

Estudi de l'Elo en els videojocs

GEP

Entrega 1: Definició de l'abast i contextualització
Especialització en Enginyeria del Software



Autor: Arnau Gesa Pascual

Director: Manuel Rello Saltor

Ponent: Ernest Teniente López

Tutor de GEP: Fernando Barrabés Naval

1 de març de 2021

Índex

1	Introducció	2
1.1	Introducció	2
1.2	Termes i conceptes	2
1.3	Identificació del problema	2
1.4	<i>Stakeholders</i>	4
2	Justificació	5
2.1	Solucions existents	5
2.2	Justificació de la solució	6
3	Abast	7
3.1	Objectius	7
3.2	Requisits	7
3.2.1	Requisits funcionals	7
3.2.2	Requisits no funcionals	8
3.3	Obstacles i riscos	8
4	Metodologia	10
	Referències	12

1 Introducció

1.1 Introducció

El meu projecte anomenat *Estudi de l'Elo en el món dels videojocs* és un Treball de Fi de Grau (TFG) realitzat en la Facultat d'Informàtica de Barcelona (FIB) de la Universitat Politècnica de Catalunya (UPC). En el meu cas, és un treball realitzat per al Grau en Enginyeria Informàtica, concretament, per a l'especialitat de Enginyeria del Software.

1.2 Termes i conceptes

En aquest apartat s'explicaran termes i conceptes que apareixeran durant tot el transcurs del projecte i que poden donar lloc a la confusió o desconeixement del significat al lector.

- **Elo:** Es coneix com a Elo el sistema de puntuació creat pel físic estatunidenc d'origen hongarès Arpad Elo, que mesura els nivells d'habilitat relatius dels jugadors en els esports com els escacs o videojocs competitius com el League of Legends. [1]
- **Matchmaking:** S'anomena matchmaking a l'algorisme utilitzat en la majoria dels videojocs multijugador que tracta d'emparellar els jugadors o equips d'un nivell o habilitats similars per enfrontar-se entre ells, mitjançant l'anàlisi de les estadístiques d'aquests. [2].
- **Videojoc competitiu:** S'entén com a videojoc competitiu un videojoc online, és dir, que es juga a través d'Internet, on els jugadors competeixen entre ells, sigui en equip o individual, per ser uns millors que els altres. Un exemple d'aquests poden ser els escacs o els jocs de lluita, com el Street Fighters o Mortal Kombat.
- **Habilitat del jugador:** Es coneix com a habilitat del jugador a la capacitat que té aquest per demostrar la seva destresa en un joc, és a dir, en altres paraules, com d'hàbil és jugant.

1.3 Identificació del problema

Avui en dia, en qualsevol moment i en qualsevol lloc es pot jugar a un videojoc, ja sigui en el carrer des d'un dispositiu mòbil, com a casa amb una consola o un ordinador. No és estrany dedicar-li temps en els moments lliures, ja sigui per jugar amb els amics, per divertir-se o per competir contra altre gent; o en els temps morts del dia, com per exemple quan es fa cua en un supermercat.

Tot això queda reflectit en la indústria dels videojocs, on cada cop més es generen més diners i s'espera que la quantitat augmenti. Això es pot veure en la gràfica de la figura 1, on s'observa que al 2019 es van generar 150,2 mil milions de dòlars i s'espera que en 2020, degut al Covid-19, es generin 179.7 mil milions de dòlars [3]. Com es pot veure, es tracta de quantitats de diners molt elevades, per tant, es pot arribar a la conclusió que desenvolupar un bon joc és molt important i pot generar molts beneficis.

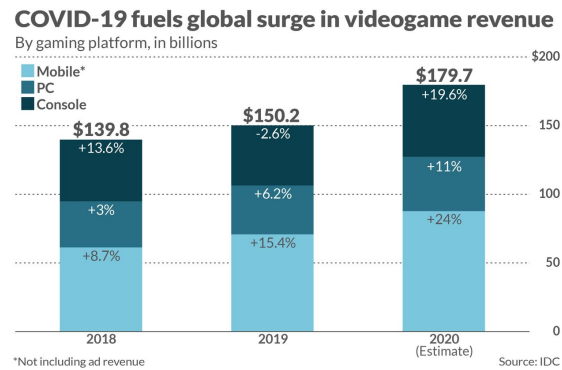


Figura 1: Ingressos dels videojocs. Font: [3]

A l'hora de dissenyar un videojoc s'han de tindre en compte molts aspectes, com la temàtica, els escenaris, els personatges, etc. Però també és molt important la jugabilitat, fer que sigui divertit i entretingut. A més, dintre dels jocs competitius, es necessita un bon sistema d'Elo i d'aparellament, per fer encara el videojoc més competitiu i just, tot i que també pot haver-hi interessos comercials.

