paràmetre simularem a l'ordinador un temps de reflexió abans de decidir la següent jugada o la resposta a una petició de desafiament per part del rival. Serà un valor no gaire elevat per generar emoció i per evitar el principal problema d'aquesta característica, que genera desesperació en l'adversari mentre espera.
 - Per simular el fet de que l'adversari està valorant les seves opcions i decidint què fer simulant un ritme de partida normal, tot i que l'ordinador ho tindrà calculat gairebé de forma immediata, s'afegeix una variable que afegeix un temps d'espera determinat a les respostes de l'ordinador davant cada possible acció a realitzar.

- Calculador: mitjançant la intel·ligència artificial implementada tots els rivals tindran la capacitat de preveure futures jugades però farem que aquesta sigui més o menys encertada segons la capacitat de càlcul que simularem a cada adversari.
 - Per representar aquesta característica als adversaris i, segons la personalitat de cadascú, escollirem un node més o menys bo com a elecció entre totes les disponibles, sempre dintre d'uns paràmetres de qualitat en el raonament, és a dir, mai s'escollirà una opció dolenta en cas de trobar-la.

- Intuïtiu: calcularem més o menys encertadament la probabilitat de que l'adversari utilitzi la fitxa de desafiament durant el nostre torn, forçant-ho

a que si ho fa es trobi en la pitjor situació possible i, per tant, més favorable per l'ordinador.

- Se simularà la intuïció de l'ordinador per preveure l'ús de la fitxa de desafiament per part del jugador humà desenvolupant el seu arbre de possibilitats, valorant els nodes i utilitzant una variable anomenada *valor_maxim_node_decidir_desafiar* per realitzar aquesta previsió amb més o menys encert.
- Nerviosos: com més elevat sigui aquest paràmetre més és precipitarà el programa en prendre la decisió, sobretot en funció de la distància en punts que el pugui separar provisionalment del rival i, per tant, de la victòria.
 - Es representarà aquesta característica de la personalitat a l'ordinador valorant els nodes (amb la funció d'utilitat) tenint en compte la distància en punts entre els dos jugadors i la puntuació realitzada al torn previ pel jugador humà.

En alguns casos és poden confondre comportaments ja que, per exemple, en el torn posterior a un en el que el jugador humà hagi aconseguit una elevada puntuació, possiblement s'arrisquin tant un jugador amb poca aversió al risc com un altre amb un elevat factor de nerviosisme, tot i que per motius diferents, o que un jugador poc calculador però amb una elevada aversió al risc decideixi córrer un risc similar al de un jugador molt més arriscat degut a que el primer ho farà perquè pensa que la seva opció no és tant arriscada de com és realment i, el segon, tot i ser conscient del risc que correrà prefereix fer-ho per aconseguir una puntuació més elevada. Aquestes similituds poden aparèixer en determinats moments de la partida tot i que això també succeiria en el cas d'enfrontar-nos a diferents rivals humans amb aquestes diferents característiques de personalitat.

Tot i que segons els factors de personalitat que adoptem poden fer que el programa jugui de forma millor o pitjor i prengui decisions més o menys encertades, sempre jugarà per guanyar i la intel·ligència artificial serà capaç de trobar (tenint en compte el factor aleatori que impliquen els daus) la millor jugada possible, tot i que no escollirà aquella per nervis, aversió al risc, manca d'intuïció, etc.

6.3 – Personalitats implementades

Així doncs, amb els paràmetres de personalitat que acabem de veure es generen 4 jugadors virtuals diferents (Abel, Beatriu, Carla i Daniel) amb les següents característiques:

- Abel: és un jugador intel·ligent al que s'ha deixat associada la intel·ligència artificial del joc sense cap tipus de modificació i, per tant, juga de manera òptima en cada moment amb la victòria com objectiu en tot moment tot i que aquesta, com hem vist al Dragon Slayer, també depèn de la sort amb els daus.

En definitiva, aquest jugador seria el màxim de calculador i intuïtiu i tindria uns valors mitjos de nervis i aversió al risc perquè no afectessin a les seves decisions.

Cal remarcar que el fet de que sigui calculador vol dir que triarà sempre la millor jugada possible coneixent el seu risc, tot i que el resultat d'aquesta, degut a l'atzar pugui ser negatiu per ell i que el fet de que sigui intuïtiu fa que pugui predir amb més possibilitats d'encert les reaccions del jugador humà segons l'estat del joc però no que ho faci realment ja que aquest té el poder de la lliure elecció.

