

Estudi i desenvolupament d'un videojoc City Builder amb Unity.

Adrià Raventós Llopart

Resum— El projecte consisteix en el desenvolupament d'un videojoc del gènere City Builder amb el motor gràfic Unity. City Builder és un gènere de videojocs de simulació, centrat en la construcció i gestió d'una ciutat a partir dels edificis que construïm i petites microgestions de la ciutat. Cada tipus d'edifici té les seves funcionalitats úniques que aporten els elements claus necessaris per a poder desenvolupar la ciutat amb èxit. El projecte utilitza el motor gràfic Unity i C# com a llenguatge de programació. S'ha treballat amb la metodologia àgil Kanban. El resultat és un prototip jugable per a PC.

Paraules clau—Unity, city, builder, C#, videojoc, simulació, edifici, mapa, tile, construir, software, agile, Kanban, Dijkstra

Abstract— The project consists of the development of a video game of the City Builder genre with the Unity engine. City Builder is a genre of simulation video games, focused on building and managing a city from the buildings we build and small micromanages in the city. Each type of building has its own unique features that provide the key elements needed to be able to develop the city successfully. The project uses Unity engine and C# as the programming language. We have worked with the agile Kanban methodology. The result is an playable prototype for PC.

Index Terms— Unity, city, builder, C#, videogame, simulation, building, map, tile, build, software, agile, Kanban, Dijkstra



1 INTRODUCCIÓ

El projecte consisteix en l'estudi i desenvolupament d'un videojoc del gènere dels City Builder [1] amb Unity [2].

City Builder és un gènere de videojocs de simulació, centrat en la construcció i gestió d'una ciutat a partir dels edificis que construïm i petites microgestions d'aquesta ciutat. Cada tipus d'edifici té les seves funcionalitats úniques que aporten els elements claus necessaris per a poder desenvolupar la ciutat amb èxit.

L'objectiu dels videojocs City Builders és un concepte una mica abstracte, ja que en la majoria d'ells aquest depèn del punt de vista del jugador. Solen ser jocs que donen llibertat al jugador perquè puguin gaudir-lo a la seva manera, autoimposant-se cada jugador un objectiu personal diferent. L'objectiu més comú en els City Builders és fer créixer la teva ciutat, alhora que intentes que aquesta superi els problemes que poden sorgir, com per exemple, la falta de diners per mala gestió de la ciutat, incendis, falta d'aliments, criminalitat, malalties...

El videojoc que inspira aquest treball és "Caesar III" [3], un City Builder de 1998 que destaca per la seva temàtica de l'antiga Roma i la gran varietat d'edificis, cadascun d'ells amb una funcionalitat única que feia l'experiència

de joc més entretinguda.

2 ESTAT DE L'ART

El gènere del City Builder actualment és un gènere que està en constant expansió i creixement. És un gènere amb menys èxit de vendes i interès que altres gèneres de videojocs com poden ser per exemple els shooters [4] o els RPG [5]... Això implica que es desenvolupen menys City Builders que no pas videojocs d'altres gèneres més exitosos. Tot i així el nombre de City Builders que es llancen al mercat cada any és prou gran per a poder afirmar que és un mercat amb una gran competència.

També és important remarcar que el nombre de videojocs indies [6] que surten al mercat ha augmentat en els últims anys, provocant que la competència sigui més gran cada any i el que el nivell d'innovació dels videojocs augmenti considerablement, ja que un videojoc indie té molt més marge d'error que el que pot tenir una empresa gran. Aquest augment del nombre de videojocs indies s'ha donat pel fet que hi hagi software gratuït de desenvolupament de videojocs com és Unity o Unreal Engine [7], que ofereixen un ventall d'eines molt completes que ajuden al desenvolupament indie.

En l'enllaç [8] es pot veure un llistat dels videojocs City Builder més rellevants de tota la història. L'enllaç [9] ens porta al portal "Steam", la plataforma de venda online de

-
- E-mail de contacte: adria.raventos@e-campus.uab.cat
 - Menció realitzada: Enginyeria del Software
 - Treball tutoritzat per: Francisco Javier Sánchez Pujadas (Ciències de la computació)
 - Curs 2019/20

videojocs més reconeguda, on estem aplicant un filtre de cerca per veure només els videojocs City Builder en venda. Si examinem la pàgina veurem que el nombre de videojocs City Builder en venda al portal Steam és bastant gran (>400 resultats).

No existeix una classificació oficial dels diferents tipus de City Builders que existeixen, ja que el concepte City Builder és massa obert. Per intentar etiquetar-los, he creat unes possibles classificacions:

- Per temàtica del videojoc:
 - Períodes històrics de la humanitat:
 - Antic Egipte
 - Antiga Roma
 - Edat medieval
 - Prehistòria
 - ...
 - Fantasia:
 - Mitologia vikinga
 - Vida a l'espai exterior
 - Defensa contra zombies
 - ...
 - Simulació d'elements del món real
 - Indústries
 - Gestió de trens
 - Construcció de zoos
 - Presons
 - Hospitals
 - ...
- Per funcionament dels edificis
 - Per àrea d'influència
 - Per proximitat amb altres edificis
 - Utilitzant intermediaris (ciutadans)
- Per tipus de problemes a controlar:
 - Incendis
 - Malalties
 - Guerres
 - Revoltes
 - Fam
 - Pobresa
 - ...
- Per recursos ingame:
 - Només gestió d'or
 - Or + altres recursos (menjar, fusta, pedra...)
- Per tipus de gràfics:
 - 2D ortogràfic
 - 3D
- Contenen o no elements de guerra
- Contenen o no esdeveniments aleatoris
- Contenen o no un mode campanya

3 OBJECTIUS

L'objectiu principal que s'ha buscat assolir en el projecte ha sigut crear un prototip jugable d'un City Builder per a PC.

En aquest prototip inicialment ha d'aparèixer un menú que permeti al jugador registrar el seu nom i li permeti crear un nou mapa de la mida que ell desitgi. Aleshores el sistema ha de crear un mapa format per tiles individuals. A partir d'aquest punt el jugador pot començar la partida.

El jugador ha de poder construir i destruir edificis pagant el seu cost. Els edificis que estaran disponibles per a construir en el prototip són:

- Casa: permeten acollir habitants. S'alimenten periòdicament de menjar.
- Carretera: Crea unions entre els edificis.
- Granja: Contracta treballadors propers, genera menjar periòdicament i ven aquest menjar als mercats propers.
- Mercat: Contracta treballadors propers i ven menjar a les cases properes. Permet al jugador guanyar or amb el menjar vengut.

En el prototip hi ha d'haver gestió de l'or del jugador, dels habitants total de la ciutat, treballadors disponibles i del menjar. El jugador ha de poder veure en tot moment el valor actual d'aquests. També ha d'haver-hi un menú de pausa i un menú de puntuacions que mostri l'històric de millors puntuacions.