Desenvolupar una bona fórmula de l'Elo equival, normalment, a representar el nivell d'habilitat del jugador i al seu progrés a l'hora de millorar. Per tant, s'ha d'estudiar rigorosament com fer-la i saber quines variables la modificaran (guanyar, perdre, morir més o menys, etc).

També cal afegir que l'Elo pot ser molt diferent depenent de la situació en que es vulgui aplicar. Per exemple, si en un videojoc es fa un campionat setmanal, on es vol reduir el número de jugadors durant el transcurs d'aquest, es pot generar un algorisme que contempli un Elo molt variat entre els diferents participants al principi de la competició, però que sigui més similar cap al final. Per una altre banda, si es juga un campionat mensual on es vol que hi participin tots els jugadors i es desitja una classificació apropiada al nivell d'habilitat de cada un, es pot desenvolupar un algorisme que agrupi les partides mitjançant un Elo similar.

Com es pot observar, la fórmula de l'Elo pot ser molt variable i s'han de fer diverses proves per arribar a l'adequada. El problema és que no hi ha cap programa que permeti simular les diferents versions per saber quina és vàlida i quina no. El que s'ha de fer actualment és plantejar una simulació que s'adeqüi a l'algorisme i fer moltes partides per veure l'evolució. Això pot tenir conseqüències, i és que testear i comparar els resultats de les diferents fórmules pot portar molt temps, el qual es podria invertir en altres aspectes del desenvolupament del videojoc.

És per aquest motiu que és necessària una eina software per a les empreses desenvolupadores de jocs amb interès comercial, que els permeti simular partides amb l'algorisme d'Elo que se li introdueixi, sigui quin sigui, i permeti avaluar els resultats. D'aquesta manera, podran escollir quin s'adapta millor a la jugabilitat i progressió del joc.

1.4 Stakeholders

Els *Stakeholders* són una part fonamental pel desenvolupament del projecte, ja que són totes aquelles persones o organitzacions afectades pel producte a desenvolupar i que tenen interès en aquest, ja sigui de forma directa o indirecta. A continuació es llistaran les parts interessades del software a desenvolupar.

- **Empreses de videojocs:** Les empreses desenvolupadores de videojocs també es veuran afectades pel projecte. Si el sistema d'Elo aplicat al producte és bo, hi ha més possibilitats de que el joc triomfi. Per tant, quan més èxit tingui el joc, més conegut serà, més jugadors captarà i més beneficis obtindrà l'empresa.
- **Jugadors:** Els jugadors de videojocs són les principals parts interessades i són les més importants. Aquests es veuran implicats de forma indirecte, ja que són els que jugaran i seran puntuats pel sistema d'Elo resultant. També són qui han de trobar el joc entretingut i divertit, ja que seran els encarregats de fer que aquest tingui èxit.
- **Desenvolupadors de videojocs:** Els desenvolupadors de videojocs són qui utilitzaran el software de forma directa. Seran els encarregats d'estudiar i plantejar les diferents fórmules del sistema d'Elo i simular-les en l'eina a desenvolupar, per així decidir quina d'elles s'adapta més al videojoc i farà que hi hagi una millor experiència.
- **Propietari del TFG:** Jo mateix soc un dels principals *stakeholders* del projecte, ja que seré l'únic desenvolupador. El meu objectiu serà fer-lo de la millor forma possible i eficient, testejar-lo i documentar tot el procés.

2 Justificació

2.1 Solucions existents

El sistema d'Elo d'un videojoc és molt important, ja que ha de representar, normalment, el nivell d'habilitat del jugador i a més, aconseguir fer les partides entretingudes. Llavors, podem arribar a la conclusió de que ha d'estar ben fet per a que un joc triomfi. És per aquest motiu que he realitzat una recerca per Internet de programes que simulin i permetin configurar el sistema d'Elo d'un videojoc, però només he trobat repositoris de Github que ho simulen, donada una fórmula predeterminada, on a vegades es poden canviar algunes variables, o bé una pàgina web que fa el mateix. Els millors resultats obtinguts han estat els següents:

- **Repositori de RemiFabre [4]:** En aquest repositori anomenat "Elo" es troba un programa fet en python que simula l'evolució de l'Elo en diversos jugadors amb puntuacions diferents. Aquest programa té dos modalitats, en la primera és que els jugadors només s'enfronten entre sí una vegada rere l'altre, mentre que en la segona modalitat s'enfronten a jugadors creats només per aquella partida i amb un "winrate" fixat. L'algorisme de l'Elo està predeterminat i com a molt es pot canviar el valor de certes variables. Posteriorment, es mostra el resultat en una gràfica.
- **Repositori de cardsorg [5]:** Aquest repositori de Github anomenat "Elo" inclou un software que simula l'enfrontament entre dos jugadors i diu el respectiu Elo de cadascú. Té tres funcionalitats diferents; la primera permet fer l'enfrontament entre dos persones amb l'Elo i el resultat que es vulguin assignar. També permet canviar el valor de la variable K, que és el màxim de punts que pot guanyar o perdre un jugador en una partida. La segona funcionalitat consisteix en mostrar la diferència entre els dos jugadors i la tercera fa el mateix, però també mostra la diferència estimada.
- **Repositori d'iaia [6]:** En aquest tercer repositori anomenat "Elo" es troba un altre programa que permet simular i modificar alguna variable del sistema d'Elo. En aquest cas, es poden crear jugadors amb la puntuació desitjada i enfrontar-los entre sí, decidint el resultat. També diu si són professionals o principiants depenent dels seus punts. Aquest software també permet varies configuracions, com la modificació de la variable K, on es pot definir depenent del número de partides fetes. També permet modificar a partir de quina quantitat de punts es deixa de ser un principiant i amb quina puntuació comença un jugador.
- **Elo Calculator [7]:** Aquesta calculadora online de la pàgina web "Omnicalculator" permet calcular l'Elo d'un jugador després de fer diferents jocs i també modificar la variable K. Hi ha dos opcions a escollir: una única partida o múltiples d'elles. La calculadora deixa assignar els punts desitjats al jugador principal i a cada un dels seus rivals, permetent també escollir el resultat de cada enfrontament. Al final, mostra una taula amb quants punts s'han guanyat i perdut en cada joc i quina és la puntuació final.

2.2 Justificació de la solució

Un cop vistes totes les solucions que hi han actualment a Internet, podem concloure que totes elles tenen aspectes positius: permeten modificar la variable K , mostren un gràfic de com va evolucionant l'Elo després de diverses partides, permeten escollir el resultat de cada joc o ensenyen quants punts es guanyen o perden en cada un. Tot i així, totes elles estan lligades a la mateixa fórmula, i és aquesta la principal diferència amb el meu projecte.

La meva solució també permet modificar el valor de diverses variables, simular diverses partides i exportar els resultats per fer una gràfica, però el més important és que permetrà canviar l'algorisme per calcular l'Elo completament. Es podrà utilitzar l'algorisme que es desitgi, ja sigui el predeterminat o un creat per l'empresa o persona que dissenyi el videojoc, permetent així veure la comparativa entre les diferents fórmules i escollint la més adequada pel videojoc.

3 Abast

En aquest apartat es comentaran els objectius que ha de tindre el projecte, igual que els requisits que ha de complir. També es contemplaran els obstacles i riscos que poden aparèixer durant tot el desenvolupament.

3.1 Objectius

L'objectiu principal d'aquest projecte és poder crear un programa software que permeti simular el càlcul de l'Elo en els videojocs i veure la seva evolució, per determinar si és un algorisme vàlid, per veure si l'Elo correspon al nivell del jugador i per veure si l'aparellament és el correcte. Però per aconseguir que el programa pugui complir amb aquestes funcionalitats, cal assolir un seguit de sub-objectius:

- **Personalització de l'algorisme del sistema d'Elo:** El programa ha de permetre tant la modificació de variables com poder canviar la fórmula sencera, permetent així introduir qualsevol algorisme per veure el seu funcionament.
- **Simulació de partides:** El software ha de permetre simular partides per així poder aplicar la fórmula de l'Elo i poder comprovar que l'aparellament de jugadors és el correcte. S'ha de contemplar que poden haver-hi variables que poden afectar al resultat de la partida, per exemple un jugador pot tenir un arma millor que l'altre, una armadura més resistent, un millor cotxe, etc. També ha de tenir en compte la diferència d'habilitat entre els contrincants, el resultat esperat vs. el real i la quantitat de punts d'Elo guanyats o perduts per cada partida.
- **Comparar resultats entre diferents variables i algorismes:** Per poder determinar si la fórmula de l'Elo és la correcta, cal poder comparar els resultats entre els diferents algorismes. És per aquest motiu que es guardaran les dades i es permetrà exportar-les a l'Excel, per poder analitzar-les i comparar-les.

3.2 Requisits

Per tal d'assolir els objectius del projecte i fer que tingui un bon funcionament, cal satisfer un seguit de requisits funcionals i no funcionals que es mostraran a continuació.