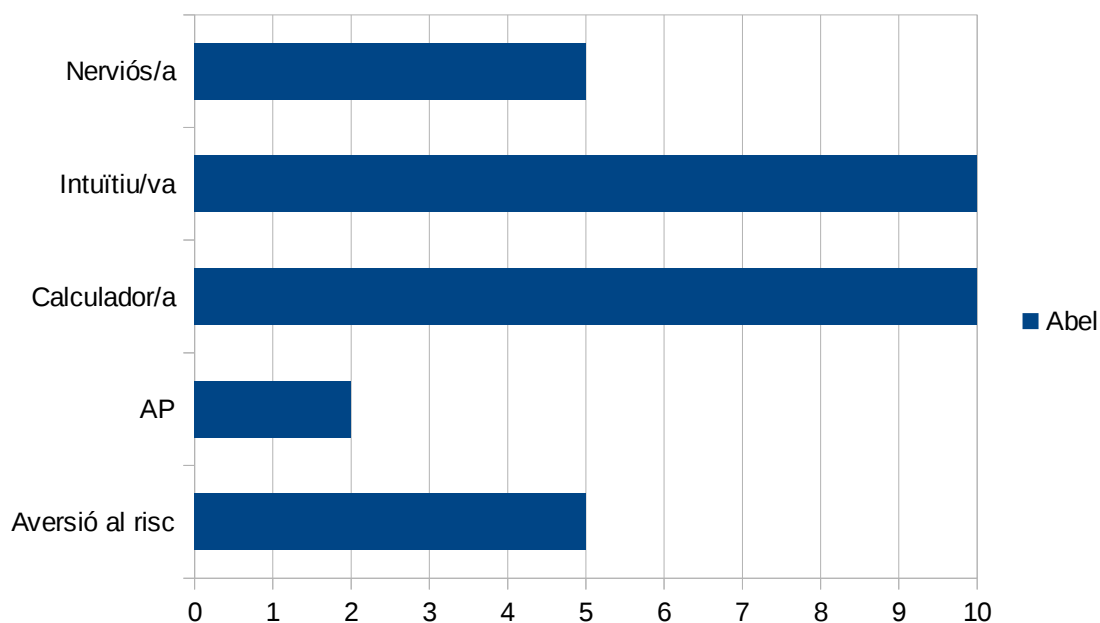


Figura 22: Gràfic de la personalitat de l'Abel

- Beatriu: és una jugadora intel·ligent però li costa decidir-se i, a més, té una elevada aversió al risc, amb el que jugarà de forma conservadora, tant a l'hora d'escollir si seguir lluitant contra els dracs a cada torn o plantar-se com, també, al moment de decidir si accepta o no el desafiament plantejat pel jugador humà. A més disposa de la intuïció suficient com per considerar la probabilitat d'ús d'aquest desafiament al final del seu torn.

Degut a la naturalesa del joc, la Beatriu possiblement vagi anotant punts (tot i que pocs) a cada torn tot perquè al no arriscar-se en excés difícilment aconseguirà en un únic torn una puntuació elevada però tampoc perdrà per temptar la sort.

Gràcies a no puntuar en excés en la majoria de torns fa que costi desfiar-la ja al jugador humà li costarà trobar l'ocasió de poder treure-li rendiment a la fitxa de desafiament.

Tot plegat la fa una rival a tenir en compte perquè si ens arrisquem per superar-la i no tenim sort ella anirà de forma més aviat lenta però segura cap a la puntuació de 40 que li donarà la victòria.