4 METODOLOGIA

Pel desenvolupament d'aquest projecte s'ha decidit utilitzar metodologies agile [10], ja que permet treballar amb iteracions curtes. Es treballa amb cicles d'iteracions format per disseny, desenvolupament i testeig.

Específicament, s'ha decidit utilitzar la metodologia agile Kanban [11], [12]. Kanban es basa en les següents pràctiques:

- **Visualitzar el treball a fer:** Utilitza un sistema de targetes per registrar i controlar la feina a realitzar, que s'organitzen en tres columnes: To do; In progress; Done. Aquestes columnes de targetes, a diferència de Scrum, no es reinicien en acabar una iteració.
- **Limitar el treball en curs:** L'objectiu és que cada treballador tingui poques tasques a fer al mateix moment. D'aquesta manera el treballador es pot centrar en una tasca.
- **Dirigir i gestionar el flux:** Volem aconseguir que les tasques es realitzin en poc temps i per tant hi hagi moviment continu, ràpid i ininterromput de feina.

Els motius pel qual s'ha escollit Kanban en lloc d'altres metodologies són:

- Kanban a diferència d'altres metodologies agile permet la feina en solitari. D'aquesta manera eliminem la possibilitat d'escollir com a metodologia Scrum, que està centrat en reunions d'equip o Extreme programming, que una de les seves bases és la programació en parelles o el feedback amb el client.
- Sento afinitat amb el pensament de centrar-me en poques tasques per a poder realitzar aquestes correctament i mantenir un bon ritme de treball.

Les tecnologies a utilitzar en el projecte són:

- Engine Unity versió 2019.3.3.
- IDE Microsoft Visual Studio [13] versió Community 2019, 16.4.5. Llenguatge C# [14].
- Bitbucket [15] pel control de versions.
- Trello [16] per la gestió i enregistrament de tasques.
- Draw.io [17] per al disseny de classes.

5 PLANIFICACIÓ

5.1 Planificació de les tasques

La planificació de tasques del projecte està formada per una planificació inicial i seguidament varies iteracions. Cada iteració del projecte, en ser iteracions agile, estan formades per disseny, implementació i testeig.

La planificació del projecte s'ha dividit en 5 grans fases:

1. **Estudi:** La fase prèvia a la programació, formada per: Disseny del calendari, estudi inicial del projecte, disseny de classes i captació de requisits.
2. **Base del projecte:** Desenvolupament dels elements base del projecte: Creació de mapa format per tiles, moviment de la càmera i ratolí, botons de construcció, preconstrucció, construcció i UI del jugador.
3. **Funcionalitats:** Primeres funcionalitats que afegeixen jugabilitat al projecte: House, Roads, Destrucció d'edificis, menús, Serialització puntuacions,
4. **Mecàniques de joc:** Elements i mecàniques de joc que afegeixen jugabilitat real: Granger, mercat, Dijkstra, mecàniques de contractar treballadors, generar i vendre menjar.
5. **Detalls:** Pulir detalls del joc: Panell informació d'edificis, transparències dels edificis al seleccionarlos, pulir tots els valors ingame dels edificis, comprovar que no hi hagin bugs.

La planificació detallada del projecte i les seves dates es

poden veure a l'apèndix del projecte.

A l'inici del projecte es va definir una planificació de tasques. A mesura que el projecte avançava, aquesta planificació va sofrir algunes modificacions. A l'apèndix es mostra el diagrama de Gantt inicial i el diagrama de Gantt final per a poder comparar aquests canvis.

Els canvis que va sofrir la planificació són els següents:

- A la fase 2 del projecte (base del projecte), es va decidir dividir la iteració Construcció en dues: Construcció i Preconstrucció. Això és perquè quan es va planificar la construcció no es va tenir prou en compte la complexitat de la preconstrucció. D'aquesta manera s'han separat dues funcionalitats complexes en dues iteracions diferents.
- També a la fase 2 del projecte es va decidir canviar la data de realització de la iteració de la UI del jugador per a realitzar-la abans del previst inicialment. Això és perquè es va decidir que era important poder veure els valors del jugador com més aviat millor, ja que així es podia avaluar visualment si els valors s'estaven modificant correctament.
- A la fase 4 del projecte (mecàniques de joc) es va afegir una nova iteració amb l'algoritme de Dijkstra [18]. Inicialment l'algoritme de Dijkstra no formava part de la planificació. En el moment de dissenyar en profunditat la granja i el mercat, es va veure que era necessari poder aplicar un algoritme de camí més curt [19] per conèixer les distàncies entre dos edificis.

5.2 Pressupost

A continuació mostro el cost que tindria el projecte. En el pressupost només es té en compte que hi hauria un sol enginyer informàtic que faria totes les tasques:

Nomines	Cost/hora	Hores	Persones	Cost
Enginyer informàtic	17,00 €	300	1	5.100,00 €
Seguretat social				1.581,00 €
Total nomines				6.681,00 €

Taula. 1 Nomina i Seguretat Social

Com a cost directe es mostren dos aspectes necessaris per al producte final: la música i el disseny gràfic. Aquests es farien contractant un servei extern o un treballador freelance. Els següents costos són una aproximació, els costos reals poden variar en funció de la complexitat de la feina a realitzar (quantitat de dibuixos, qualitat del detall, nombre de cançons, nombre d'instruments a gravar...) i en funció del preu particular [24], [25].

Cost Música	Cançons	Cost/cançó	Cost
Composició	15	150,00 €	2.250,00 €
Gravació	15	40,00 €	600,00 €
Masterització	15	60,00 €	900,00 €
Total			3.750,00 €

Taula. 2. Cost Música

Cost disseny	Dibuixos	Cost/disseny	Cost
Disseny	30	150,00 €	4.500,00 €
Total			4.500,00 €

Taula. 3. Cost Disseny gràfic

Cost directe	Preu
Música	3.750,00 €
Disseny gràfic	4.500,00 €
Cost directe	8.250,00 €

Taula. 4. Cost Directe

Cost total	14.931,00 €
------------	-------------

Taula. 5. Cost Total

Si s'hagués de llogar un local per a treballar, aquesta seria una aproximació del cost indirecte i el nou cost total.

