



Landowners

Jordi López Belzunces

Màster Universitari en Disseny i Programació de Videojocs
Treball Final de Màster

Helio Tejedor Navarro

Joan Arnedo Moreno

07/06/2020



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

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Landowners</i>
Nom de l'autor:	<i>Jordi López Belzunces</i>
Nom del consultor/a:	<i>Helio Tejedor Navarro</i>
Nom del PRA:	<i>Joan Arnedo Moreno</i>
Data de lliurament (mm/aaaa):	<i>06/2020</i>
Titulació o programa:	<i>Màster Universitari en Disseny i Programació de Videojocs</i>
Àrea del Treball Final:	<i>Videojocs</i>
Idioma del treball:	<i>Català</i>
Paraules clau:	city-building, videogame, sandbox
Resum del Treball (màxim 250 paraules):	
<p>Crear un videojoc que sigui alhora simple i atractiu en un límit de temps curt pot semblar una tasca impossible, sobretot tenint en compte les tendències actuals. No obstant, l'aparició de motors de jocs genèrics de gran potència ha permès reduir els temps i també centrar-se més en allò que es vol crear i no tant en com es crearà. Si, a més, eliminem la interdependència entre els components al màxim, no intentem reinventar la roda i abstraïem una sèrie de conceptes podem obtenir el producte buscat sense gaires escarafalls. Si mirem una mica més enllà i ens fixem en els recursos més limitats d'altres móns (com el dels jocs de taula) trobarem formes simples de representar idees complexes sense atabalar al jugador.</p> <p>Aquest projecte intenta recrear un joc de construcció de ciutat clàssic com els de la sèrie City Building d'Impressions o SimCity. Però fent-ho en la seva forma més simple, un retorn a les arrels del gènere. Per emfatitzar aquesta simplicitat, construirem una interfície directa però intuïtiva. L'ingredient final serà afegir un petit gir a la mescla amb un component sandbox: farà les sessions prou flexibles com per a evitar mecàniques complexes i micro-gestió. Per a tot això farem servir Unity, un popular i ben conegut motor de jocs genèric.</p>	

Abstract (in English, 250 words or less):

Creating a videogame both simple and appealing within a short time limit may seem an impossible task, especially if one takes into account current trends. However, the emergence of powerful generic game engines has allowed time reduction and also focusing in what one wants to create and not so much in how it will be created. If, in addition, we dispose component interdependence as much as possible and abstract some of the components we can obtain the product wanted without trouble. Furthermore, we can rely on worlds on resources are more limited (like on boardgames) and will found simpler ways for complex ideas to be represented without overwhelming the player.

This project aims to recreate a classic city-building videogame like the ones in Impressions City Building series or SimCity. But doing it in its simplest form, a return to the roots of the genre. To emphasize this simplicity, we'll build a very straightforward but intuitive user interface. The final ingredient will be adding a little twist to the mix with a sandbox component: making its sessions flexible enough to avoid complex mechanics and micro-management. For all this we will use Unity, a popular well known generic game engine.

Índex

1. Introducció.....	1
1.1 Context i justificació del Treball	1
1.2 Objectius del Treball.....	1
1.3 Enfocament i mètode seguit	1
1.4 Planificació del Treball.....	2
1.5 Breu sumari de productes obtinguts	3
1.6 Breu descripció dels altres capítols de la memòria	3
2. Estat de l'art	5
2.1. Definint el gènere de <i>Landowners</i>	5
2.2. Tecnologia.....	6
3. Definició del joc	8
3.1. Breu descripció del joc	8
3.2. Interacció joc-jugador	8
3.3. Plataforma de destí	9
3.4. Ambientació.....	9
3.5. Definició dels elements.....	9
3.6. Interacció entre els actors del joc	10
3.7. Objectius plantejats al jugador	10
3.8. Concept art.....	11
4. Disseny tècnic	12
4.1. Entorn de desenvolupament.....	12
4.2. Requeriments tècnics de l'entorn de desenvolupament	14
4.3. Inventari d'eines emprades	15
4.4. Inventari d'assets i recursos emprats	15
4.5. Arquitectura del projecte.....	16
5. Disseny de nivells.....	19
5.1. HUD de recursos	19
5.2. Log de joc.....	20
5.3. Caselles.....	20
5.4. HUD de control.....	22
5.5. Altres	23
5.5.1. Pantalla d'ajuda.....	23
5.5.2. Pantalla d'esdeveniments	23
5.5.3. Pantalla de mercat	24
6. Manual d'usuari	25
6.1. Requeriments tècnics	25
6.2. Menú principal	25
6.3. Menú d'opcions	26
6.4. Configurant una partida	26
6.5. Jugant a <i>Landowners</i>	27
6.5.1. Recursos	28
6.5.2. Construcció	28
6.5.3. Millora.....	29
6.5.4. El mercat	30
6.5.5. Treballadors	31

6.5.6. Esdeveniments.....	32
7. Conclusions.....	35
8. Glossari	37
9. Bibliografia.....	38

Llista de figures

Figura 1: Vista cenital amb HUD a Lincity	6
Figura 2: Vista isomètrica amb HUD a Lincity-NG.....	7
Figura 3: Exemple de HUD a Civilization VI	11
Figura 4: Exemple de HUD temporal a Civilization VI	11
Figura 5: Concept art del mapa de Landowners.....	11
Figura 6: Diagrama d'arquitectura de Landowners.....	18
Figura 7: Mapa de joc.....	19
Figura 8: Detall del HUD de recursos.....	19
Figura 9: Molí generant felicitat	20
Figura 10: Detall de l'efecte +1 descartat	20
Figura 11: Detall del log	20
Figura 12: Detall de les caselles.....	21
Figura 13: Caselles actives amb radi 2	22
Figura 14: Detall del HUD de control.....	22
Figura 15: Botons de construcció	23
Figura 16: Pantalla d'ajuda.....	23
Figura 17: Esdeveniment amb dos accions possibles.....	24
Figura 18: Pantalla de mercat	24
Figura 19: Menú principal de Landowners	25
Figura 20: Menú d'opcions de Landowners.....	26
Figura 21: Configurant la partida	27
Figura 22: Cost del molí	29
Figura 23: Edifici en construcció.....	29
Figura 24: Botons de construcció	29
Figura 25: Cost de millora del Castell.....	30
Figura 26: Comprant recursos.....	31
Figura 27: Venent recursos	31
Figura 28: Bescanviant recursos	31
Figura 29: Una mina acaba de produir un recurs roca	32
Figura 30: Un molí que ha completat 6/10 de la producció	32
Figura 31: Esdeveniment aleatori amb una acció possible	33
Figura 32: Pagament del delme	33
Figura 33: Esdeveniment de final de partida	34

Llista de taules

Taula 1: Planificació original per setmanes	2
Taula 2: Planificació final.....	3
Taula 3: Exemples d'esdeveniments aleatoris	8
Taula 4: Resum comparativa de motors.....	12
Taula 5: Inventari d'eines emprades.....	15
Taula 6: Inventari d'assets i recursos	16
Taula 7: Requeriments tècnics	25

1. Introducció

1.1 Context i justificació del Treball

Cal dissenyar un videojoc senzill que es pugui implementar en el temps especificat. Per a fer-ho, és important tenir en compte que el principal enemic és aquest límit de temps. Això implica que s'ha de buscar un concepte simple, gens rebuscat, a poder ser reutilitzant conceptes i eines que se saben vàlids i provats.

Com que no es disposa d'un equip d'artistes i dissenyadors, també s'haurà de pensar una interfície igualment simple però, alhora, fàcil d'entendre i utilitzar.

A més a més, s'hauria d'assegurar una certa rejugabilitat. Això es pot aconseguir per exemple afegint un gran número de nivells però ens complicaria el calendari dissenyar-los. Una altra manera de fer-ho, que presentarem en aquest Treball, és ideant algun sistema que es pugui aleatoritzar (p.ex: un mapa) i que permeti una certa configurabilitat.

D'aquesta manera s'obtindrà un producte que, sense voler ser un triple A (ni tan sols un triple B), permetrà mantenir l'atenció de l'usuari més enllà de la novetat.

1.2 Objectius del Treball

- Dissenyar un videojoc factible en el temps especificat
- Implementar el disseny definit desenvolupant el videojoc complet
- Alta configurabilitat dels diferents paràmetres de la partida
- Aconseguir una rejugabilitat mitja-alta (és a dir, que no cansi a la primera)
- Interfície simple però intuïtiva

1.3 Enfocament i mètode seguit

En tot procés creatiu sempre hi ha dos camins a seguir:

- Crear
- Reutilitzar.

En el cas que ens ocupa, crear de zero implica recórrer el camí que altres ja han fet. Camí que està ple d'errors, incompatibilitats i obstacles que altres ja han solventat o esquivat en recórrer-lo. Desgraciadament, sortejar tots aquests impediments requereix un temps del que no disposem. També s'han de tenir en compte les pròpies limitacions (no és el mateix crear art sense haver-ho fet mai que sent dibuixant professional).

Per això, creiem fermament que l'enfocament a seguir és el de la reutilització (allà on es pugui). Per exemple: agafant un gènere ja existent, prenent idees de jocs amb conceptes similars o utilitzant llibreries de codi per certes funcionalitats.

En general, s'ha reutilitzat la feina d'altres creadors (art, so, ...), sempre amb llicències lliures, i conceptes genèrics com les mecàniques de joc.

La metodologia, doncs, es basarà en les següents premisses:

- No reinventar la roda
- Abstreure components de forma que es puguin reutilitzar
- Centralitzar les dades de forma que es puguin accedir des de qualsevol lloc (no necessitem seguretat màxima)
- Fer que tots els engranatges siguin independents (o gairebé) de la resta, de forma que en tocar un no afectem directament a la resta (no ens hauria de caler revisar el codi d'altres parts).