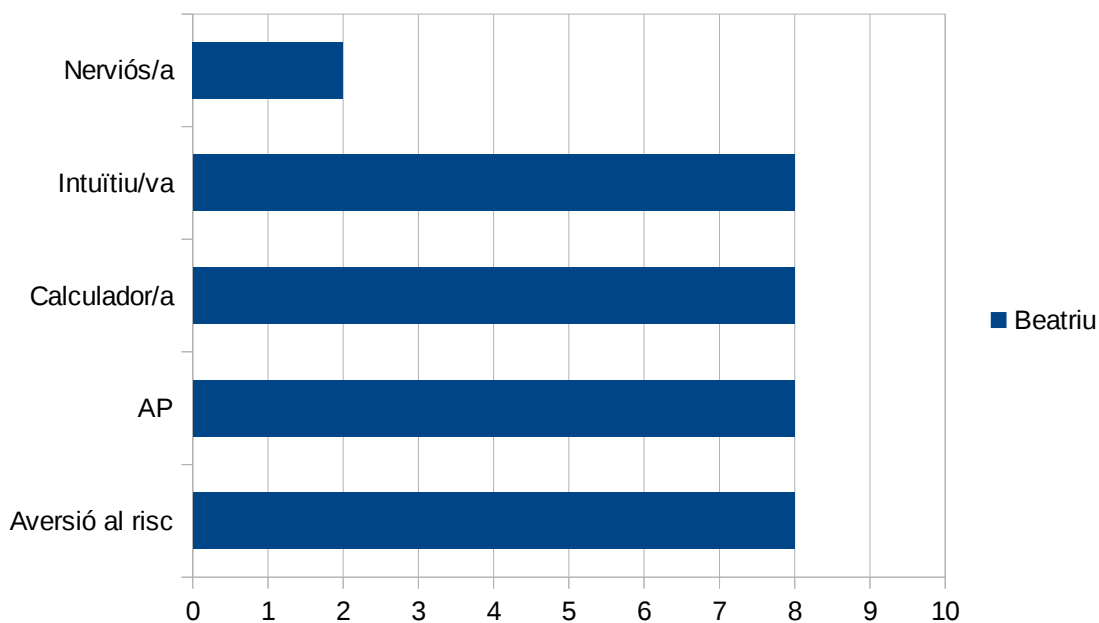


Figura 23: Gràfic de la personalitat de la Beatriu

- Carla: aquesta adversària jugarà arriscant-se més que la resta de rivals vistos fins ara, cosa que li pot donar força punts si té sort en les seves tirades però molts cops pot perdre punts acumulats només per l'ambició d'aconseguir-ne més. Això li permet jugar sense haver de pensar-se massa les seves decisions i, quan guanya, ho fa amb una diferència de punts respecte al rival molt important.

A més, intueix força bé les intencions de l'adversari pel que fa a l'ús de la fitxa de desafiament, sobretot perquè a l'arriscar-se força ens trobarem que el jugador humà, al contrari que amb la Beatriu, tindrà molt clar quan utilitzar la fitxa de desafiament (en el torn on el risc que ha assumit la Carla li hagi servit per aconseguir una puntuació força elevada)

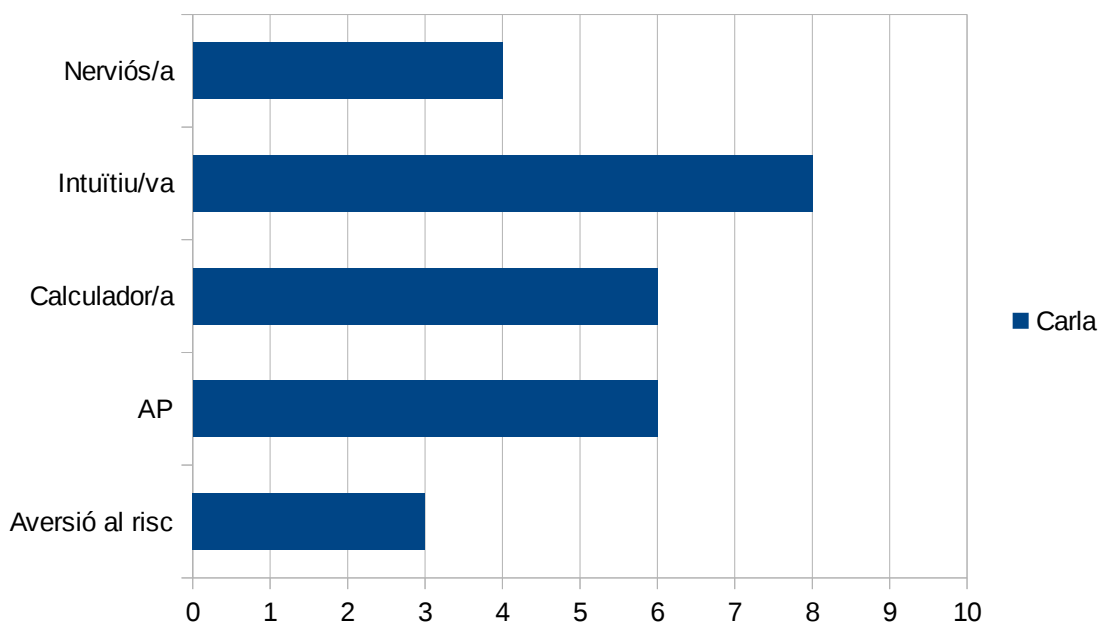


Figura 24: Gràfic de la personalitat de la Carla

- Daniel: és un jugador força nerviós i amb poca aversió al risc, pel que si veu que va perdent o l'última jugada del seu rival ha sigut força bona intentarà temptar molt la sort per aconseguir reduir aquesta diferència sense jugar de forma conservadora que, a llarg termini, pot ser una estratègia més adequada. No és massa intuïtiu ni es pensa massa les jugades a realitzar.

Sempre que es precipiti en una jugada serà més per nervis i no per ganes d'arriscar-se, cosa que també el fa un jugador que, per una banda, és relativament previsible (si el posem nerviós puntuant més que ell s'arriscarà i sinó no) però a la vegada és un dels més capaços, sobretot si té sort en les tirades, de remuntar la partida al jugador humà.

De totes maneres, conscient de les seves pròpies limitacions, en Daniel intentarà sempre que pugui fer ús de la fitxa de desafiament contra el seu rival per evitar que s'escapi en el marcador de puntuació, situació que sap que l'incomoda i li genera els nervis que li fan prendre decisions no sempre encertades ja que, en la majoria de partides, l'estratègia de puntuar poc però sovint és més beneficiosa que la d'intentar aconseguir una gran puntuació a cada torn pel risc de perdre'l de forma continuada.