Cost indirecte	Cost/mes	Mesos	Cost
Lloguer local	500,00 €	4	2.000,00 €
Tarifa internet	12,40 €	4	49,60 €
Electricitat	20,00 €	4	80,00 €
Aigua	5,00 €	4	20,00 €
Total			2.149,60 €

Taula. 6. Cost indirecte

Cost total	19.830,60 €
------------	-------------

Taula. 7. Nou Cost total

5.3 Riscos

A continuació es llistaran els possibles riscos del projecte, que poden generar problemes:

Errors de programació:

- **Origen:** Errors de programació en el producte.
- **Descripció:** Errors de programació en el producte no descoberts o no arreglats.
- **Impacte sobre el projecte:** Pot causar que no es pugui jugar correctament el joc, que es tanqui inesperadament, que hi hagi funcionalitats que no funcionin degudament... Necessitat de descobrir l'error, arreglar-lo i llançar una nova versió del joc amb l'error arreglat.
- **Impacte:** Crític.

Clients insatsfets:

- **Origen:** Insatisfacció dels jugadors
- **Descripció:** Clients insatsfets per diferents possibles motius: joc poc entretingut/divertit, falta d'objectius, falta de recompenses ingame, falta de

contingut, mala implementació de mecàniques, errors...

- **Impacte sobre el projecte:** Pèrdua de jugadors i interès global pel joc. Males crítiques. Necessari escoltar les crítiques dels jugadors i estudiar una solució específica i idònia al respecte.
- **Impacte:** Crític.

Finançament:

- **Origen:** Falta de finançament.
- **Descripció:** Problemes relacionats amb el finançament o pressupost en el cas que aquest sigui necessari, com podrien ser falta de finançament, pocs o cap inversors en el projecte, mal càlcul del pressupost del projecte...
- **Impacte sobre el projecte:** Impossibilitat de continuar amb el desenvolupament del projecte. Obliga a buscar noves fonts de finançament i plantejar el pressupost del projecte.
- **Impacte:** Crític.

Competidors:

- **Origen:** Videojocs del mateix gènere
- **Descripció:** Tots els videojocs ja existents o en desenvolupament que comparteixen el mateix gènere o idea, poden suposar competència directa a l'hora de vendre el joc.
- **Impacte sobre el projecte:** Obliga al projecte a aportar idees innovadores o diferenciadores per destacar i així aconseguir bones vendes.
- **Impacte:** Alt.

Falta d'experiència:

- **Origen:** Falta d'experiència.
- **Descripció:** Problemes que puguin sorgir en el desenvolupament generats per la falta d'experiència amb relació a la gestió de projectes, desenvolupament o coneixement de les tecnologies.
- **Impacte sobre el projecte:** Pot generar problemes molt variats com per exemple: temps esperat de desenvolupament erroni, objectius incomplets, desviament de la idea inicial, augment del cost i temps estimat del projecte...
- **Impacte:** Alt.

6 DESENVOLUPAMENT

Els reptes més important d'aquest projecte han sigut principalment: l'aprenentatge de les eines que ofereix Unity i la seva API, crear un bon disseny perquè l'estructura del joc sigui robusta i sigui fàcil afegir noves funcionalitats, i que la implementació del codi i del test sigui correcta i completa.

6.1 Requisits

A la fase inicial del projecte es van recollir el llistat de requisits que havia de complir el prototip. Aquests requisits es van seguir a l'hora de dissenyar i implementar el projecte. A la fase final del projecte s'ha actualitzat el llistat de requisits perquè aquests siguin complets, ja que hi ha hagut requisits que han anat sorgint en el desenvolupament.

A continuació llistaré de forma molt resumida, els requisits principals que ha de complir el prototip:

- El sistema ha de permetre al jugador poder generar un mapa format per tiles.
- El sistema ha de permetre al jugador poder controlar el moviment de la càmera amb el ratolí.
- El sistema ha de permetre al jugador poder seleccionar quin edifici vol construir.
- El sistema ha de permetre al jugador poder construir un edifici seleccionat en una àrea definida pel jugador.
- El sistema ha de permetre al jugador poder destruir edificis.
- El sistema ha de permetre al jugador poder visualitzar els recursos: or, habitants totals, habitants sense feina i menjar.
- El sistema ha de permetre construir cases on visquin els habitants de la ciutat.
- El sistema ha de permetre que els habitants s'alimentin periòdicament.
- El sistema ha de permetre construir carreteres que serveixin com a connexió amb entre edificis.
- El sistema ha de permetre construir granges que produeixin menjar periòdicament.
- El sistema ha de permetre construir mercats que periòdicament venguin menjar a les cases més properes.
- El sistema ha de permetre que les granges i els mercats contractin treballadors de les cases més properes amb treballadors disponibles.
- El sistema ha de permetre utilitzar l'algoritme Dijkstra per a calcular el camí més curt entre dos edificis connectats per carretera.
- El sistema ha de permetre calcular el rendiment d'un edifici basat en el percentatge de treballadors/treballadors necessaris que tingui.
- El sistema ha de permetre mostrar un menú inicial del joc que permeti crear un nou mapa.
- El sistema ha de permetre mostrar un menú de pausa del joc.
- El sistema ha de permetre mostrar un menú de puntuacions on es mostrin les 5 millors puntuacions i el nom del jugador, ordenats de millor a pitjor.

cions i el nom del jugador, ordenats de millor a pitjor.

- El sistema ha de permetre calcular la puntuació actual del jugador. El sistema ha de permetre guardar i carregar puntuacions.
- El sistema ha de permetre mostrar un panell d'informació de cada edifici.

Requisits no funcionals:

- El videojoc ha de ser funcional per a sistemes operatius Windows.
- La UI del videojoc ha de ser intuïtiva.
- El videojoc ha de suposar un repte per al jugador.
- El videojoc ha de ser divertit.
- El videojoc ha de ser funcional.
- El videojoc no ha de contenir errors.

El llistat complet dels requisits es pot veure en el dossier del projecte.

6.2 Arquitectura i disseny

A l'apèndix del projecte es pot veure el diagrama de classes inicial, el diagrama de classes complet i el diagrama de classes complet sense mostrar els mètodes ni els atributs.

En el projecte s'han utilitzat dos patrons de disseny:

- **Singleton:** [21] Les classes que només necessiten una instància utilitzen el patró Singleton per a assegurar la seva instància única i per a poder accedir a elles més globalment.
- **State:** [22] S'utilitza State per a canviar fàcilment de mode el MouseMode entre MouseConstruct i MouseDestroy.

A continuació s'explicaran les funcionalitats que conté el prototip:

Elements bàsics:

- Generació d'un nou **terreny/mapa** d'una mida X,Y seleccionada pel jugador. El mapa està format per **tiles** individuals.