1.4 Planificació del Treball

La planificació original era tal com s'indica a la Taula 1.

PAC1 Disseny	Setmana 1	- Conceptualització i definició de la idea
	Setmana 2	
PAC2 Versió parcial	Setmana 1	- Creació de les diferents escenes - Creació del menú principal - Creació de la pantalla d'opcions - Creació de la escena de joc (buida) - Creació de la pantalla de fi
	Setmana 2	- Gestió del mapa de joc (bàsic) - Creació del HUD
	Setmana 3	- Interacció amb les caselles - Gestió del mapa de joc (definitiu)
	Setmana 4	- Sistema de gestió dels treballadors (bàsic)
	Setmana 5	- Sistema econòmic (bàsic)
PAC3 Versió jugable	Setmana 1	- Anàlisi de les possibles modificacions al disseny orig. (1h)
	Setmana 2	
	Setmana 3	- Sistema econòmic (definitiu)
	Setmana 4	
	Setmana 5	- Ampliació dels sistemes
	Setmana 6	- Proves de joc complert i refinament del gameplay
	Setmana 7	- Correcció de bugs trobats
PAC Final	Setmana 1	- Correcció de bugs trobats - Traduccions
	Setmana 2	- Addició de millores - Port per a mòbils/PS Vita (si queda temps)

Taula 1: Planificació original per setmanes

Es va plantejar que hi hauria 10-15 hores lliures per dedicar al projecte setmanalment. A la pràctica, això va resultar impossible i realment es va veure que es tenien unes 8-10 hores. Per aquesta raó es van haver de

fer una sèrie d'adaptacions i la planificació final va quedar tal com es veu a la Taula 2.

PAC1 Disseny	Setmana 1	- Conceptualització i definició de la idea
	Setmana 2	
PAC2 Versió parcial	Setmana 1	- Creació de les diferents escenes - Creació del menú principal - Creació de la pantalla d'opcions - Creació de la escena de joc (buida) - Creació de la pantalla de fi
	Setmana 2	- Gestió del mapa de joc (bàsic) - Creació del HUD
	Setmana 3	- Interacció amb les caselles - Gestió del mapa de joc (definitiu)
	Setmana 4	- Sistema de gestió dels treballadors (bàsic)
	Setmana 5	- Sistema econòmic (bàsic)
PAC3 Versió jugable	Setmana 1	- Anàlisi de les possibles modificacions al disseny orig. (1h)
	Setmana 2	- Sistema econòmic (bàsic)
	Setmana 3	- Sistema de gestió dels treballadors (definitiu)
	Setmana 4	
	Setmana 5	- Sistema econòmic (definitiu) Mercader
	Setmana 6	- Ampliació dels sistemes
	Setmana 7	- Proves de joc complert i refinament del gameplay
PAC Final	Setmana 1	- Correcció de bugs trobats - Traduccions
	Setmana 2	- Addició de millores - Port per a mòbils/PS Vita (si queda temps)

Taula 2: Planificació final

1.5 Breu resumari de productes obtinguts

- El videojoc *Landowners* per a PC amb SO Windows/Linux/OSX.

1.6 Breu descripció dels altres capítols de la memòria

- Capítol 2: Estat de l'art

En aquest capítol veurem una revisió sobre el gènere escollit i també la tecnologia utilitzada per desenvolupar *Landowners*.

- Capítol 3: Definició del joc

En aquest capítol conceptualitzarem el videojoc *Landowners*.

- Capítol 4: Disseny tècnic

En aquest capítol revisarem la part més tècnica del projecte, llistant les eines i recursos emprats i per què.

- Capítol 5: Disseny de nivells

En aquest capítol destacarem els diferents aspectes del mapa de joc i quines decisions han portat al seu aspecte i funcionament.

- Capítol 6: Manual d'usuari

Requeriments tècnics del maquinari per jugar i manual d'instruccions de joc.

- Capítol 7: Conclusions

Recopilarem tota la informació generada pel projecte i n'estudiarem el resultat en forma de conclusions.

2. Estat de l'art

Tractarem en aquesta secció tot allò relatiu al gènere i la tecnologia que envolta jocs com *Landowners*.

2.1. Definint el gènere de *Landowners*

Tot i el seu aspecte imitant un joc de taula, *Landowners* es podria definir com un joc de simulació de construcció de ciutats (*city-building*, en anglès). És cert que beu, com veurem més endavant, de conceptes i mecàniques típics de jocs de taula (caselles hexagonals, posicionament de treballadors, ...) però és cert també que només ho fa estèticament. Interiorment, es comporta com un *city-building* clàssic: absència de torns, gestió semi-automàtica, ...

No obstant, durant el procés de desenvolupament s'ha hagut de redirigir lleugerament la temàtica cap a la del simulador *sandbox*. Això no vol dir que el joc s'aparti del *city-building* clàssic totalment però implica un cert canvi de model de joc cap a les partides espontànies més experimentals (molt configurable, sense guardat, ...).

El *city-building* clàssic per excel·lència és *SimCity*, del que ja s'ha escrit i parlat de forma bastant extensa al llarg dels anys. Tot i que, als fonaments, hi ha alguna cosa de *SimCity* a *Landowners* no n'és el referent principal. Tampoc ho són els hereus del gènere *city-building* com *Cities XXL*, per això remarcuem el terme "clàssics". Els nostres referents són els jocs desenvolupats per *Impressions Games* durant els 90 (especialment *Zeus: Master of Olympus* però també *Pharaoh* i, en menor mesura, *Caesar*).

El gènere apareix a finals del 80 (3) amb el mencionat *SimCity* que ha arribat a tenir múltiples entregues en diferents plataformes (fins i tot hi havia un projecte per portar-lo a NES¹ (1)) amb gran èxit de vendes. De fet, es podria dir que és l'únic supervivent del gènere a dia d'avui per al gran públic junt amb la incombustible saga *Anno*, que barreja la construcció de ciutats amb tocs d'estratègia en temps real.

Durant els anys 90 va tenir una segona joventut amb la sèrie *City Building* (4) de la mencionada *Impressions Games* que, sota l'ala de *Sierra Entertainment*, també va aconseguir un gran èxit de vendes amb els seus jocs de gestió de ciutats a èpoques antigues. També és durant aquesta època (1998) que apareix *Anno 1602*, el primer joc de la saga.

Al segle XXI, com ha passat amb altres gèneres típics dels 80 i 90 (com les aventures gràfiques), el gènere perd pistonada tot i tenir cert èxit com a model free-to-play (*Townsmen*, *Settlers Online*) i en alguns títols experimentals com ara *Dwarf Fortress*. També es podria dir que es

¹ https://en.wikipedia.org/wiki/Nintendo_Entertainment_System

transforma en els actuals gèneres híbrids (com *Anno* o *Tropico*) o de gestió i microgestió al que pertany *Cities*.

Els desenvolupadors *Impressions* i *Maxis* (tot i haver creat l'exitós *Sims*) es veuen obligats a tancar portes i/o reinventar-se. Pel que fa a *Impressions* antics treballadors van formar *Tilted Mill Entertainment* i van recuperar el que va ser el seu últim títol, *Children of the Nile* així com continuar la saga *Caesar* i, fins i tot, van publicar un membre de la saga *SimCity!* (*SimCity Societies*) (2).

2.2. Tecnologia

Els jocs de construcció de ciutats no han canviat gaire amb el pas del temps. Segurament dissenyats i implementats amb motors *in-house*, com ha estat tendència fins fa relativament poc (sobretot amb l'aparició de motors genèric com *RenderWare*, *Unreal Engine* o *Unity*), la majoria de títols no deixen de ser una reinvençió o reambientació dels ja existents.

Generalment, s'ha fet servir la vista zenital o isomètrica i una sèrie de botons que permeten construir els diferents edificis i gestionar les infraestructures (camins, xarxes de transport i abastiment, ...). Podem veure exemples d'això a *Lincity*, un clon open-source de *SimCity*:



Figura 1: Vista cenital amb HUD a Lincity²

La figura 1 mostra la vista cenital, més utilitzada en els primers temps quan no es disposava de tanta potència de procés ni de GPUs

² Font original: web oficial (<http://lincity.sourceforge.net/screenshots/index.html>)

avançades. Podem trobar més exemples als primers *SimCity* i en versions dels jocs per a dispositius amb pocs recursos (p.ex: el mencionat port de *SimCity* per a NES).

El HUD pot presentar totes les opcions de cop o, si és més complex, diferents menús o pestanyes (com en el cas de la sèrie *City Building* d'Impressions) amb diferents opcions.



Figura 2: Vista isomètrica amb HUD a Lincity-NG³

La figura 2 mostra la versió NG de *Lincity*, que fa servir la vista isomètrica i un HUD més modern.

Pel que fa a la gestió dels diferents aspectes del joc, se sol tenir al sistema o diferents sistemes actuant en temps real independentment de les accions del jugador. És a dir, es "simulen" les diferents variables d'un entorn: demografia, clima, esdeveniments aleatoris (p.ex: desastres naturals), ... Els jocs de la saga *City Building* afegeixen també alguns aspectes interessants degut a la seva ambientació al món antic. Tenim divinitats i aliats/enemics que poden ajudar (o posar més obstacles si s'emprenen), la crecuda del Nil en el cas del joc *Pharaoh*, ...

En resum, el jugador controla la construcció i manteniment de la ciutat i els seus serveis i infraestructures però no controla l'univers de joc que està interactuant amb aquest. Això contrasta, per exemple, amb els jocs d'estratègia en temps real (*real-time strategy* o RTS, en anglès) on el mapa de joc sol ser inert i està al servei dels que participen de la partida: hom pot recollir recursos, construir o destruir a voluntat. Però la única amenaça real son els altres jugadors.

³ Font original: Wikipedia (<https://en.wikipedia.org/wiki/File:Lincity-ng.png>)

3. Definició del joc

Passem en aquest punt a especificar una mica més el concepte de *Landowners* i quina és la idea que dona peu a la seva creació.