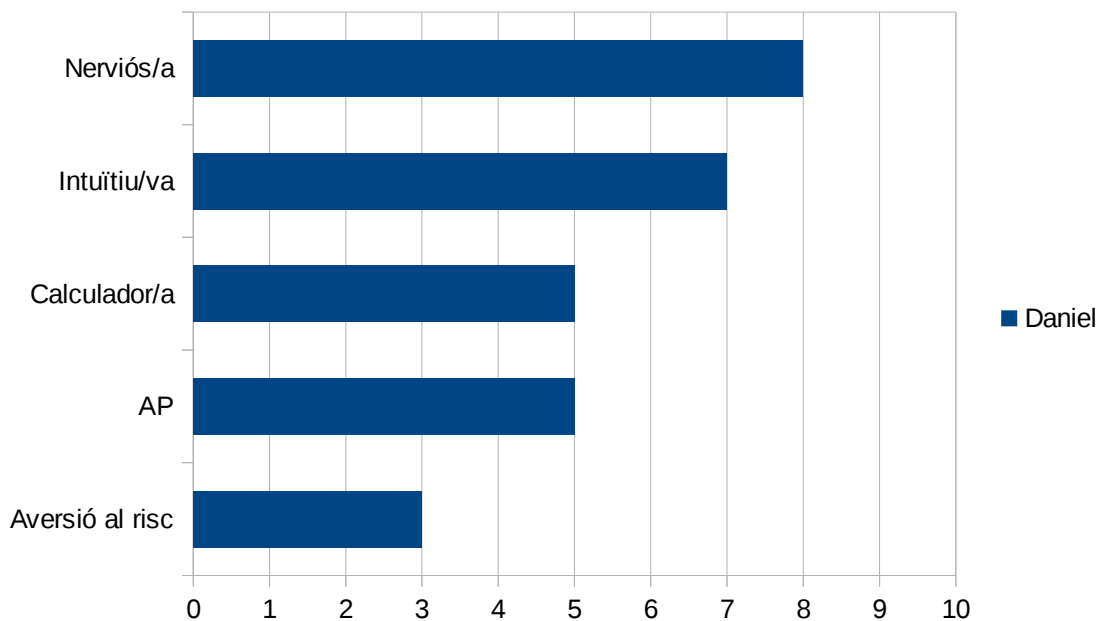


Figura 25: Gràfic de la personalitat del Daniel

Abans d'iniciar la partida el jugador humà pot triar contra quin dels quatre contrincants enfrontar-se mitjançant un desplegable tot i que, en cap moment, se l'informa de les característiques de cadascun, sinó que ho anirà descobrint a mesura que hi vagi jugant, de la mateixa manera que succeeix quan es juga contra altres jugadors humans.

Segons la personalitat de cada rival s'utilitzaran funcions d'utilitat diferents per valorar els nodes de l'arbre generat.

7 - Disseny de la GUI

Tot i no ser l'objectiu principal del projecte cal dissenyar una GUI (*graphical user interface*) agradable a l'usuari i que li permeti tant jugar com interactuar amb el programa.

Les principals característiques d'una bona GUI són les següents:

- Familiaritat de l'usuari: la GUI ha d'utilitzar paraules i imatges conegudes per l'usuari i estar relacionades amb l'àmbit que representa.

S'utilitzen els gràfics i el disseny visual del joc de taula de tots els components: daus, taulers de puntuació, etc.

- Uniformitat d'estats: tant les comandes com els menús han de tenir el mateix estil i format.

Se simula una taula amb els components del joc situats de la forma més similar a com ens les trobaríem si hi juguéssim en format físic amb algú altre.

- Recuperació d'estats: és un dels principis més importants al dissenyar una GUI ja que cal entendre que l'usuari pot cometre errors i, per tant, el sistema ha de ser capaç de proporcionar-li alguna eina per solucionar-los o permetre'l tornar a un estat previ.

Es situen els elements de joc suficientment separats i diferenciats per evitar errors de l'usuari ja que un cop tirats els daus no es podrà rectificar cap decisió i, per evitar la incomoditat durant la partida de verificar la decisió del jugador, només es preguntarà per assegurar una decisió presa en moments molt puntuals (com pot ser l'ús de fitxa de desafiament).

- Guia d'usuari: la GUI ha de proporcionar al jugador assistència indicant perquè serveix cada element de la mateixa i quines accions té disponibles en cada moment, de forma clara i sense saturar amb un excés d'informació a l'usuari.

Es configuren tres sistemes per mantenir informat a l'usuari:

- *Missatges pop-up: informen i permeten deixar temps de reflexió al jugador per observar la pantalla abans de que l'ordinador segueixi jugant.*
- *Log: on es detalla tot el que va succeint durant la partida i, quan s'espera que l'usuari realitzi alguna acció sempre se l'informa mitjançant aquest sistema.*

- *Ajudes: situant el cursor del ratolí sobre qualsevol element del joc s'explica breument per a què serveix i quina funció té.*
- Retroalimentació: la GUI ha de donar resposta a les accions (ressaltant l'opció escollida, comunicar l'èxit o fracàs d'una acció, reflectir la situació actual, etc).