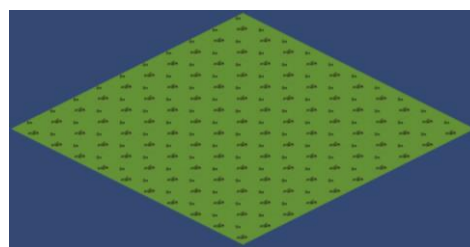


Fig. 1. Mapa generat de mida mínim 10x10

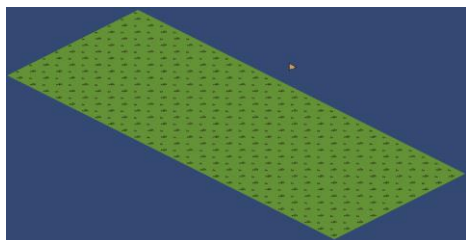


Fig. 2. Mapa generat de mida 25x10

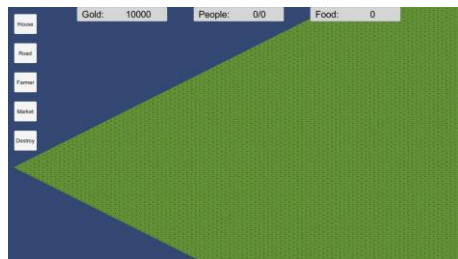


Fig. 3. Mapa generat de mida màxim 100x100. No pot aparèixer tot per pantalla, ja que la distància màxima de la càmera està limitada.

- Moviment de la **càmera** en funció de la posició del ratolí en pantalla i zoom de la càmera en funció del moviment de la rodeta del ratolí.
- **Preconstrucció:** El jugador selecciona amb els botons de construcció quin edifici vol construir. Un cop seleccionat un edifici, mentre el jugador premi el botó esquerre del ratolí, apareixerà l'àrea de preconstrucció que permet al jugador visualitzar l'àrea del que construirà. Aquesta àrea va des de la tile inicial on ha començat a clicar, fins a la tile actual on té el ratolí. L'àrea de preconstrucció s'actualitza cada cop que el jugador mou el ratolí de posició. També apareix un comptador al costat del punter del ratolí que mostra el cost actual de la construcció.



Fig. 4. Àrea de preconstrucció i cost

Cada tile de l'àrea canvia de color en funció de si és possible o no construir en aquella zona.

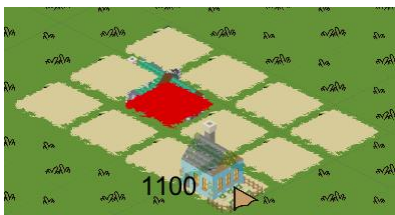


Fig. 5. Àrea de preconstrucció on es mostra en vermell que no es pot construir un dels edificis perquè ja hi ha un edifici construït en aquella tile.

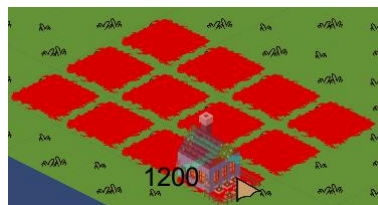


Fig. 6. Àrea de preconstrucció on es mostra tota l'àrea en vermell perquè no es pot construir cap edifici, ja que el jugador no té l'or necessari per pagar el cost.

- **Construcció:** Mentre el jugador estigui preconstruint, quan deixi de prémer el botó esquerre del ratolí, es construeixen tots els edificis que es puguin construir, a sobre de les tiles corresponents. A l'hora de permetre o no una construcció es té en compte el cost dels edificis, les monedes del jugador, i si la tile on es vol construir és buida. L'àrea de construcció es defineix per la posició inicial i final del ratolí en començar a clicar i deixar de clicar el botó esquerre.



Fig. 7. Construcció de cases finalitzada

Edificis:

Hi ha 3 punts a explicar per a cada edifici: el que ocorre al moment que construeixes un edifici, el que ocorre durant cada iteració interna del joc i el que ocorre en destruir un edifici.

- **Casa:**

En construir: La casa rep el nombre màxim de persones que pot rebre i això fa augmentar el nombre d'habitants disponibles per a treballar i el nombre d'habitants total de la ciutat. Obliga a totes les empreses construïdes que no tenen el màxim de treballadors a comprovar si poden contractar.

A cada iteració: Cada 7 segons, s'alimenten de 3 unitats de menjar que tinguin emmagatzemades. Si l'edifici passa 35 segons sense menjar, es destrueix, representant que els habitants han mort de gana.

En destruir: Tots els habitants de la casa marxen de la ciutat, disminuint així el nombre d'habitants disponibles i habitants total de la ciutat. Si aquella casa proporcionava treballadors a alguna empresa, s'acomiaden aquells treballadors de l'empresa i obliga a l'empresa a realitzar de nou la funció de contractar treballadors.

- **Granja:**

En construir: Calcula per a cada casa amb treballadors disponibles i que estigui connectada per un camí amb la granja, el camí més curt des de la granja fins a aquella casa (Dijkstra). Ordena els resultats de menor distància a major distància. Contracta treballadors de les cases del llistat fins a complir el nombre de treballadors necessaris i per tant, va reduint el nombre de treballadors disponibles de la ciutat.

A cada iteració: La granja cada X segons realitza accions que explicaré a continuació. El temps que triga a realitzar aquestes accions depèn d'un càlcul entre el valor mínim (2s) i màxim (7s) de temps i del nombre de treballadors que té la granja. Si té 0 treballadors, no pot fer cap acció. La granja genera menjar i l'emmagatzema. Seguidament, si té menjar emmagatzemat, busca el mercat més proper, connectat per un camí amb la granja, que no estigui a màxima capacitat i li ven el menjar produït.

En destruir: Elimina el menjar emmagatzemat que no s'hagi venut. Acomiada tots els treballadors i aquests retornen com a treballadors disponibles de cada casa que estigues proporcionant treballadors a aquella granja, per tant, augmenta el nombre de treballadors disponibles de la ciutat.

- **Mercat:**

En construir: Calcula per a cada casa amb treballadors disponibles i que estigui connectada per un camí amb el mercat, el camí més curt des del mercat fins a aquella casa (Dijkstra). Ordena els resultats de menor distància a major distància. Contracta treballadors de les cases del llistat fins a complir el nombre de treballadors necessaris i per tant, va reduint el nombre de treballadors disponibles de la ciutat.

A cada iteració: El mercat cada X segons realitza accions que explicaré a continuació. El temps que triga a realitzar aquestes accions depèn d'un càlcul entre el valor mínim (2s) i màxim (7s) de temps i del nombre de treballadors que té el mercat. Si té 0 treballadors, no pot fer cap acció. Si té menjar emmagatzemat, crea un llistat de totes les cases que no estiguin a màxima capacitat de menjar i que estiguin connectades per un camí amb el mercat. Ordena el llistat per distància més propera i ven el màxim de menjar a cada casa del llistat, fins a quedar-se sense existències o fins que totes les cases estiguin a màxima capacitat. El menjar venut genera or per al jugador.