3.2.1 Requisits funcionals

Pel que fa als requisits funcionals, ens centrarem en l'entrada de fitxers Lua, en tot el procés aparellament i simulacions de partides, en l'emmagatzematge de dades i en la sortida de fitxers en format Excel.

- **Entrada de fitxers Lua:** Lua és un llenguatge de programació de script potent, eficient i lleuger que pot ser integrat dins d'altres softwares o pot ser compilat en qualsevol d'ells sempre que estigui basat en aquest llenguatge C, ja que és com està fet l'interpret [8]. El programa ha de permetre poder carregar un fitxer de script en Lua que contingui l'algorisme del sistema d'Elo, per així poder-lo executar.
- **Matchmaking:** L'eina a desenvolupar ha de ser capaç d'aparellar els diversos jugadors segons la diferència d'Elo demanada, per poder simular posteriorment les partides.
- **Simulació d'una partida:** El sistema ha de simular una partida entre dos jugadors o equips i determinar un resultat, tenint en compte totes les variables necessàries.
- **Càlcul de l'Elo:** El software ha de permetre poder executar l'algorisme de l'Elo i modificar la puntuació de cada un dels jugadors de la forma adequada.
- **Configuració de la simulació:** El sistema ha de permetre la configuració d'una sèrie de variables per fer la simulació, com el nombre de partides a simular o la diferència de puntuació entre els jugadors d'una.
- **Emmagatzematge de dades:** El software ha de poder emmagatzemar les dades extretes de les simulacions per poder-les utilitzar i actualitzar més tard.
- **Sortida de fitxers Excel:** L'eina a desenvolupar ha de poder permetre exportar les dades obtingudes a un fitxer Excel per així poder-les analitzar i comparar.

3.2.2 Requisits no funcionals

Pel que fa als requisits no funcionals, ens centrarem en el rendiment de l'aplicació i la usabilitat per l'usuari.

- **Facilitat d'ús:** El programa ha de ser intuïtiu i simple per a que qualsevol persona el pugui utilitzar.
- **Aprenentatge:** El software ha de ser fàcil d'aprendre per a qualsevol usuari.
- **Eficiència:** El programa ha de ser eficient degut a la gran quantitat de simulacions que ha de fer, per tant, ha d'anar el més ràpid possible.
- **Aparença:** L'eina ha de ser atractiva visualment pels usuaris que l'utilitzin.

3.3 Obstacles i riscos

Durant tot el transcurs del projecte poden aparèixer diversos obstacles i riscos que poden alentir el desenvolupament del projecte. Per aquest motiu, cal identificar-los i saber-ne d'ells.

- **Data d'entrega fixa:** El TFG té una data límit fixada i, per tant, que s'ha de complir. Però els possibles problemes que ens podem trobar durant el desenvolupament del projecte poden generar endarreriments i, per tant, empitjorar la qualitat del programa o eliminar alguna funció no essencial per falta de temps.

- **Desconeixement en simulació:** Un aspecte fonamental del programa és la simulació de partides. Però degut al meu desconeixement en l'àmbit de la simulació dins la informàtica, es probable que es generin endarreriments i possibles errors a l'hora de programar.
- **Desconeixement en les tecnologies emprades:** Degut al meu desconeixement sobre el llenguatge Lua o sobre el poc ús fet de QT, és probable que s'alenteixi el desenvolupament del projecte ja que hauré d'aprendre a utilitzar-los correctament.
- **Bugs i errors:** Durant tot el transcurs de programar i testejar el programa, poden aparèixer diferents errors i bugs que em poden fer perdre molt temps, i per tant, endarrerir el desenvolupament.

4 Metodologia

La metodologia que es durà a terme per desenvolupar el projecte és un aspecte molt important a tenir en compte, ja que pot canvia per complet com evolucionarà el projecte. Hi han dos grans metodologies: la cascada o *waterfall* i l'*agile*.

La primera, l'anomenada "*waterfall*" és la més tradicional. Aquesta té en compte els *stakeholders* i clients al principi del projecte, i després es desenvolupa un pla seqüencial per satisfer els seus requisits. Aquesta metodologia té aquest nom perquè cada fase del pla proposat passa a la següent un cop s'ha acabat l'anterior [9].

Aquesta metodologia és utilitzada quan els requeriments estan molt clars i fixes, quan la definició del producte és estable, quan la tecnologia no evoluciona constantment i quan el projecte és curt [10].

Per una altre part, la metodologia *agile* consisteix en una continuació iteració de desenvolupament i testeig del software on s'interacciona i constantment amb els clients i *stakeholders* per saber el seu *feedback* [11].