Els elements de joc es van movent per la taula i, amb un cop d'ull, el jugador pot veure ràpidament quin és l'estat actual de la partida: els daus eliminats se situen a part, uns cubs marquen la puntuació, els dracs vençuts apareixen a part, etc.

- Minimitzar la necessitat de memorització: utilitzar controls gràfics, limitar la càrrega d'informació a curt termini i proporcionar la necessària en cada moment amb símbols auto explicatius i fàcils de recordar.

S'utilitza sempre que es pot elements gràfics i no botons per interactuar amb el joc, els daus representen visualment tematitzat quin és el seu resultat i quines implicacions té en el joc, s'afegeixen uns cubs de puntuació fantasmes (amb colors més clars) que indiquen la puntuació total que aconseguiria el jugador actual si es plantés, etc.

- Percepció i legibilitat: els elements han de mostrar-se distribuïts, sense sobrecàrrega i amb un contrast adequat amb l'entorn.

Seguint aquesta premissa es distribueixen els elements de joc i es configura un fons (que simula una taula) de colors relativament clars per ressaltar daus, taulers i fitxes.

Finalment s'han afegit a l'inici de la partida la possibilitat de jugar de dues formes diferents segons l'experiència del jugador humà:

- interactiva: es mostren finestres que demanen l'acceptació de l'usuari amb missatges sobre la situació actual i quines decisions està prenent el rival. Això permet adequar el ritme de partida al del jugador humà i pot ser útil en les primeres partides per entendre les mecàniques de joc. A més de a les finestres que interrompen la partida, totes les accions i situacions del joc es registren també al log.
- Automàtica: es mostra al log de l'entorn gràfic informació sobre el desenvolupament de la partida però aquesta no s'atura en cap moment per mostrar cap tipus d'informació. D'aquesta manera la sensació és més semblant a jugar una partida en un entorn real tot i que aquesta opció de joc requereix que el jugador humà conegui les regles i les situacions que es poden anar produint durant la partida.

A la figura 26 es mostra el disseny final de la GUI del joc amb els elements distribuïts a la pantalla mentre que a la figura 27 es mostra com apareixen els elements del joc durant el desenvolupament d'una partida (log, daus, missatges, estat, etc).