En destruir: Elimina el menjar emmagatzemat que no s'hagi venut. Acomiada tots els

treballadors i aquests retornen com a treballadors disponibles de cada casa que estigues proporcionant treballadors a aquell mercat, per tant, augmenta el nombre de treballadors disponibles de la ciutat.

- **Carretera:** Connecta els edificis. Els edificis necessiten estar connectats entre ells per carreteres per funcionar correctament.

Interfície gràfica:

- Botons per a iniciar la construcció de cada tipus d'edifici i botó per a iniciar la destrucció.
- UI del jugador format per la visualització de recursos: diners, habitants sense feina, habitants totals de la ciutat i menjar.

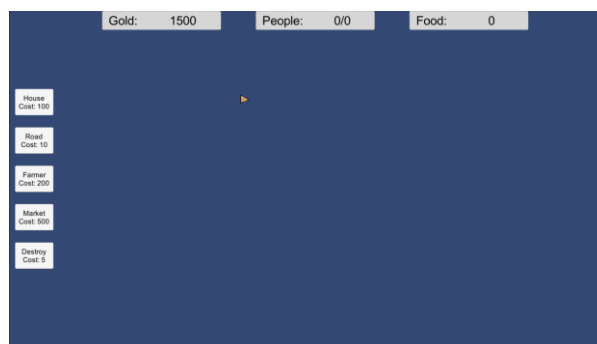


Fig. 8. UI ingame: Botons de construcció a l'esquerra i UI del jugador a dalt.

- **Menú inicial:** permet construir un nou mapa de la mida desitjada, introduir el nom del jugador, obrir el menú de puntuacions i sortir del joc.



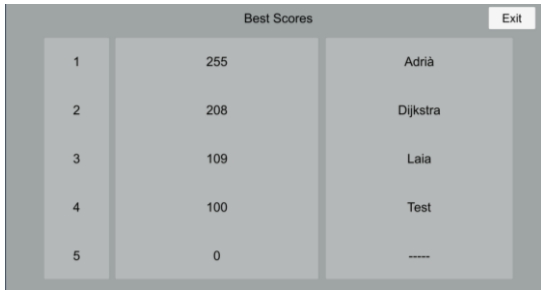
Fig. 9. Menú inicial.

- **Menú de pausa:** Pausa el joc. Permet retornar a la partida, obrir el menú de puntuacions, anar al menú inicial eliminant el mapa actual i sortir del joc.



Fig. 10. Menú de pausa

- **Menú de puntuacions:** Mostra l'històric de les 5 millors puntuacions i el nom dels jugadors, ordenat de major a menor puntuació. Permet tornar al menú anterior (menú de pausa o menú inicial).



Best Scores		
1	255	Adrià
2	208	Dijkstra
3	109	Laia
4	100	Test
5	0	

Fig. 11. Menú de puntuacions

La fórmula que utilitza el videojoc per a calcular la puntuació del jugador és la següent:

Habitants total + Menjar total a les cases + Or / 100

Estem tenint en compte el menjar que hi ha a les cases però no estem tenint en compte el menjar que hi ha als mercats o a les granges. Això és perquè fins que el menjar no arriba a les cases no compta com a menjar vàlid, ja que tant el mercat com la granja son part de la producció del menjar, per tant és menjar que encara està en producció.

L'or del jugador està dividit entre 100 per treure-li valor, ja que inicialment el jugador comença amb un nombre alt d'or i si l'or tingués molt valor a la puntuació, la millor puntuació es produiria quasi sempre en iniciar la partida, quan el jugador encara no ha construït res.

- **Panell d'informació d'edificis:** Quan el jugador posa el punter del ratolí sobre un edifici, apareix el panell d'informació d'aquell edifici. Aquest mostra el nombre de treballadors disponibles i màxims, el nombre de menjar actual i el temps d'espera de producció d'aquell edifici. Quan s'activa el panell es marquen tots els edificis com a semitransparents excepte l'edifici seleccionat i els últims edificis als quals aquest edifici ha venut menjar.



Fig. 12. Panell d'informació d'edificis

Altres:

- **Dijkstra:** Utilitzem l'algoritme de Dijkstra per a calcular el camí més curt entre dos edificis connectats per carreteres. Dos edificis no connecten entre si com a nodes veïns. Un edifici i una carretera o dues carreteres si connecten entre si com a nodes veïns.
- **Serialització:** Serialitzem i deserialitzem les 5 millors puntuacions i el nom de cada jugador. Les puntuacions estan ordenades de major a menor. Utilitzem la serialització binària de C# amb fitxers de tipus ".dat" per a serialitzar.

6.3 Test

Per a testear el codi hem utilitzat l'eina Test Runner de Unity. Aquesta eina ens permet testear únicament des de codi (Edit Mode) o utilitzant el codi i el motor gràfic de Unity (Play Mode). En aquest projecte s'ha utilitzat el Test Runner en Play Mode, ja que molts elements del projecte depenen del motor gràfic i per tant, volem testear-ho en el mateix context en el qual funciona el joc.

A la figura X es pot veure una imatge del Test Runner de Unity on mostra que tots els tests han finalitzat amb resultats correctes.



Fig. 13. Test Runner de Unity en el projecte.

A tots els tests que hem realitzat, abans de començar a testejar hem carregat l'escena principal del joc, perquè hi hagi tots els elements bàsics del joc instanciats.

S'ha realitzat test de caixa negra amb valors límits i frontera, per a totes les funcions implementades, excepte per a les següents:

- **Funcions Start, Awake, OnDestroy:** No s'ha testejat aquestes funcions perquè només les hem utilitzat per a inicialitzar/actualitzar variables o per a cridar individualment funcions ja testeja-des.
- **Funcions Update:** No s'ha testejat aquestes funcions, ja que les hem utilitzat per a cridar funcions ja testeja-des. Update és l'equivalent de Unity a una funció main().
- **Funcions que només criden a un conjunt de funcions:** No s'ha testejat aquestes funcions ja s'ha testejat individualment cada funció que la forma.

Per a testejar funcions que requerien prèviament el resultat d'altres funcions per a poder funcionar, s'han cridat aquestes funcions prèvies a l'inici del test, com a prerequisit o durant el test si és necessari.

La documentació dels tests es pot veure en el dossier del projecte.

6.4 Sprites

El projecte utilitza sprites ortogràfics 2D.

Els sprites dels edificis utilitzats en el videojoc formen part d'un conjunt d'sprites de lliure ús de la botiga de Unity anomenat "Simplex isomètric House pack" [23]. Els sprites de carretera i de Tiles els he dibuixat jo mateix, ja que no vaig trobar cap sprite ortogràfic de lliure ús que m'agradés o em semblés adequat.