3.1. Breu descripció del joc

A *Landowners* els jugadors interpreten el paper d'un terratinent que ha rebut unes terres d'un senyor feudal (Lord Bot) per a gestionar-les. Buides i desamparades, els jugadors hauran d'intentar aprofitar-les per sobreviure atraient nous pobladors i, qui sap, creant els fonaments del que serà una important ciutat en el futur.

El senyor exigirà la seva part dels rendiments del terreny i, amb els sobrants, s'haurà d'alimentar la població i comerciar. Tot això mantenint als vilatans relativament feliços per evitar una revolta que pot acabar arrasant els camps o, fins i tot, fent que el terratinent perdi el seu lloc de poder!

3.2. Interacció joc-jugador

El mapa de joc estarà format per caselles hexagonals que el jugador podrà seleccionar per obtenir informació sobre aquestes i també veure les accions disponibles. Mitjançant un HUD clàssic no-diegètic s'anirà informant al jugador de l'estat general de les coses. A més, s'afegiran elements espacials per notificar alguns aspectes com ara els recursos generats.

A banda hi haurà esdeveniments aleatoris que es notificaran al HUD amb possibles accions o decisions a prendre. Alguns d'aquests esdeveniments (p.ex: un incendi) poden durar un temps o requerir una acció immediata (p.ex: pagar un rescat) o alguna fita per desaparèixer (p.ex. aconseguir entregar una certa quantitat d'aliments).

Esdeveniment	Tipus	Acció/Fita
Incendi	Temporal	-
Senyor exigent	Fita	Pagar delme extra
Robatoris	Acció	Decidir càstig
S'organitza un festival	Acció/Temporal	Reunir una quantitat d'aliments per entregar al senyor (o a un aliat)

Taula 3: Exemples d'esdeveniments aleatoris

Es pot configurar la partida a partir d'uns quants paràmetres disponibles:

- Recursos inicials
- Configuracions dels esdeveniments

- Alguns valors de final de partida
- Comportament del marxant

3.3. Plataforma de destí

El desenvolupament se centrarà en la versió per a PCs amb SO Windows/Linux/OSX. Ni que no hauria de ser cap problema portar-ho a dispositius mòbils, el temps disponible és limitat. Es considerarà fer-ho si el temps ho permet.

Com a bonus personal, seria molt interessant fer una versió per a la consola PlayStation Vita (recordem que, durant un temps, va estar disponible a Unity) però s'ha descartat completament ja que Sony l'hi va retirar el suport fa temps.

3.4. Ambientació

El concepte de *Landowners* permet ambientació en qualsevol època adaptant alguns elements. De moment, però, ens centrarem en que està ambientat en època medieval i es deixa l'addició d'altres escenaris per a més endavant.

S'intenta representar el pas del feudalisme a un primer capitalisme com el que es va donar a les ciutats que van acabar formant la Lliga Hanseàtica.

Existeix un personatge anomenat "Lord Bot" que fa les vegades de senyor del jugador, aquell qui li ha entregat la custòdia de les seves terres i a qui hem d'accontentar.

3.5. Definició dels elements

Mapa de joc

El mapa de joc representa un territori del que, inicialment, només es controla una petita part. Aquesta part representa la donació inicial del senyor al jugador. Es poden desbloquejar noves parts del territori per mitjà de noves donacions.

Construccions

Cada casella pot contenir una construcció que es pot obtenir previ pagament dels recursos necessaris corresponents. Les construccions no estan disponibles immediatament, triguen un temps a ser finalitzades. La excepció és el castell, que estarà disponible des d'un inici fent el paper de centre de poder.

Un cop finalitzades, algunes requereixen de treballadors per funcionar. Els treballadors fan que la construcció pugui acomplir el seu propòsit. Per exemple: un camp no es podrà collir si no hi ha treballadors (en

aquest cas: pagesos). També pot passar que baixi el rendiment si la plantilla no és completa.

Les construccions es poden millorar un cop s'acompleixen una sèrie de condicions.

Les millores concedeixen bonus a la producció i rendiment, així com altres recompenses.

Treballadors

Els treballadors s'ocupen de fer les diferents feines disponibles. S'han d'alimentar regularment o es moren.

Cal notar que tots els habitants d'una casa no estan en edat de treballar (hi ha nens i avis al món).

Marxant

Cada cert temps arribarà un mercader ambulant disposat a comerciar, però només estarà disponible durant un cert temps.

3.6. Interacció entre els actors del joc

El programa gestionarà automàticament un llistat de treballadors disponibles en edat adulta i llocs de feina disponible. Els treballadors s'assignaran automàticament als llocs disponibles o quedaran a l'atur. Si es mantenen gaire temps aturats, podrien decidir marxar.

Les construccions actives defineixen quins i quants llocs de feina hi ha disponibles i produeixen recursos que després es poden utilitzar o vendre.

El marxant permet comprar i vendre diferents recursos ja sigui a canvi de diners o d'altres recursos. Això permet compensar la falta d'espai disponible que limita la varietat de recursos que es poden aconseguir.

3.7. Objectius plantejats al jugador

Bàsicament l'objectiu del jugador és aguantar el màxim de temps possible aconseguint engrandir la seva zona d'influència. El més important serà:

- No perdre la confiança del senyor (ni acabar-li la paciència!) per no ser expulsat
- Aconseguir mantenir la felicitat dels vilatans en un nivell prou alt com per a que no marxin
- Augmentar la producció per tenir recursos extra per vendre o emmagatzemar

3.8. Concept art

Es vol crear una experiència minimalista similar a la que presenten jocs com *Zeus: Master of Olympus* o *Civilization VI*. S'aprofitaran recursos gratuïts amb llicència d'ús lliure com els que publica el web www.kenney.nl

El HUD hauria d'ocupar només les vores de la pantalla, mostrant els recursos a la part superior, com a la captura següent⁴:

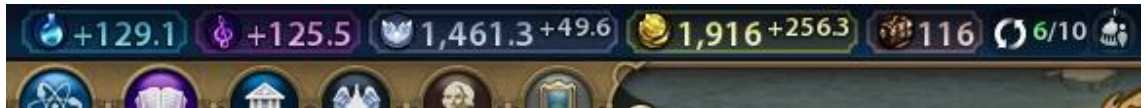


Figura 3: Exemple de HUD a Civilization VI

Per no ocupar pantalla, hi haurien elements temporals, per exemple al posar el cursor sobre una casella hauríem de poder consultar informació referent a aquesta. Mostrem un exemple orientatiu a la següent captura⁵:



Figura 4: Exemple de HUD temporal a Civilization VI

En la següent captura podem veure una part del mapa de joc amb una sèrie de construccions: un Habitatge (nivell 1) on es mostra que els habitants son feliços, un Molí (nivell 2) i un tercer edifici (nivell 1) inactiu, sense treballadors.



Figura 5: Concept art del mapa de Landowners

⁴ <http://www.allgames.com/news/83254-civilization-vi-feels-like-a-new-beginning-for-the-series>

⁵ <http://www.allgames.com/news/83254-civilization-vi-feels-like-a-new-beginning-for-the-series>

4. Disseny tècnic

4.1. Entorn de desenvolupament

A l'hora d'escollir un entorn de desenvolupament primer s'han de tenir en compte les possibilitats i les limitacions. En el nostre cas, per tal de fer un videojoc, una de les decisions més importants és escollir quin motor es farà servir. Un cop escollit el motor podem veure quines limitacions tenim (p.ex: SO on està disponible, requeriments tècnics, llicències, ...) i podem definir finalment quin camí se seguirà.

Pel que fa al motor s'han tingut en compte quatre opcions (Unity, Amethyst, raylib, motor propi). Cadascuna de les opcions (excepte la del motor propi, òbviament) correspon a una categoria amb diferents motors, de la qual s'ha escollit un candidat finalista. Les categories són:

- Motors genèrics professionals: **Unity**, Unreal Engine, Renderware, GameMaker, ...
- Motors genèrics fets per la comunitat: **Amethyst**, Piston, Godot, ...
- Motors minimalistes (professionals o no): moaiSDK, **raylib**, love2D, ...

Mostrem a la taula un resum dels principals arguments a favor i en contra de cada opció finalista i en farem un anàlisi més detallat a continuació.

Motor	A favor	En contra
Unity	<ul style="list-style-type: none">- Potència- Facilitat d'ús- Comunitat enorme- Recursos gairebé infinits	<ul style="list-style-type: none">- Complexitat innecessària- És una "caixa negra"- Millor per a 3D
Amethyst	<ul style="list-style-type: none">- <i>Data-driven</i>- ECS	<ul style="list-style-type: none">- Zero pràctica- Poca pràctica amb Rust- Comunitat petita- Recursos limitats
Raylib	<ul style="list-style-type: none">- Simplicitat- Altament modular- Disponibles bindings per a multitud de llenguatges- Fet a casa	<ul style="list-style-type: none">- Temps limitat- Comunitat petita- Recursos limitats
Motor propi	<ul style="list-style-type: none">- Es pot fer molt específic- Fita personal	<ul style="list-style-type: none">- Manca de temps- Poca experiència- Trobarem problemes que altres motors ja han resolt

Taula 4: Resum comparativa de motors

Revisem ara una mica més extensament cada un dels candidats:

Unity

<https://unity.com/>

En els darrers temps s'ha convertit en el fill predilecte de la comunitat. De gran potència (com demostren els tràilers que alliberen de tant en tant⁶) i facilitat d'ús: perfecte per als que acaben d'arribar i també per als experts.

La quantitat de recursos disponibles és pràcticament infinita i la comunitat és tan gran que és gairebé impossible no trobar resposta als problemes o qüestions que puguin aparèixer (excepte casos molt rars relacionats amb temes interns). El resultat és que es pot dedicar més temps a crear i millorar.