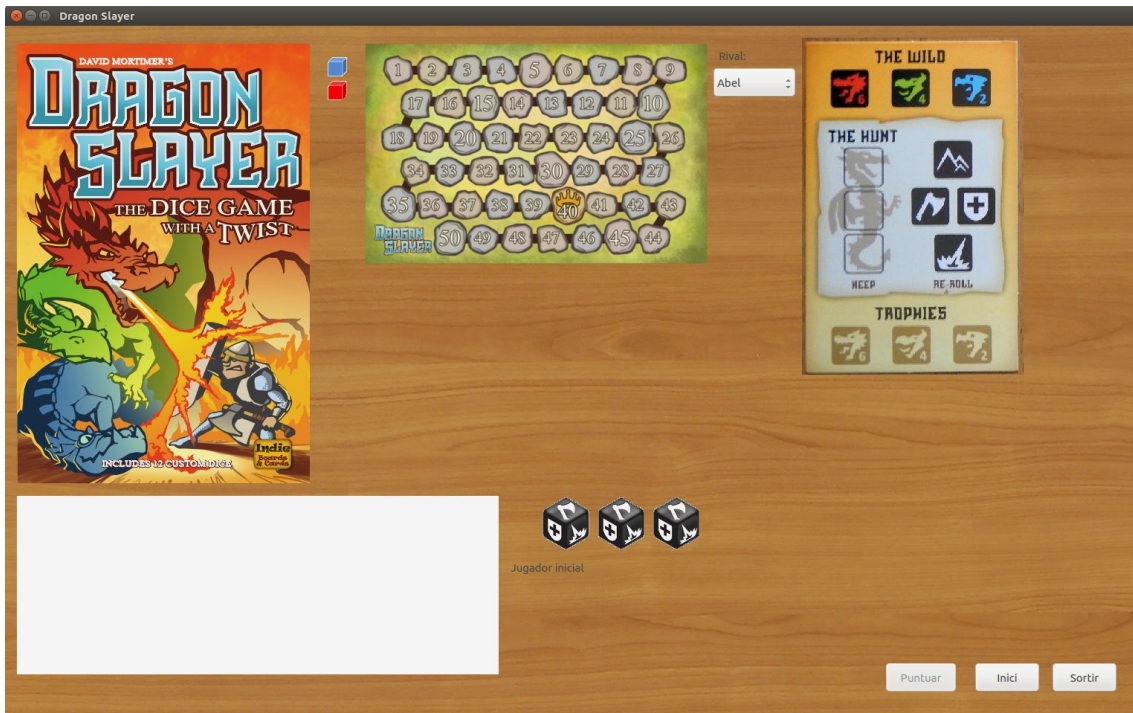


Figura 26: Disseny de la GUI del Dragon Slayer

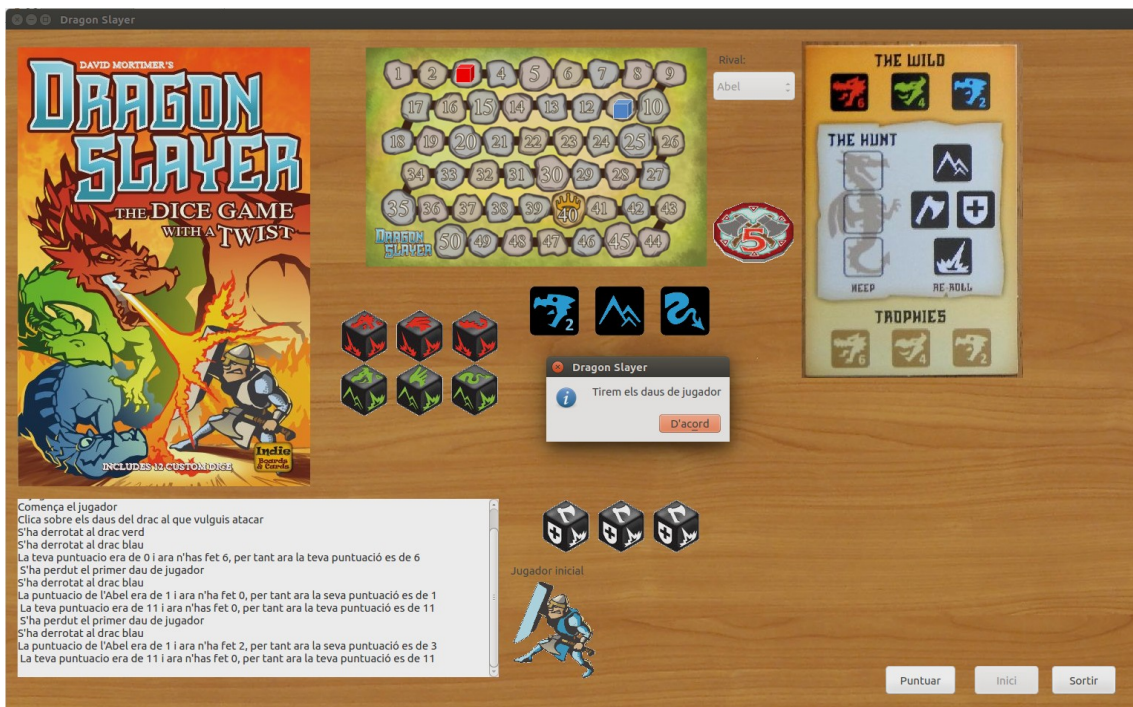


Figura 27: Ús de la GUI en el transcurs d'una partida

8 - Conclusions

Gràcies a la realització d'aquest treball s'ha treballat i aprofundit, per una banda, la complexitat de planificar i realitzar des de zero un projecte que toca diferents àmbits (videojocs, intel·ligència artificial, tipus de jugadors segons els estudis de gamificació, etc) amb un equip d'una sola persona.

Per altre banda, tot i que per mecàniques, el Dragon Slayer no requereixi de cerques de gran profunditat degut a l'impacte de la probabilitat en cada decisió, la que requereix ja fa intuir que si l'atzar tingués menys influència seria fàcil trobar-nos amb un nombre intractable d'estats i requerir de poda (o altres sistemes) per poder generar les jugades en un temps i unes necessitats de memòria ajustades al ritme de joc i que, encara que no sigui incorporant diferents personalitats a una intel·ligència artificial si que sembla interessant que aquesta tingui diferents formes de jugar (més d'atac, més conservadora, etc) i no se centri sempre única i exclusivament en l'obtenció del resultat òptim en entorns com el lúdic.

Un cop finalitzat el treball, en principi s'han aconseguit els objectius prèviament planificats, ja que s'ha implementat un joc, se li ha afegit un rival que disposa d'una intel·ligència artificial que suposa un repte intentar vèncer i, a més, podem triar diferents personalitats perquè els adversaris tinguin comportaments més variats que el "simple" triar la millor opció en cada moment.

És cert que, possiblement, en un altre tipus de joc seria més notable aquesta diferenciació ja que aquí, amb el factor atzar com una característica més del joc pot passar que amb una elecció bona es perdi el torn i els punts acumulats o amb una de no tant bona s'aconsegueixi una puntuació elevada o que no s'escolleixi una opció ja sigui per aversió al risc com per nerviosisme (evitant la diferenciació en jugades determinades d'algunes característiques de la personalitat) tot i que durant la partida queda clarament diferenciat cada rival al que ens enfrontem.