6.5 Jugabilitat

En aquest apartat explicaré el cicle que hauria de seguir el jugador per tenir bons resultats:

Inicialment hauria de construir una granja, un mercat, algunes cases fins a tenir treballadors en les dues empreses i carreteres que connectin tots els edificis. Amb aquests elements el joc podrà començar a generar i vendre menjar.

A mesura que aconsegueixi or venent menjar a les cases, hauria de construir més cases i carreteres per a poder vendre més quantitats de menjar i guanyar més or. Quan la granja comenci a no produir prou menjar per totes les cases, hauria de construir més granges. El jugador ha d'anar alternant entre construir cases i granges.

Quan el jugador considera que té un nombre bastant alt

de cases i el mercat comenci a quedar allunyat de les cases, construirà un altre mercat lluny de l'antic mercat. A haver-hi ara dos mercats, les granges repartiran el menjar al mercat més proper, per tant és possible que el jugador hagi de construir granges properes al nou mercat.

Anirà repetint aquest cicle de construcció durant tot el joc.

7 RESULTATS

El resultat del projecte, tal com es va establir inicialment, és un prototip funcional del videojoc i la documentació necessària per a poder comprendre el desenvolupament, el disseny, el testeig i el funcionament intern del videojoc i del codi. Tot el codi desenvolupat està comentat per a millor comprensió.

S'ha desenvolupat una base sòlida del videojoc, a partir de la qual, si es vol continuar amb el desenvolupament del videojoc un cop acabat el treball, es poden anar afegint noves funcionalitats amb facilitat.

Actualment el projecte és només un prototip de videojoc, no és un videojoc complet que pugui sortir al mercat, ja que la seva jugabilitat és petita ara mateix, però això entra dins del que esperava del projecte. Això és perquè la base necessària per a poder començar a afegir funcionalitats de jugabilitat directa és gran.



Fig. 14. Imatge d'una partida

8 Conclusions

Com a conclusions avaluaré diferents aspectes:

Respecte a la planificació temporal del projecte, de forma general crec que ha sigut bastant encertada, ja que s'han aconseguit desenvolupar les funcionalitats en el temps esperat. Com a aspectes negatius de la planificació podem remarcar que es van produir canvis inesperats com el fet d'haver d'afegir noves iteracions que no estaven planificades inicialment (Dijkstra) o el fet d'haver de dividir una planificació en dues.

Respecte a la fase d'implementació considero que he

complet amb l'objectiu esperat. He desenvolupat les funcionalitats establertes inicialment i he desenvolupat els casos de tests per a cada iteració. M'agradaria remarcar la dificultat afegida de desenvolupar amb un llenguatge i IDE bastant desconegut per a mi, on he hagut d'aprendre com funciona per a poder desenvolupar correctament. Això ha provocat que en alguns moments hagi tardat més del que esperava a desenvolupar certes funcionalitats, ja que primer havia d'entendre aspectes concrets de Unity abans de poder continuar desenvolupant.

Respecte al disseny de classes i els requisits presos a l'inici del projecte, s'ha seguit l'estructura inicial en tot el desenvolupament del projecte però s'ha necessitat afegir algunes classes inesperades i modificar alguns elements sobre el camí. En aquest aspecte es podria concloure que el disseny inicial va ser bastant encertat tot i que ha necessitat millores a mesura que avançava el projecte.

Es pot concloure doncs que el projecte s'ha desviat una mica de la idea inicial, ja que s'han afegit i modificat elements pel camí, però no ha suposat un problema, ja que s'ha pogut complir amb la planificació, mantenir l'estructura del disseny inicial i desenvolupar els elements esperats.

9 Futures millores

Un cop desenvolupada la base del projecte, se li poden afegir noves funcionalitats amb certa facilitat. Això és perquè tots els edificis hereten la classe "Building" que conté totes les funcionalitats necessàries per a un edifici. Si es vol crear un nou edifici, el nou script només haurà d'heretar de "Building" i ja es pot afegir-li les seves funcionalitats específiques.

D'aquesta manera el projecte té un marge de creixement molt gran, tot depèn dels nous edificis que es volguessin afegir, que permetrien tenir una jugabilitat més diversa i entretinguda al videojoc.

El videojoc final requeriria implementar la serialització i deserialització de partides i els menús de càrrega i guardat corresponents. També requeriria la implementació de música i efectes de so.

AGRAÏMENTS

Voldria agrair a Francisco Javier Sánchez Pujadas, tutor d'aquest treball, per la seva motivació a l'hora de tutoritzar-me en el projecte, per les bones idees aportades i per la seva implicació en tot moment.

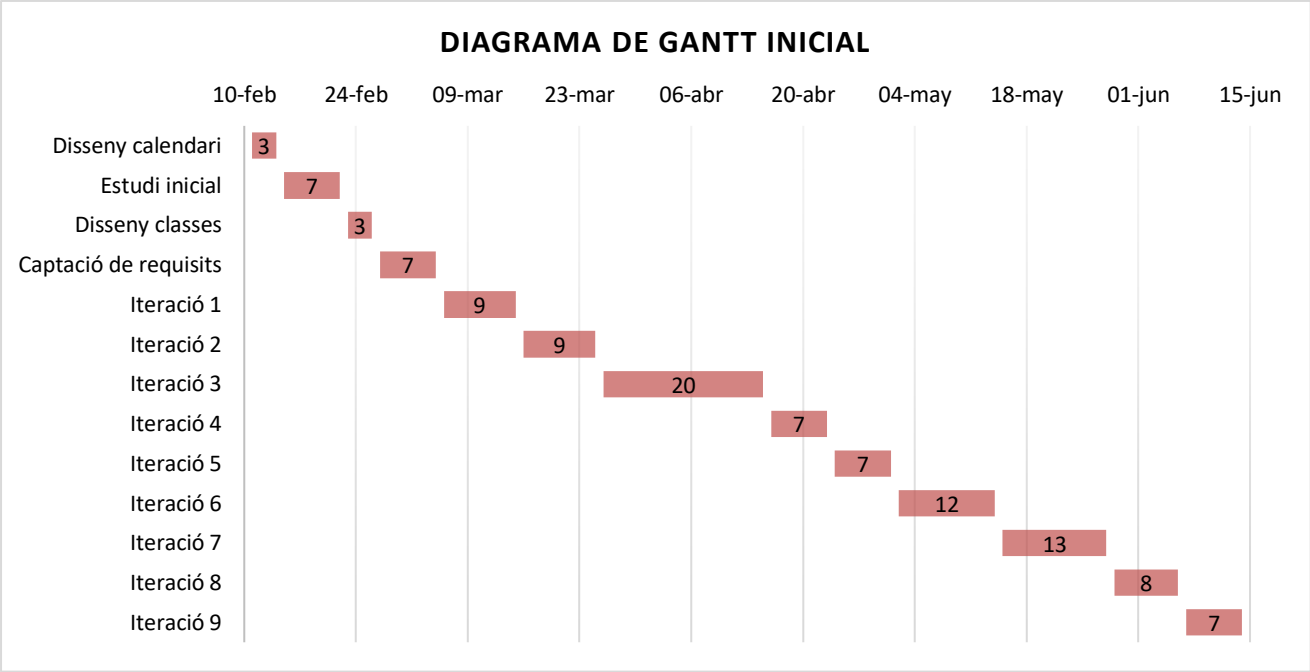
Voldria agrair també a tots aquells professors de la carrera, que a l'hora d'ensenyar comparteixen amb passió els seus coneixements amb els alumnes.