Ni que s'han mencionat un parell d'aspectes, a la pràctica no hi ha arguments en contra.

Com s'ha vist hi ha alternatives igual o més potents com el conegut Unreal Engine d'Epic Games, però la falta d'experiència els ha descartat.

Amethyst

<https://amethyst.rs/>

Menys conegut que Unity i desenvolupat en Rust. Rust és un llenguatge enfocat en la seguretat i la eficiència del codi, fins al punt que el compilador és capaç de detectar possibles *leaks*.

Amethyst abraça amb força i de forma rigurosa el concepte ECS (*Entity Component System*, també disponible a Unity) que tan de moda està darrerament en els cercles *gamedev*.

Té en la seva contra una comunitat petita i pocs recursos a comparació amb altres contendents com ara Unity, Unreal o Godot. A més a més, per a un projecte petit com el nostre, no ens cal un sistema ECS.

Raylib

<https://www.raylib.com/>

Petita llibreria escrita en C altament modular i increïblement simple, amb possibilitat de programar en diferents llenguatges (p.ex: C#, com Unity).

Ideal per a projectes petits (com el presentat en aquest document) però amb gran potencial. No és per a tothom, ja que no disposa d'eines

⁶ Amb la versió 2019.3 es va fer l'espectacular curtmétratge *The Heretic* disponible a Youtube (<https://www.youtube.com/watch?v=iQZobAhgayA>)

visuals com el cas d'Unity ni que té eines auxiliars (<https://www.raylibtech.com/>). Com diu l'autor a la web "*just coding in the most pure spartan-programmers way*".

El principal argument en contra és el temps limitat del projecte, ja que s'haurà de dedicar una part a construir els fonaments i es té molt poca experiència amb C/C++.

Motor propi

El meu objectiu des de petit havia estat entendre com algú podia fer que d'un tros de plàstic sortissin jocs com Sonic o Ecco The Dolphin. De gran jo també crearé videojocs, em deia. La realitat és més dura del que molts voldríem i, al final, et vas adaptant al que hi ha sense a vegades arribar a la meta (o arribant parcialment).

El cas és que m'agradaria fer un motor propi (i, de fet, n'hi ha hagut uns quants intents) però és més fàcil dir-ho que fer-ho. I com que sé de què parlo, lamentablement aquest cop tampoc serà possible ja que no hi ha temps material per dur la tasca a terme.

Sortosament serà una nova experiència que podré aplicar en el proper intent que, qui sap, potser serà el definitiu.

Motor escollit: Unity

L'elecció no ha estat fàcil ja que tots els candidats (excepte Unity) representaven un cert repte personal. Però el principal contrincant és el temps i, al final, la selecció s'ha reduït a dos candidats: Unity i raylib.

Finalment, i a contracor, el seleccionat serà Unity per que al cap i a la fi el que ens interessa aquí és fer la millor versió possible d'un videojoc. La facilitat per crear i reutilitzar *prefabs* i les eines visuals permeten reduir el cost en temps. Ja es tindrà temps de fer i aprendre en un futur els ets i uts del que conforma un motor de joc i la programació a més baix nivell.

Ara que ja tenim un guanyador, podem començar a muntar l'entorn de desenvolupament al seu voltant revisant les limitacions imposades⁷.

4.2. Requeriments tècnics de l'entorn de desenvolupament

L'entorn de desenvolupament estarà basat en el sistema operatiu Windows 10, actualitzat a la darrera versió.

La màquina utilitzada és un PC personalitzat amb les següents característiques principals:

⁷ Per exemple: l'Editor d'Unity està disponible per a Windows i OSX però, al no disposar de cap màquina Apple, forçadament haurem de fer servir Windows.

- CPU AMD Ryzen 5 2400G de 8 nuclis/16 fils a 3,0 Ghz
- GPU AMD Radeon RX Vega 11 integrada al processador
- Memòria RAM: 16GB DDR4 3200Mhz compartida amb la GPU integrada (2 GB)

També se fa servir com a ordinador de suport un PC portàtil model Dell Inspiron 15R 5537 amb les següents característiques principals:

- CPU Intel Core i5-4200U de 2 nuclis/4 fils a 1,6 Ghz
- GPU Intel HD 4400 integrada al processador
- GPU AMD Radeon HD 8600M amb 4 GB de VRAM
- Memòria RAM: 8GB DDR3 1600Mhz compartida amb la GPU integrada (2 GB)

4.3. Inventari d'eines emprades

Nom	Tipus	Versió
Unity	Motor + editor visual	2019.3.3f1
Visual Studio Code	Editor de codi	1.42.+
Sourcetree	Gestió de versions	3.0.+
GIMP	Editor de gràfics	2.10.18
Microsoft Word	Processador de textos	Office 365 ProPlus
Audacity	Editor de so	2.4.1
OBS Studio	Captura de vídeo	25.0.8
OpenShot	Editor de vídeo	2.5.1
Dia	Editor de diagrames	0.97.+

Taula 5: Inventari d'eines emprades

4.4. Inventari d'assets i recursos emprats

Nom	Autor	Llicència	URL
Hexagon Pack	Kenney	CC0 1.0	https://kenney.nl/assets/hexagon-pack
1bit Pack	Kenney	CC0 1.0	https://kenney.nl/assets/bit-pack
Emotes Pack	Kenney	CC0 1.0	https://kenney.nl/assets/emotes-pack
Medieval town	syncopika	CC-BY 3.0	https://opengameart.org/content/medieval-town
Medieval: King's Feast	RandomMind	CC0 1.0	https://opengameart.org/content/medieval-kings-feast
Sad Strings	Trinnox	CC-BY 3.0	https://opengameart.org/content/sad-strings
Kenney Fonts	Kenney	CC0 1.0	https://kenney.nl/assets/kenney-fonts
Wood texture	JCW	CC0 1.0	https://opengameart.org/content/wood-texture-tiles
Action sounds	pauliuw	CC0 1.0	https://opengameart.org/content/action-sounds

Female Merchant Voices	Holly Daniel	CC-BY 3.0	https://opengameart.org/content/female-merchant-voices
16 buttons clicks	Independent.nu	CC0 1.0	https://opengameart.org/content/16-button-clicks

Taula 6: Inventari d'assets i recursos

Tot i que en general s'han aprofitat tal qual, alguns d'aquests recursos s'han modificat lleugerament. En el cas de l'art per generar noves icones com la dels treballadors, que combina dos imatges en una.

En el cas dels efectes de so, retallant-los per quedar-nos la part que ens interessava.

A última hora, en el moment de provar les traduccions, s'ha detectat un petit problema amb les fonts. No disposen d'alguns símbols (com ara lletres accentuades) i l'efecte en ser substituïts pels símbols de la font per defecte és bastant lleig. No restava temps per corregir-ho i s'ha hagut de deixar tal qual.

4.5. Arquitectura del projecte

Pel que fa a l'arquitectura del projecte, se segueix el camí marcat anteriorment: evitar interdependència entre components i centralitzar dades. Es pot consultar un diagrama a la Figura 6.

Destaquem aquí una llista d'aspectes a tenir en compte sobre com funciona el projecte a la part menys visible (sense tenir en compte el gameplay):

- Es fa servir Resources per carregar dades que es necessiten en temps d'execució com ara les traduccions o diferents sprites.
- Hi ha una classe Enums que permet definir diferents estructures enum representant estats, tipus o possibilitats.
 - o Per definir l'estat del joc es fa servir una classe estàtica Game que fa de "màquina d'estats", a més de guardar informació sobre els diferents valors que afecten la partida (quantitat de recursos, nivell dels edificis, etc)
 - o Els diferents estats del joc es defineixen a Enums.cs
 - o Els diferents elements haurien de poder funcionar atenent només a l'estat marcat a Game. Això assegura la independència de tots els elements i simplifica o anul·la la necessitat d'interrelacionar-los. També redueix els possibles errors.
- S'ha creat un sistema de traducció molt simple que permet traduir on-the-fly tots els elements. Només cal marcar un element amb el tag "Translate" i fer que l'script corresponent heredi de TranslatableMonoBehaviour. Es traduiran tots els element TMP_Text al mètode Awake().

- El sistema de traducció carrega les diferents traduccions des d'un JSON a Resources. Això permet editar-les sense haver de tocar el codi.
- S'ha creat un sistema de definicions que permet carregar diferents aspectes del joc des d'un JSON a Resources.
 - Recursos disponibles a la partida
 - Tipus d'edificis
 - Esdeveniments
- S'ha creat un generador de mapes configurable
- Cada casella del mapa és independent, té el seu propi controller.
- S'ha crea un sistema de "logging" per mostrar missatges en pantalla
- S'ha creat un sistema de gestió d'esdeveniments (aleatoris i "fixes")

Les dades estan centralitzades a la classe estàtica Game, de forma que tots els components hi tenen accés. Si es volgués guardar la partida hauria de valer amb guardar l'estat d'aquesta classe. Potser el millor seria ampliar l'abast de GameStatus i fer servir aquesta classe com a contenidor de l'estat, però ambdues opcions serien bones. No és un aspecte que ens preocupi en aquest projecte ja que, en principi, no es guardarà la partida. Landowners no és un joc de partides de llarga durada, per lo que tampoc té gaire sentit. Si que tindria més sentit en la versió per a dispositius mòbils, on les sessions se poden interrompre inesperadament (p.ex: una trucada) però de moment queda fora.

Cada escena té un controlador que s'encarrega de gestionar què s'ha de fer segons l'estat del joc. S'intenta evitar la interdependència per que això porta a l'aparició d'errors que poden arribar a tenir una gran complexitat degut a la implicació de múltiples components. Si alguna dada o procés ha de ser accedit des de més d'un controlador, llavors s'ha de moure a Game o a alguna altra classe estàtica. Així podem tenir un controlador per a cada element disponible al món del joc (com el cas de les caselles) treballant independentment de la resta.