De totes maneres, és positiu que Dragon Slayer incorpori un component d'atzar perquè, el fet d'haver-lo de valorar, permet afegir una característica més a la personalitat a la que s'enfronta el jugador humà.

Cal apuntar que la inexperiència inicial en aquest tipus de treballs ha fet que calgués anar ajustant la planificació al llarg del projecte (sempre mantenint la primera com històric del plantejament inicial) per adaptar-la a la situació actual de l'estat a cada entrega.

Tot i que la metodologia prevista ha estat força adequada, hi ha hagut tasques que han requerit un temps de treball superior al planificat a l'inici del projecte, bàsicament degut a que la intel·ligència artificial del programa ha de prendre un seguit de decisions en moments molt diferents per les quals s'ha hagut de replantejar la generació de l'arbre i la forma de valorar els nodes, ja que no es tracta de l'habitual joc en que cada jugador mou una fitxa per torn i li toca al rival sinó que hi ha canvis de torn inesperats, moments en que cal interrompre al rival, respondre a desafiaments, jugar amb la probabilitat, etc.

Degut a les mecàniques pròpies del joc, sobretot al fet de que després de cada drac podem realitzar una nova elecció (seguir o plantar-nos) ha fet que els arbres amb més nivells planificats inicialment obtinguessin els mateixos resultats que els utilitzats finalment.

Finalment, des d'un punt de vista del joc com entreteniment es podrien haver realitzat les següents millores:

- actualment el joc està implementat per jugar un humà contra l'ordinador, però es podria incrementar el nombre de jugadors per acceptar-ne més, ja siguin humans o portats per l'ordinador i assignar una personalitat diferent a cadascun d'ells.
- les partides es realitzen en un mateix ordinador i es podria implementar el joc en xarxa per jugar amb altres jugadors a través d'Internet.

- a mesura que es juguen partides es podrien guardar estadístiques de victòria i comportament de cada personalitat i, d'aquesta manera, comprovar quina és la que té un índex més elevat de victòries. A més, de la mateixa manera, mitjançant un login d'usuari (per si hi juguen humans diferents), es podrien guardar estadístiques del jugador i aprendre del seu comportament, amb el que algunes suposicions que fa l'ordinador, com pot ser la possibilitat de que el jugador humà utilitzi la fitxa de desafiament en un moment donat, serien més acurades i tindrien un fonament basat en l'experiència.

- aprofitant el log de text on es va mostrant el desenvolupament de la partida es podria crear una comunicació unidireccional amb el jugador, simulant pensaments o comentaris en veu alta del rival segons la seva personalitat per augmentar la diferenciació que un humà pot percebre entre els diferents tipus d'adversaris implementats i, a la vegada, incrementar la diversió.

9 - Glossari

- **AP** (o Anàlisi paràlisi): temps excessiu que tarda un jugador a dur a terme la seva jugada degut a que analitza totes les possibles, no es decideix per cap i torna a analitzar-les de nou un determinat nombre de vegades fins que, finalment, en selecciona una.
- **BGG** (Board Game Geek): base de dades que recull tota la informació disponible sobre jocs de taula de tot tipus i, de cadascun d'ells en particular la detalla en diferents apartats (fitxers disponibles, puntuacions dels usuaris, imatges, fòrums, articles, vídeos, etc) amb un sistema d'usuaris per puntuar, pujar partides a la xarxa, compra venda de jocs, notícies, etc.
- **Daus de jugador**: cadascun dels tres daus blancs que, inicialment, es llencen a cada torn juntament amb els de drac fins que es derrota o es perden ja que per cada flamarada rebuda (i no aturada amb un escut) es perd un dau de jugador.
- **Daus de drac**: cadascun dels tres daus que defineixen un drac i que contenen segons el tipus de drac i la cara que aparegui parts del seu cos (objectiu dels jugadors), muntanyes (valor nul) o flames.
- **Filler**: s'anomena així a un tipus de jocs de taula que agrupa a tots aquells que compleixen un seguit de condicions:
 - mecànica senzilla i que es pot explicar en cinc o deu minuts
 - durada de la partida inferior a una hora
 - normalment no ocupen massa espai i es distribueixen en caixa petita
 - a diferència de jocs amb tauler i un nombre elevat de tokens, miniatures i daus la preparació per iniciar la partida es pot realitzar en breus moments.
- **Fitxa de desafiament**: fitxa que disposa inicialment cada jugador de Dragon Slayer i que pot utilitzar, un cop per partida, per forçar al rival a seguir lluitant quan ha decidit plantar-se. De totes maneres, pot plantar-se igual però amb la penalització de rebre només la meitat dels punts aquell torn i que l'adversari que li ha plantejat el desafiament en guanyi 5.
- **KS** (Kickstarter): plataforma de micromecenatge en la que els usuaris poden patrocinar futurs projectes personals o de petites empreses i que, d'altre manera, difícilment veurien la llum. Un cop finalitza el període de

captació de mecenes, que acostuma a ser de vora un mes, el creador del kickstarter fabrica i distribueix el producte o servei entre tots els patrocinadors. Actualment i, com a curiositat, la plataforma mou molts més diners mitjançant el mecenatge de jocs de taula que no pas en altres àmbits del món de l'oci que també contempla, com poden ser la música, el cinema o els videojocs.