BIBLIOGRAFIA

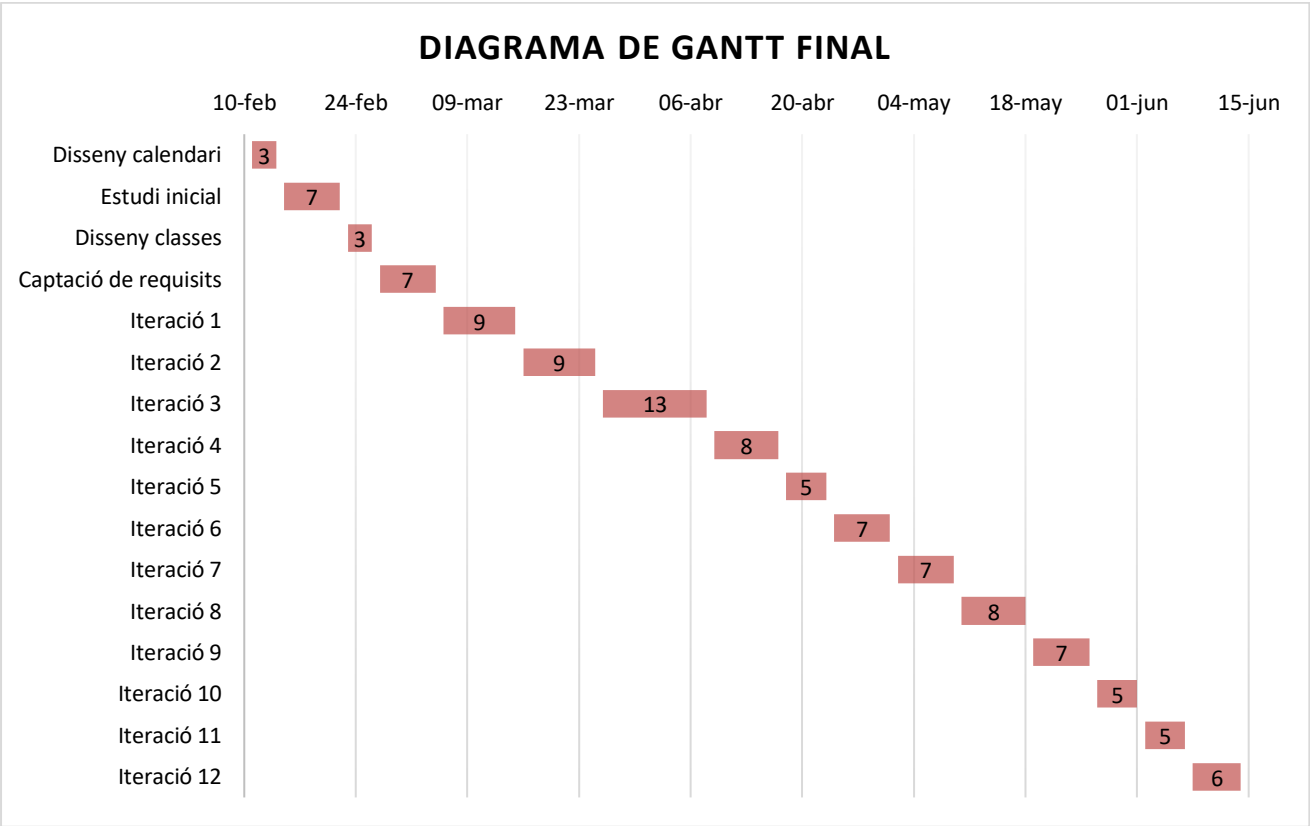
- [1] "City-building game". Wikipedia: https://en.wikipedia.org/wiki/City-building_game
- [2] Pàgina web de Unity: <https://unity.com/es/products/core-platform>
- [3] "Caesar III". Wikipedia: https://es.wikipedia.org/wiki/Caesar_III
- [4] "Shooter game". Wikipedia: https://en.wikipedia.org/wiki/Shooter_game
- [5] "Role-playing video game". Wikipedia: https://en.wikipedia.org/wiki/Role-playing_video_game
- [6] "Videojuego independiente". Wikipedia: https://es.wikipedia.org/wiki/Videojuego_independiente
- [7] Pàgina web de Unreal Engine: <https://www.unrealengine.com/en-US/>
- [8] "List of city-building games". Wikipedia: https://en.wikipedia.org/wiki/List_of_city-building_video_games
- [9] Pàgina web de Steam, cerca de videojocs City buildings: <https://store.steampowered.com/search/?tags=9,4328>
- [10] "Desarrollo ágil de software". Wikipedia: https://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_softw_are
- [11] "Kanban (desarrollo)". Wikipedia: [https://es.wikipedia.org/wiki/Kanban_\(desarrollo\)](https://es.wikipedia.org/wiki/Kanban_(desarrollo))
- [12] "Kanban: explicación para principiantes". <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban/>
- [13] Pàgina web de Visual Studio: <https://visualstudio.microsoft.com/es/>
- [14] "C Sharp". Wikipedia: https://es.wikipedia.org/wiki/C_Sharp
- [15] Pàgina web de Bitbucket: <https://bitbucket.org/>
- [16] Pàgina web de Trello: <https://trello.com/>
- [17] Pàgina web de Draw.io: <https://www.draw.io/>
- [18] "Algoritmo de Dijkstra" Wikipedia: https://es.wikipedia.org/wiki/Algoritmo_de_Dijkstra
- [19] "Problema del camino mas corto" Wikipedia: https://es.wikipedia.org/wiki/Problema_del_camino_m%C3%A1s_corto#Algoritmos
- [20] "Dijkstra algorithm Unity". Github: <https://github.com/N3evin/Dijkstra-Algorithm-Unity>
- [21] "Singleton". Wikipedia: <https://es.wikipedia.org/wiki/Singleton>
- [22] "State". Wikipedia: [https://es.wikipedia.org/wiki/State_\(patr%C3%B3n_de_dise%C3%B1o\)](https://es.wikipedia.org/wiki/State_(patr%C3%B3n_de_dise%C3%B1o))
- [23] "Simpleee Isometric House Pack". Asset Store de Unity: <https://assetstore.unity.com/packages/2d/environments/simplee-isometric-house-pack-76685>
- [24] "Mezcla y mastering online": <https://www.lucastoledo.me/>
- [25] Cerca de disenyadors gràfics a Malt.es: https://www.malt.es/s?q=dise%C3%B1ador+gr%C3%A1fico&fam=web_graphic_design&f-cat=graphic_designer