Això ens porta a l'aparició de DefinitionsManager i TranslationsManager. Tots dos gestionen dades que poden ser accedides des de qualsevol punt del programa: els textos de les traduccions i les definicions (edificis, recursos, ...). Els textos són necessaris en qualsevol pantalla del joc ja sigui pels botons, esdeveniments, missatges de log, etc. Per altra banda, les definicions s'han de carregar al iniciar el joc però també es poden fer servir, per exemple, per generar la barra de recursos segons els recursos definits.

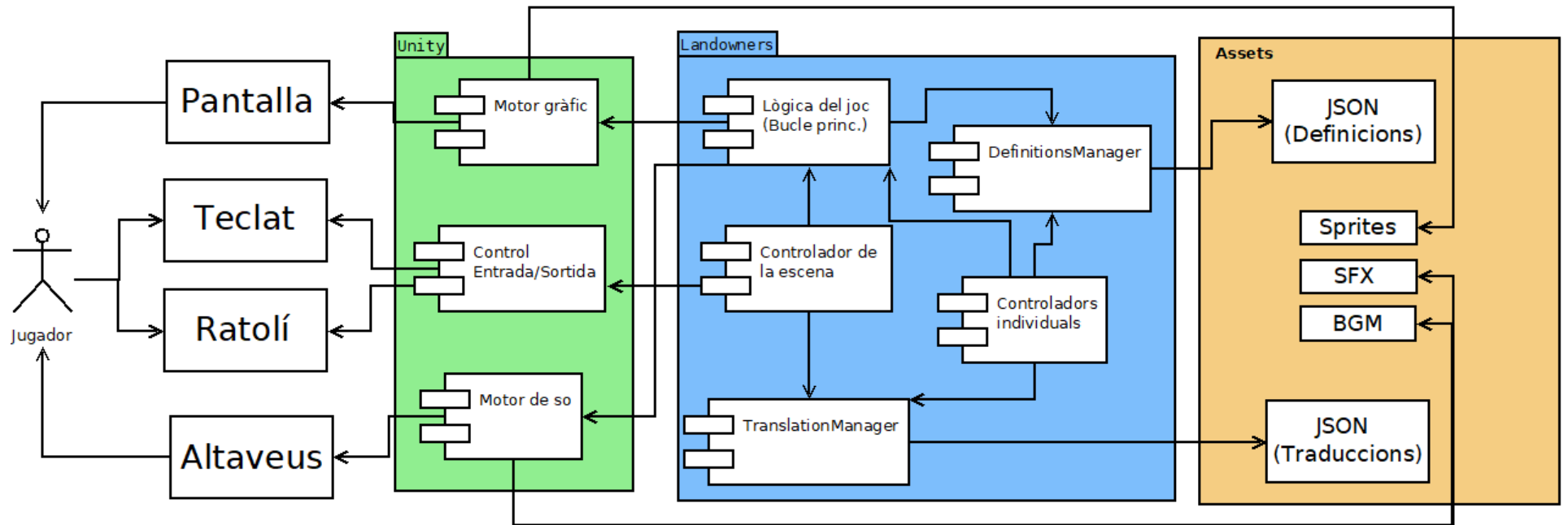


Figura 6: Diagrama d'arquitectura de Landowners





5. Disseny de nivells

En aquesta secció comentarem els diferents aspectes relatius al disseny de nivells.

A Landowners només existeix un nivell, el mapa de joc, que analitzarem de dalt a baix.



Figura 7: Mapa de joc

	HUD de recursos
	Log de joc
	Caselles actives
	HUD de control

5.1. HUD de recursos

El HUD de recursos es troba a la part superior de la pantalla i indica la quantitat de recursos actuals



Figura 8: Detall del HUD de recursos

Està pensat per ocupar el mínim espai possible aportant la màxima informació. En un primer moment es va voler afegir un efecte gràfic cada vegada que es generava un recurs amb una animació ascendent similar a la del Log (veure 5.2). L'efecte està creat (Figura 9) però al final no està implementat per que altres coses més importants li han passat endavant.

A més, hi ha una eina més potent per aconseguir el mateix efecte i son els elements espaials, com el de la Figura 9. El jugador rep la informació millor i més ràpidament ja que apareix al mapa, on està l'acció.

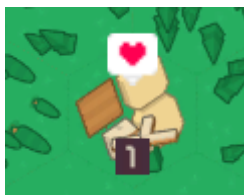


Figura 9: Molí generant felicitat

Al final, l'important amb el HUD de recursos era mostrar la informació rellevant i això és justament el que fa, ni que es podria ampliar i millorar (p.ex: afegint un efecte al passar el ratolí per damunt com a altres parts del joc).

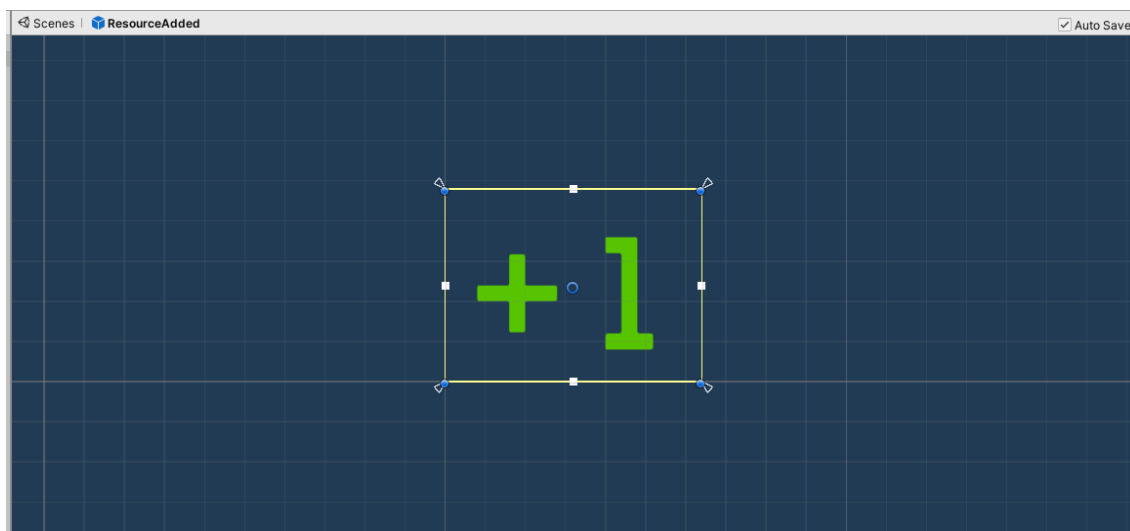


Figura 10: Detall de l'efecte +1 descartat

La barra es genera automàticament a partir dels recursos definits al JSON. Per a cada un dels recursos s'afegeix un prefab a la barra i aquest prefab recull el valor de la variable corresponent.

5.2. Log de joc

A la part superior central també es mostren alguns missatges informant sobre algunes coses que passen durant la partida. Es pot desactivar des del menú d'opcions.

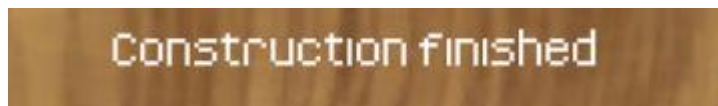


Figura 11: Detall del log

Es tracta de missatges dirigits al jugador o que li poden resultar d'interès, però no són vitals per a la partida.

5.3. Caselles

Les caselles hexagonals formen la part més important del nivell, ja que representen allò que el jugador està gestionant directament. Només es

mostra el terreny controlat pel jugador, que depèn de la felicitat de Lord Bot.

Les caselles actives, sobre les que es pot actuar, estan representades totalment opaques. Les caselles adjacents a aquesta zona es mostren representades semi-transparentes. Aquí s'intenta representar un "horitzó" virtual, a mesura que ens apropem al marge del món conegut veiem a la llunyania noves terres. Això dóna peu a implementar limitacions a les caselles (p.ex: si no hi ha roques no es pot construir una mina) de forma que aquesta visibilitat guanyaria un component estratègic (planificar la expansió futura).



Figura 12: Detall de les caselles

Es poden disposar rotades aleatòriament en homenatge a la disposició típica de jocs de taula com Carcassonne, on les diferents fitxes (quadrades en aquest cas) s'han d'orientar fent que el dibuix encaixi.

El funcionament del "tauler" és el següent:

- El castell representa la peça central de la zona visible pel jugador, és el seu centre de poder.
- A partir del castell es calcula un radi de control, inicialment 1 (mesurat en "caselles de distància"). El radi va creixent a mesura que Lord Bot "regala" control al jugador.
- Se consideren caselles actives totes aquelles caselles que cauen dins del radi de control.

Cada casella pot contenir un edifici, que no es pot substituir ni destruir. Per tant, les decisions son permanents i s'han de planificar bé, sobretot al principi de la partida si es té poc espai.



Figura 13: Caselles actives amb radi 2

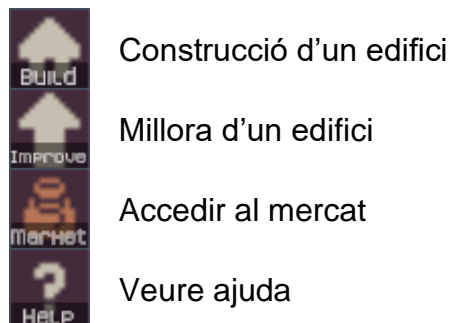
Els edificis ja construïts es poden millorar, obtenint bonus en rendiments o quantitat de recursos generats.

5.4. HUD de control

El HUD de control se situa a la part inferior i permet controlar les diferents accions.



Figura 14: Detall del HUD de control



En clicar una de les opcions es poden mostrar botons extra. Per exemple, si es clica a "Build" apareixen els botons de construcció



Figura 15: Botons de construcció

5.5. Altres

A banda de tots aquests elements hi ha uns components extra de la UI que son temporals. Aquests son la pantalla d'ajuda, la pantalla del mercat i els esdeveniments. Tots tres estan implementats de forma similar, directament a la UI.