- **"Push your luck"** (temptar la sort): mecànica utilitzada en certs jocs de taula que consisteix en realitzar accions que depenen de l'atzar fins que el jugador es planta o, degut a l'atzar, per part o tots els beneficis que portava obtinguts fins aquell moment, ja sigui el torn, puntuació, etc. L'implementen jocs clàssics com el *Set i mig* o el *Black Jack* de cartes o de moderns com el *Dragon Slayer* o els més populars *Zombie Dice* o *Dungeon Roll*.

Aquesta mecànica també es coneix com "press your luck".

- **Token:** fitxa de cartró que representa alguna propietat de les mecàniques d'un joc de taula. En el cas del *Dragon Slayer* tenim els tokens (o fitxes) de desafiament, però en altres joc poden representar la vida d'un personatge, indicar el jugador inicial, punts d'acció, etc.

10 - Bibliografia

Mecàniques i característiques del joc:

- Fitxa del Dragon Slayer a la BGG:
<https://boardgamegeek.com/boardgame/158243/dragon-slayer>
(març, abril i maig de 2018)
- Mecenatge de Dragon Slayer a Kickstarter:
<https://www.kickstarter.com/projects/2012515236/dragon-slayer-0>
(març de 2018)

Programació i suport::

- **Daniel Campos y José Luís Redrejo** (2015) «*Gambas. Programación visual con software libre*»
(abril i maig 2018)
- Comunitat Gambas.es:
<https://www.gambas-es.org/>
(abril i maig 2018)
- Gimp: Manuals i ajudes del programa de manipulació d'imatges
<https://docs.gimp.org/2.8/es/>
(del 16 de març al 25 d'abril de 2018)
- Manual Audacity online:
<https://manual.audacityteam.org/>
(del 12 al 14 d'abril de 2018)

Disseny de la GUI:

- **Scott Rogers** (2014) «*Level Up! The Guide to Great Video Game Design*», John Wiley & Sons Inc.
(del 15 al 23 d'abril de 2018)

- **Gómez, Leopoldo Sebastián M.** (2000) "*Diseño de Interfaces de Usuario Principios, Prototipos y Heurísticas para Evaluación.*" Departamento del Laboratorio Pericial Informático Judicial del Neuquen (15 d'abril de 2018)
- **Galitz, Wilbert O.** (2007) "*The essential guide to user interface design: an introduction to GUI design principles and techniques.*" John Wiley & Sons. (15 i 16 d'abril de 2018)
- **Joan Arnedo** (2015) "*Un algoritmo de movimiento básico en videojuegos*" <http://informatica.blogs.uoc.edu/2015/03/02/un-algoritmo-de-movimiento-basico-en-videojuegos/> (23 d'abril de 2018)

Disseny de la intel·ligència artificial:

- **Margaret A. Boden** (2016) "*Inteligencia artificial*" , Turner publicaciones SL (abril 2018)
- **Fernando Sancho Caparrini** "*Minimax: Juegos con adversario*" <http://www.cs.us.es/~fsancho/?e=10> (del 1 al 5 de maig)
- Minimax (wikipedia) <https://es.wikipedia.org/wiki/Minimax> (del 9 al 15 de maig)
- Poda (wikipedia) https://es.wikipedia.org/wiki/Poda_alfa-beta (del 9 al 15 de maig)

Disseny de les diferents personalitats:

- **J. A. Vallejo-Nágera** (2006) "*Guía práctica de psicología*" Ediciones Temas de Hoy S.A. (19 i 20 d'abril de 2018)
- **Aitor Barbosa** (2016) "*Tipos de jugadores en Gamificación*"

<http://aitorbarbosa91.wixsite.com/kinderteach/single-post/2016/1/11/Tipos-de-jugadores-en-gamificaci%C3%B3n>

(del 7 al 9 de maig)

- **Ferran Altarriba Bertran** (2013) "*Tipos de jugadores en Gamification: teoría Barle*" <https://www.iebschool.com/blog/tipos-jugadores-gamification-2-innovacion/>

(del 7 al 10 de maig)

- **Sara Campos** (2018) "*Tipos de jugador. La clasificación de Marcewski*"

<https://edintech.blog/2018/03/20/claves-para-la-gamificacion-los-tipos-de-jugador-la-clasificacion-de-marcewski/>

(del 7 al 13 de maig)

- **Yu-kai Chou** (2015) "*Octalysis, the complete gamification framework*"

<http://yukaichou.com/gamification-examples/octalysis-complete-gamification-framework/>

(del 7 al 13 de maig)