APÈNDIX

A1. DIAGRAMA DE GANTT: INICIAL



A2. DIAGRAMA DE GANTT: FINAL



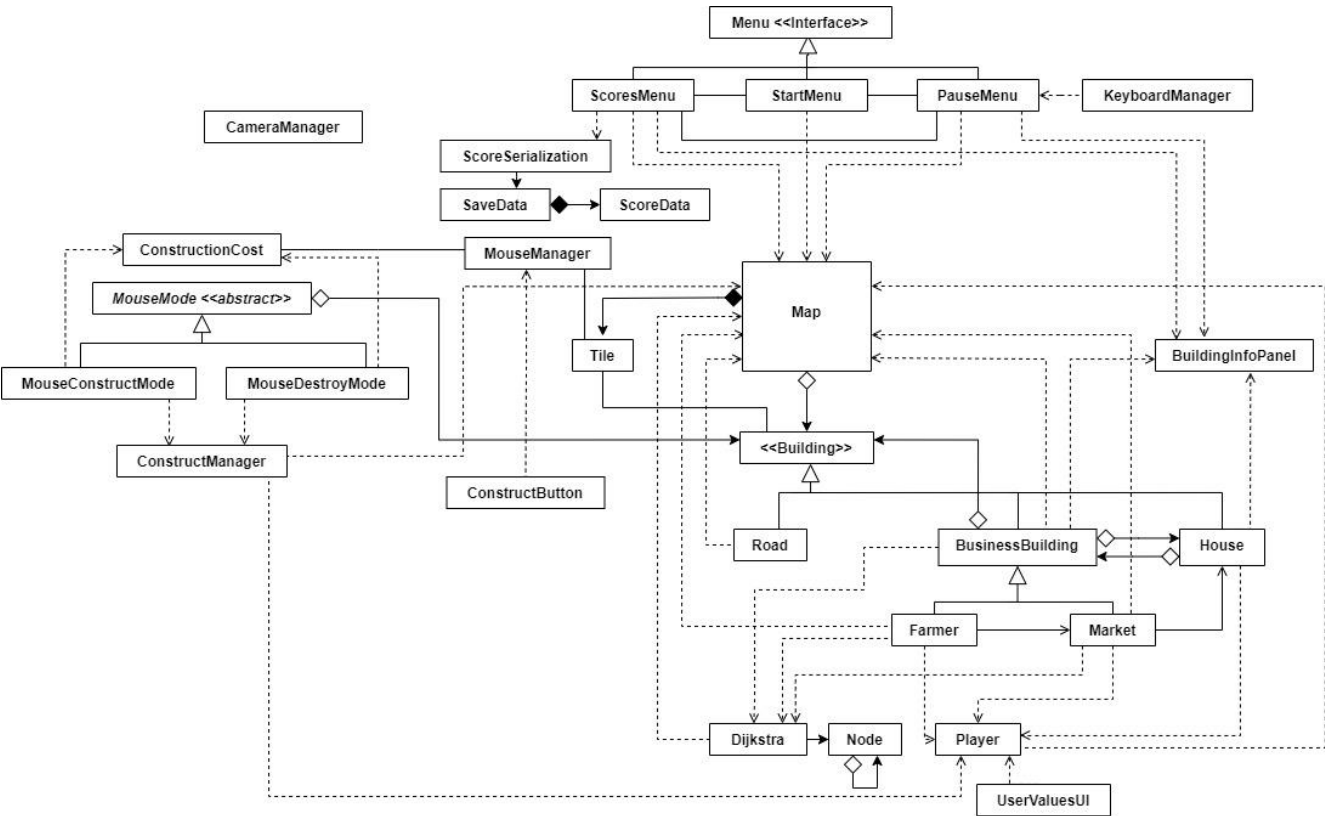
A3. ITERACIONS

- Disseny del calendari/planificació del projecte
- Estudi inicial del projecte
- Disseny de classes
- Captació de requisits.
- **Iteració 1:** Creació de mapa format per tiles
 - Tile
 - Map
- **Iteració 2:** Moviment de la càmera
 - Càmera
- **Iteració 3:** Preconstrucció d'un edifici
 - Building
 - Botó construcció
 - Ratolí (preconstruir)
 - Input Manager
 - Player (Monedes)
- **Iteració 4:** Construcció d'un primer edifici genèric
 - Ratolí (construir)
- **Iteració 5:** UI Jugador
 - UI Cost
 - Cost al construir
 - UI Habitants
 - UI Aliments
- **Iteració 6:** Construcció d'habitatges
 - House
 - UI d'habitants.
- **Iteració 7:** Construcció de carreteres
 - Road
- **Iteració 8:** Destrucció d'edificis
 - Botó destrucció
 - Ratolí (destruir)
- **Iteració 9:** Jugador, menús
 - Jugador
 - Menú
 - Menú de pausa
 - Menú inicial
 - Menú de puntuacions
- Serialització puntuacions
- **Iteració 10:** Dijkstra [18]
 - Dijkstra
 - Node
- **Iteració 11:** Construcció granger
 - Farmer
- **Iteració 12:** Construcció mercat
 - Market
 - UI de menjar
 - Panell informació edificis

A continuació es mostren les dates d'inici - final i els dies dedicats per a cada iteració:

Nom activitat	Data inici	Data final	Dies
Disseny calendari	11-feb	14-feb	3
Estudi inicial	15-feb	22-feb	7
Disseny classes	23-feb	26-feb	3
Captació de requisits	27-feb	05-mar	7
Iteració 1	06-mar	15-mar	9
Iteració 2	16-mar	25-mar	9
Iteració 3	26-mar	08-abr	13
Iteració 4	09-abr	17-abr	8
Iteració 5	18-abr	23-abr	5
Iteració 6	24-abr	01-may	7
Iteració 7	02-may	09-may	7
Iteració 8	10-may	18-may	8
Iteració 9	19-may	26-may	7
Iteració 10	27-may	01-jun	5
Iteració 11	02-jun	07-jun	5
Iteració 12	08-jun	14-jun	6

A4. DIAGRAMA DE CLASSES FINAL




```

Menu <<interface>>
+ MenuObject: GameObject
# previousMenu: Menu
+ OpenMenu(): void
+ CloseMenu(): void

```