5.5.1. Pantalla d'ajuda

La pantalla d'ajuda simplement mostra un text recollit del JSON de traduccions.

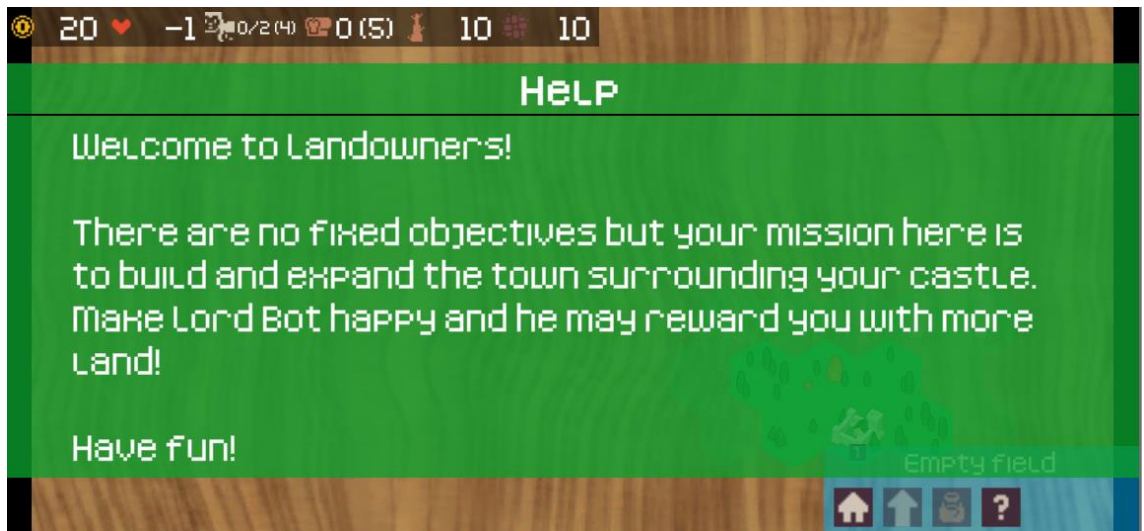


Figura 16: Pantalla d'ajuda

5.5.2. Pantalla d'esdeveniments

La pantalla d'esdeveniments s'activa quan el joc llença un esdeveniment i mostra el títol a la part superior, el text d'ambientació a la part central i un botó per a cada acció disponible definida a l'esdeveniment.

Els tres botons estan inclosos a la UI directament però es gestionen automàticament a partir del definit a l'esdeveniment. Per exemple, en la Figura 16 veiem un esdeveniment que només ha definit acció pels botons 2 i 3. Hi ha una sèrie d'accions disponibles (pay, payDebt, lossPopulation, etc) a partir de les quals es decideix el resultat (pagar recursos, perdre treballadors, ...).

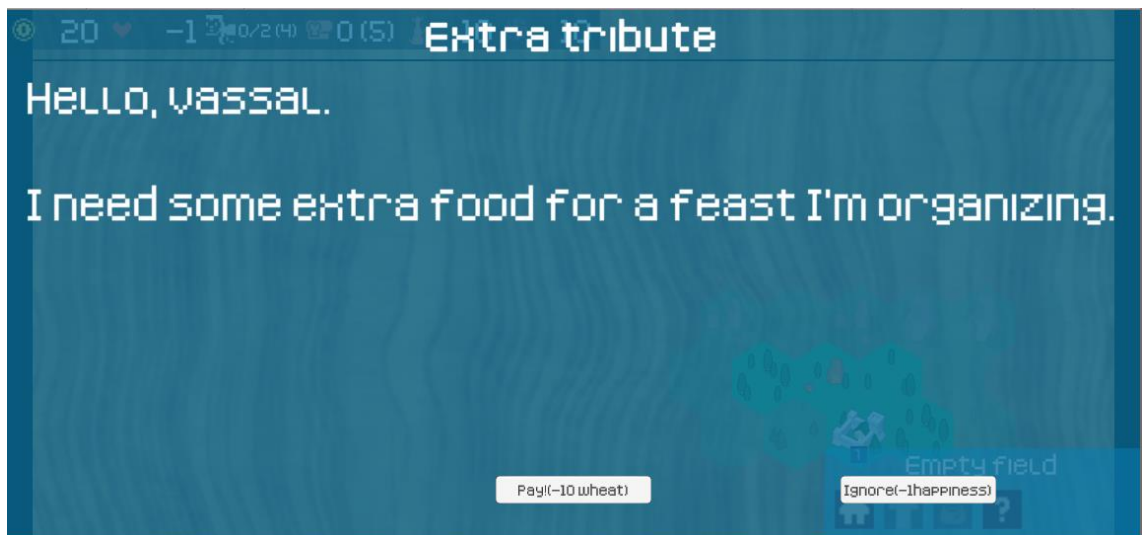


Figura 17: Esdeveniment amb dos accions possibles

5.5.3. Pantalla de mercat

En quant al mercat, és una pantalla inclosa a la UI però el contingut de la qual es genera com el HUD de recursos, automàticament. Per a cada recurs disponible, s'afegeix un prefab que permet comprar, vendre i intercanviar.

La pantalla només està activa si el marxant es troba a la vila. Cada cert temps decidirà si es queda o si ja s'ha avorrit i marxa, per lo que no sempre estarà disponible.



Figura 18: Pantalla de mercat

És una pantalla que podria donar molt joc per exemple afegint preus variables segons el que es va comprant i venent. Però, com sempre, el temps és el que és i s'ha fet de forma simple i amb preus fixos.

6. Manual d'usuari

6.1. Requeriments tècnics

Com que la base de Landowners és Unity, els requeriments deriven directament dels requeriments d'aquest. Al final, s'ha descartat tant la versió per a mòbils com la versió per a PSVita. Per tant, ens centrarem en els requeriments per a PC i Mac.

Segons informació oficial de Unity, la versió 2019.3 fixa els requeriments tècnics segons la següent taula⁸:

Sistema operatiu	Windows	macOS	Linux
Versió	Windows 7 (SP1+) o superior	Sierra 10.12+	Ubuntu 16.04 o superior
CPU	x86/x64 amb suport per a SSE2	x64 amb suport per a SSE2	x64 amb suport per a SSE2
API gràfica	DX10/DX11/DX12	Metal	OpenGL 3.2+/Vulkan

Taula 7: Requeriments tècnics

6.2. Menú principal

Després de visualitzar els diferents logos tindrem accés al menú principal. Hi ha tres botons disponibles amb els que podem començar una nova partida, accedir al menú d'opcions o bé sortir del joc.



Figura 19: Menú principal de Landowners

El menú està disponible també durant la partida i s'hi pot accedir si es prem la tecla 'Esc'.

⁸ <https://docs.unity3d.com/Manual/system-requirements.html>

6.3. Menú d'opcions

El menú d'opcions presenta algunes configuracions possibles com són:

- Canviar la resolució de pantalla
- Canviar l'idioma
- Possibilitat de jugar a pantalla completa o bé en una finestra
- Activar/desactivar el log de joc visible durant la partida
- Activar/desactivar rotació aleatòria de les caselles
- Activar/desactivar el fons de fusta durant la partida.

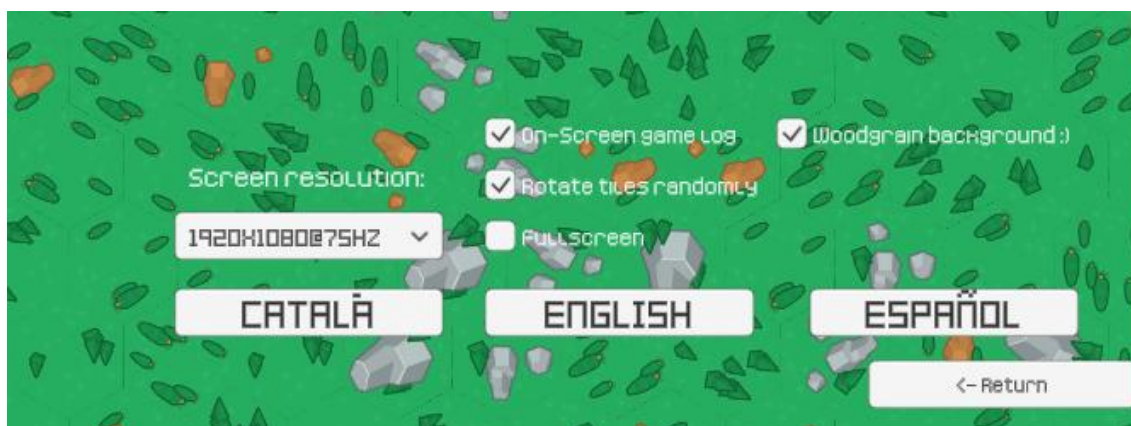


Figura 20: Menú d'opcions de Landowners

Algunes d'aquestes opcions estan disponibles durant la partida si es prem la tecla 'Esc'.

6.4. Configurant una partida

Abans de començar a jugar es mostra una pantalla amb algunes configuracions possibles (Figura 20).

Hi ha dos seccions principals, la secció "Recursos inicials" permet configurar els recursos que es tindran ja d'entrada al començar. És útil per configurar diferents escenaris més o menys complexos de superar.

En conjunció amb aquesta secció tenim la secció "Misc", que permet configurar altres aspectes més relacionats amb el *gameplay*, que passem a detallar:

- **Ratios d'esdeveniments** Indica el valor entre 0 i 1 que es tindrà en compte al calcular si s'ha de llençar un esdeveniment aleatori. Els números aleatoris menors o iguals que el valor indicat aquí desencadenen esdeveniment.
- **Temp. Tributs** Indica el temps en segons que desencadenarà el pagament de tributs anual a Lord Bot.
- **Recompensa felicitat (base)** Indica el valor base per a la recompensa per felicitat.