Aquesta és utilitzada per projectes llargs on es vol el *feedback* del client constantment, quan es vol treballar amb tecnologies que evolucionen constantment, també quan els requeriments no estan clars i poden canviar freqüentment, igual que la definició del producte. Com es pot observar, és la contrapart de la metodologia *waterfall*.

Per tant, un cop explicades les dos metodologies, es pot observar que el meu projecte s'assimila a les condicions de la primera metodologia. El període de temps és curt, ja que el TFG dura entre tres i quatre mesos, i els requeriments ja estan definits, igual que el producte.

Llavors, un cop decidida la metodologia, s'ha d'explicar com funciona. Com bé es pot observar en la figura 2, aquesta metodologia consta de sis fases. La primera d'elles s'anomena "Requeriments", i és on s'especifiquen i redacten tots els requeriments. La segona s'anomena "Disseny", i és on s'especifica el disseny del sistema i la seva arquitectura. La següent és la fase de "Desenvolupament", on el programa és dividit en petites unitats on cada una farà una funcionalitat i la testearà. Posteriorment ve la fase de "Testeig", on s'ajunta tot el codi fet per cada una de les unitats anteriors i es testea el resultat. Seguidament es troben els errors i es solucionen. La següent fase és la de "Desplegament", on el producte es llença al mercat o és entregat al client. Finalment hi ha la fase de "Manteniment", on es manté i millora el programa [12].



Figura 2: Fases de la metodologia *Waterfall*.
Font: [12]

Per una altra part, per poder desenvolupar el projecte, caldrà utilitzar un seguit d'eines per poder validar que s'assoleixen els objectius.

- *Trello* [13]: Trello és una eina per gestionar els projectes i que serà utilitzada per apuntar les diverses tasques que s'han de realitzar i així saber quines s'estan fent, quines s'han de fer i quines s'han realitzat del tot.
- *Github* [14]: Github és un servei de repositoris que serà emprat per guardar i compartir el codi i la documentació feta durant tot el projecte. Té l'avantatge de que es poden accedir a versions anteriors del codi i també gestionar els errors i bugs trobats.

Finalment, per tal de mantenir al director informat i poder comentar qualsevol dubte o error, s'ha acordat fer, sempre que sigui possible i necessari, una reunió per setmana. D'aquesta manera es podrà discutir el progrés del projecte i observar si es va pel bon camí.

Referències

- [1] *Elo rating system*. Wikipedia. URL: https://en.wikipedia.org/wiki/Elo_rating_system (cons. 26-02-2021).
- [2] *Definición de Matchmaking*. GamerDic, Diccionario online de términos sobre videojuegos y cultura gamer. URL: <http://www.gamerdic.es/termino/matchmaking> (cons. 27-02-2021).
- [3] *Videogames are a bigger industry than movies and North American sports combined, thanks to the pandemic*. MarketWatch. URL: <https://www.marketwatch.com/story/videogames-are-a-bigger-industry-than-sports-and-movies-combined-thanks-to-the-pandemic-11608654990> (cons. 26-02-2021).
- [4] RemiFabre. *Elo*. Github. URL: <https://github.com/RemiFabre/Elo> (cons. 27-02-2021).
- [5] cardsorg. *Elo*. Github. URL: <https://github.com/cardsorg/Elo> (cons. 27-02-2021).
- [6] iain. *Elo*. Github. URL: <https://github.com/iain/Elo> (cons. 27-02-2021).
- [7] Wojciech Sas. *Elo Calculator*. Omnicalculator. URL: <https://www.omnicalculator.com/sports/Elo> (cons. 27-02-2021).
- [8] *Lua - About*. Lua. URL: <http://www.lua.org/about.html> (cons. 28-02-2021).
- [9] *The Ultimate Guide... Waterfall Model*. Project Manager. URL: <https://www.projectmanager.com/waterfall-methodology> (cons. 28-02-2021).
- [10] *SDLC - Waterfall Model*. Tutorials Point. URL: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm (cons. 28-02-2021).
- [11] *Agile Vs Waterfall: Know the Difference Between Methodologies*. Guru99. URL: <https://www.guru99.com/waterfall-vs-agile.html> (cons. 28-02-2021).
- [12] *Waterfall Model*. Castellan Systems. URL: <https://castellansystems.com/kb/Waterfall.cshtml> (cons. 28-02-2021).
- [13] *Trello*. Trello. URL: <https://trello.com> (cons. 28-02-2021).
- [14] *Github*. Github. URL: <https://github.com/> (cons. 28-02-2021).