- **Recompensa felicitat (multiplicador)** Indica el valor que es multiplicarà pel valor base per saber quan s'han d'adjudicar les properes recompenses. (P.ex: si valor base 1 i multiplicador 5 es rebran les recompenses un cop la felicitat arribi a 5, 10, 15, ...)
- **Ratio màx pèrdua població** Indica el % màxim (0-1) de població que es pot arribar a perdre en un esdeveniment aleatori.
- **Valor final de partida (blat)** Indica el valor que marca el final de la partida per manca d'aliments
- **Valor final de partida (felicitat)** Indica el valor que marca el final de la partida per infelicitat dels vilatans i Lord Bot.
- **Aparició mercat** Indica el valor entre 0 i 1 amb la probabilitat que el mercat estigui disponible.
- **Avorriment del marxant** Indica el valor que es tindrà en compte per decidir si el marxant s'ha cansat d'estar a la vila. El valor anirà decreixent fins a arribar a 0 i, segons la configuració d'"Aparició mercat", es decidirà si el marxant marxa o es queda una estona més (reiniciant el comptador).

Jugant amb els valors de les dos seccions es poden aconseguir diferents nivells de dificultat. Per exemple, es pot desactivar completament el marxant indicant un 0 a "Aparició del mercat" complicant bastant el joc. O bé, pel contrari es pot indicar un 1 fent que estigui sempre disponible, simplificant la partida.

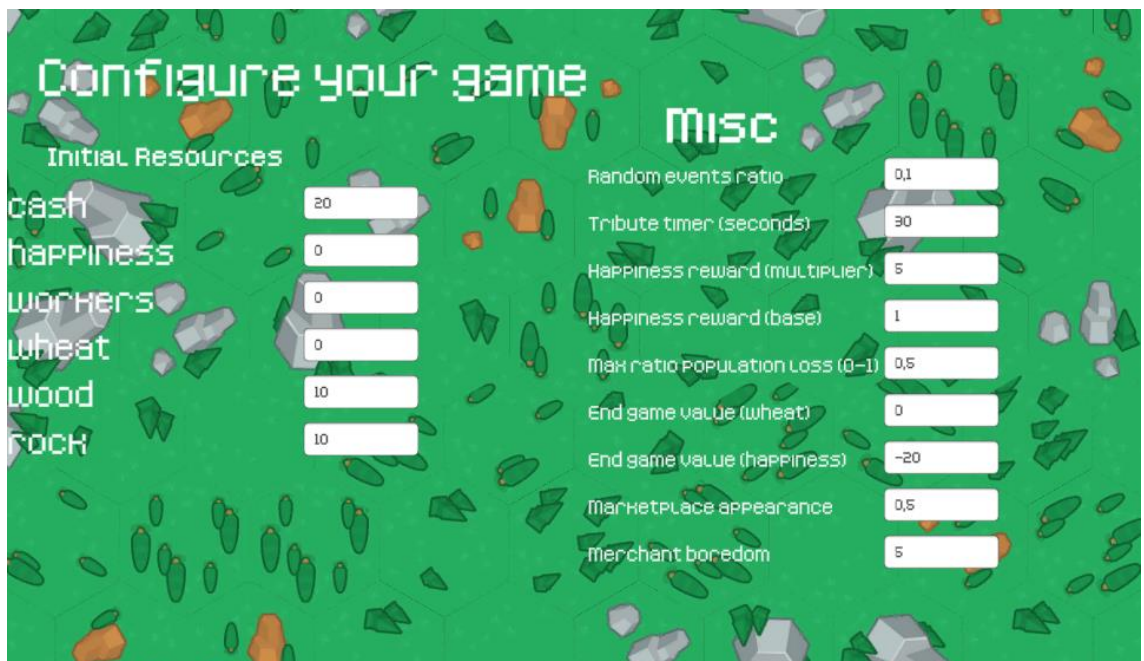
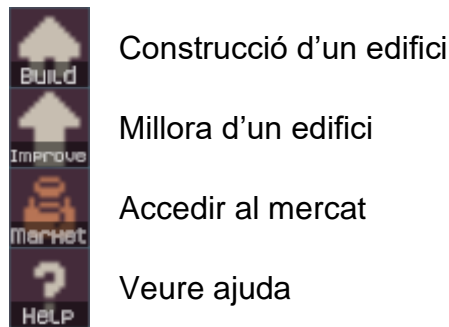


Figura 21: Configurant la partida

6.5. Jugant a Landowners

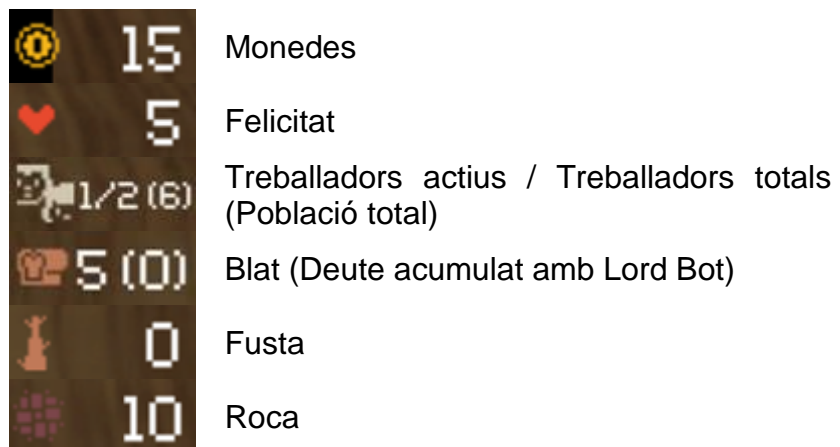
El concepte és ben simple i s'ha intentat construir de forma intuïtiva. Es disposa d'una parcel·la de terreny representada per les caselles hexagonals a les quals es té accés (no transparents).

Clicant en una d'aquestes caselles s'activa el HUD de control que, com ja hem vist, disposa de diverses opcions. La casella activa es marca amb un efecte visual.



6.5.1. Recursos

Els recursos son la base de qualsevol joc de construcció de ciutats, i *Landowners* no en podia ser menys. A la part superior de la pantalla es poden veure els recursos actuals, que passem a detallar ràpidament.

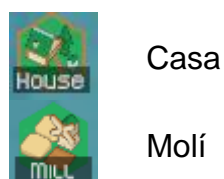


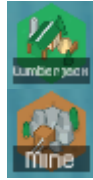
A la secció 6.5.5. referent als treballadors, es comentarà com es produeixen aquests recursos.

6.5.2. Construcció

ATENCIÓ! L'acció només estarà disponible si la casella està buida.

En clicar la opció "Construir" se'ns mostren els diferents edificis disponibles (Figura 23).





Llenyataire

Mina

Per construir un edifici, s'han de pagar els recursos corresponents, que s'indiquen al passar el ratolí per sobre.

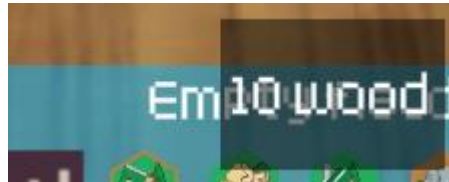


Figura 22: Cost del molí

Si es disposa dels recursos, en clicar un dels edificis disponibles, començarà la construcció. La construcció trigarà un temps en estar disponible i es mostrarà al mapa incompleta (Figura 22). Si es passa el ratolí per sobre es pot veure quant temps queda.

Un cop finalitzada la construcció, l'edifici començarà a buscar treballadors (si els requereix) i a produir quan els trobi.

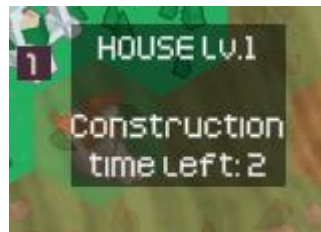


Figura 23: Edifici en construcció

Els nous edificis seran sempre de nivell 1.

Existeix un edifici especial que es construeix a l'inici de la partida automàticament: el castell. Només se'n pot tenir un i, per tant, la seva construcció està deshabilitada. El castell produeix felicitat a partir del nivell 2 (veure apartat 6.5.2).



Figura 24: Botons de construcció

6.5.3. Millora

ATENCIÓ! L'acció només estarà disponible si la casella conté un edifici millorable.

Per millorar un edifici, s'han de pagar els recursos corresponents, que s'indiquen al passar el ratolí per sobre:



Figura 25: Cost de millora del Castell

La millora trigarà a estar disponible i el funcionament és el mateix que el de l'acció "Construir": un cop s'han pagat els recursos es mostra l'edifici incomplet. Es pot consultar el temps restant passant el ratolí per sobre.

Les millores fan que l'edifici augmenti la producció i pugui encabir més treballadors (o més famílies en el cas de les cases). El nivell màxim en aquesta versió és 3 amb la excepció de les cases que només arriben al 2.

6.5.4. El mercat

ATENCIÓ! El marxant, com el seu nom indica, és un mercader ambulant que va fent la seva ruta i no estarà sempre disponible.

No hi ha forma d'afectar aquest comportament apart de la configuració inicial.

Per accedir al mercat cal que el marxant estigui a la vila. Al clicar el botó "Mercat" es pot accedir a una pantalla on es permet comprar, vendre i bescanviar els recursos disponibles: fusta, blat i roca.

Passant el ratolí per sobre de cada botó podem veure què comporta cada acció.

Compra

Es pot comprar un recurs per 2 monedes

ATENCIÓ! Només disponible si es tenen prou monedes



Figura 26: Comprant recursos

Venda

Es poden vendre 2 recursos iguals per obtenir 1 moneda.

ATENCIÓ! Només disponible si es tenen prou recursos



Figura 27: Venent recursos

Bescanvi

Es poden bescanviar 2 recursos iguals qualsevol per 1 d'un altre tipus

ATENCIÓ! Només disponible si es tenen recursos



Figura 28: Bescanviant recursos

6.5.5. Treballadors

Landowners gestiona automàticament tant la població total com els treballadors disponibles.

Tot i que el castell permet mantenir un cert número de famílies, aviat s'haurà d'ampliar aquest número construint cases. Entre els membres de la família pot ser que alguns estiguin ja en edat de treballar i, per tant, passen a comptar com a treballadors disponibles.

Quan un edifici troba un treballador disponible i el "contracta", aquest passa a ser considerat treballador actiu. A més a més, es produeix 1 únic recurs de felicitat. Lord Bot, molt en el fons, vol que la vila prosperi⁹.

A partir del moment en que un edifici té almenys un treballador, comença a produir. Cada cert temps, produirà els recursos corresponents.







Figura 29: Una mina acaba de produir un recurs roca

El temps restant per acabar la producció es pot consultar si es passa el ratolí per sobre de l'edifici.



Figura 30: Un molí que ha completat 6/10 de la producció

El recurs produït depèn del tipus d'edifici:

	Felicitat (a partir del nv.2)
	Blat
	Fusta
	Roca

6.5.6. Esdeveniments

Hi ha dos tipus d'esdeveniments: aletaroris i fixes. Es pot configurar parcialment el comportament del sistema al respecte des de la pantalla de configuració de la partida.

⁹ Però no li heu de treure mai el tema, no ho reconeixerà!

Esdeveniments aleatoris

Cada cert temps el sistema genera un esdeveniment que escull de forma aleatòria de la llista d'esdeveniments definits.

Quan es llença un esdeveniment aleatori, aquest es mostra a la pantalla juntament amb les diferents accions possibles. Al text de l'acció pot aparèixer el resultat entre parèntesi (p.ex: pagar una quantitat d'un recurs).

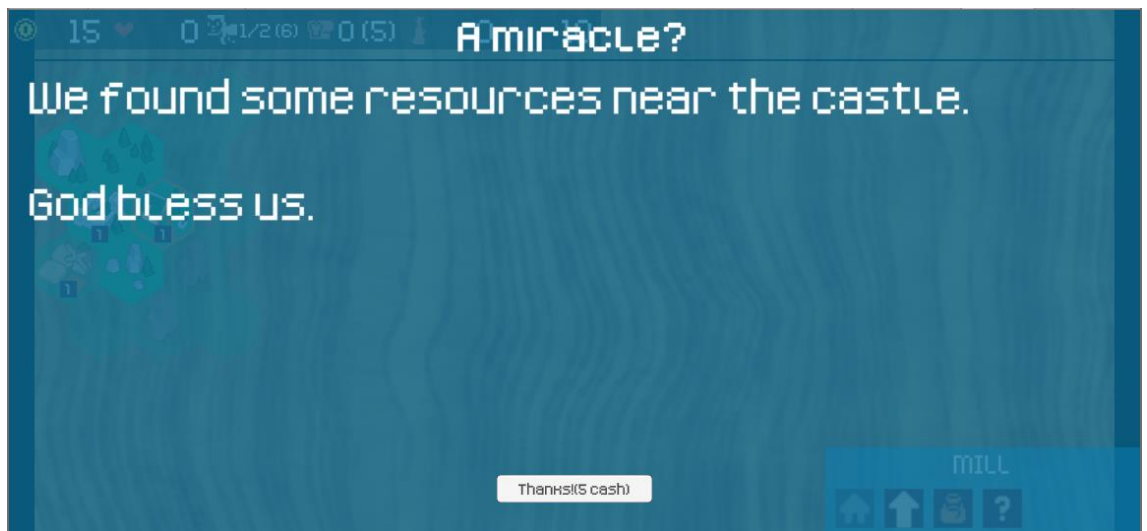


Figura 31: Esdeveniment aleatori amb una acció possible

Un cop seleccionada l'acció, es reprèn el joc normalment.

Esdeveniments fixes

Dintre dels esdeveniment fixes trobem el pagament dels tributs i els esdeveniments de final de partida. El primer es llença cada 30 segons (configurable) representant el pagament del delme al pas d'un any.



Figura 32: Pagament del delme

Es pot aplaçar el pagament, acumulant deute però això afectarà negativament la felicitat. Aquesta opció serà especialment útil als primers moments de la partida.

Per altra banda tenim els esdeveniments de final de partida que es llencen inevitablement al complir un dels requisits. La única acció disponible és acabar la partida, acció que ens porta a la pantalla de final de partida.



Figura 33: Esdeveniment de final de partida

7. Conclusions

Tot i tenir experiència prèvia en projectes d'aquest tipus, el repte ha estat important especialment en la gestió del calendari. El principal objectiu del projecte, hereu d'aquesta experiència passada, era simplificar al màxim la idea justament per equilibrar el pes del temps limitat.

Però tot i així, no ha estat fàcil. Certament hi ha hagut imprevistos (hola, covid19¹⁰) però s'hauria pogut reduir els seus efectes amb una millor planificació.

Respecte als objectius que ens havíem fixat, creiem que s'han superat tots amb èxit, però especialment el disseny i la rejugabilitat. És cert que en aquest segon aspecte encara hi ha feina per fer, com veurem de seguida, però ara per ara el rang d'experimentació amb el joc és bastant ampli. Això no vol dir que no hi hagi marge de millora ni que el joc resultant sigui perfecte. Podríem dir fins i tot que la resta d'objectius s'han aconseguit només en part. Encara es pot ampliar la llista de variables configurables al principi de la partida i, sobretot, la quantitat i varietat d'esdeveniments. No s'han pogut implementar del tot, per exemple, els desastres (incendis, etc) que provocarien destrucció d'edificis obligant a invertir recursos en la reconstrucció. Per altra banda, la interfície no és gaire agradable i hauria estat millor fer-la més dinàmica (per exemple: que aparegués allà on està el punter del ratolí). Amb tot, estem contents amb el resultat.

Tots aquests detalls que han quedat fora, s'haurien pogut incloure segurament amb una millor planificació. El càlcul del temps realitzat en un primer moment no va ser realista i es va haver d'anar adaptant tot. A més, sobretot en la última PAC, s'ha hagut d'improvisar una mica i desviar-se de la planificació establerta. Pel que fa a la metodologia, ni que pugui haver alguna excepció, creiem que s'ha seguit fil per randa i, de fet, és una de les claus del resultat. Fer que cada component sigui pràcticament independent, utilitzar els prefabs i centralitzar les dades en una classe estàtica ha reduït àmpliament la complexitat del projecte podent aplicar solucions ràpides i quirúrgiques (contrarestant les presses).

El projecte té molt potencial futur. Alguns aspectes ja els hem vist: més configurabilitat, millorar la UI, més i millors esdeveniments, ... Altres línies de treball possibles inclouen guardar els json que defineixen els diferents aspectes en un lloc accessible per l'usuari (permetent l'aparició de *mods*), guardar partides o introduir un nou mode de joc amb escenaris predefinits (mode Història/Campanya). Respecte a aquest

¹⁰ La pandèmia i el confinament derivat ni que, en un primer moment, pugui semblar una oportunitat per dedicar més hores, ha esdevingut una font d'estrès important en quant a temes laborals i personals. Per no parlar de la incertesa que ha provocat en moltes persones. I crec que parlo per la pràctica totalitat de la Humanitat.

darrer punt seria interessant el que es comenta al punt 3.4 d'introduir altres ambientacions. Això permetria afegir més varietat al joc ja que els recursos, esdeveniments, etc serien diferents segons a quina època es fa referència (com passa als jocs de la saga Civilization).

8. Glossari

City-building (game)

Joc de construcció de ciutats

CPU

Unitat central de processament, sol fer referència al processador o microprocessador central d'una màquina (*Central Processing Unit*, en anglès)

GPU

Unitat de procés de gràfics, sol fer referència al coprocessador de gràfics, generalment contingut en una tarjeta gràfica (*Graphics Processing Unit*, en anglès)

HUD

Head-Up Display, en anglès. Informació mostrada a la pantalla referent a l'estat del joc (vides, punts, etc)

In-house

Desenvolupat internament, codi propietari no alliberat al públic

Mods

Modificacions d'un joc fetes, generalment, per usuaris. Poden incloure nous nivells, canvis o millores gràfiques, etc.

Prefab

Component reutilitzable

NES

Nintendo Entertainment System. Consola de tercera generació (8-bit) creada per la companyia japonesa Nintendo i comercialitzada a partir de l'any 1983. Coneguda com a *Famicom* (Family Computer) al Japó.

Real-time strategy (game)

Joc d'estratègia en temps real

RTS

veure *real-time strategy*

sandbox

Un entorn sandbox (caixa de sorra) sol referir un entorn, normalment aïllat, de proves o experimental. En referència als videojocs, però, es tracta d'un gènere amb partides obertes i no-lineals. Dos exemples ben coneguts serien *Minecraft* i *Terraria*

9. Bibliografia

1. **Machkovech, Sam.** Ars Technica. *Ars Technica*. [En línia] 26 / 12 / 2018. <https://arstechnica.com/gaming/2018/12/you-can-now-download-the-long-lost-and-unfinished-simcity-nes-port/>.
2. **Tilted Mill Entertainment.** *Tilted Mill Entertainment*. [Online] [Cited: 04 28, 2020.] <http://tiltedmill.com/games/>.
3. **City-building game - Wikipedia.** *City-building game - Wikipedia*. [En línia] https://en.wikipedia.org/wiki/City-building_game.
4. **City Building (series) - Wikipedia.** *City Building (series) - Wikipedia*. [En línia] [https://en.wikipedia.org/wiki/City_Building_\(series\)](https://en.wikipedia.org/wiki/City_Building_(series)).