



Institut Puig Castellar

Santa Coloma de Gramenet



FRUIT QUEST

Projecte de desenvolupament

CFGM SISTEMES MICROINFORMÀTICS I XARXES

Steven Soriano i Álvaro Fernández

2smxb



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

B) GNU Free Documentation License (GNU FDL)

Copyright © 2025 Alvaro Fernandez i Steven Soriano.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

C) Copyright

© (Alvaro Fernandez i Steven Soriano)

Reservats tots els drets. Està prohibit la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor o dels límits que autoritzi la Llei de Propietat Intel·lectual.

Resum del projecte:

Fruit Quest és un videojoc 2D de plataformes en estil pixel art. El protagonista, un divertit personatge semblant a un gat amb armadura espacial, explora diferents nivells plens de fruites col·leccionables. L'objectiu principal del joc és recollir totes les fruites, evitar les trampes i superar els nivells amb dificultat progressiva.

El nostre objectiu es crear un joc accessible, entretingut i senzill d'entendre, però que requereixi habilitat, rapidesa i bons reflexos. Està pensat per a jugadors casuals i amants dels jocs retro (a partir de 8 anys), sense necessitat de tutorials complexos, amb mecàniques clàssiques de plataformes i reptes progressius que motivin a continuar jugant.

Hem seguit una metodologia àgil i progressiva. El desenvolupament s'ha realitzat utilitzant el motor Godot Engine i el llenguatge GDScript. També hem utilitzat GitHub per al control de versions i recursos externs per a sprites, sons i música. El desenvolupament s'ha dividit en fases iteratives: anàlisi de requisits, disseny de nivells i mecàniques (moviment, salts, col·lisions, recollida de fruites, puntuació), integració d'efectes sonors i música, creació de menús i HUD, proves contínues i correcció d'errors.

Hem aconseguit desenvolupar un videojoc funcional amb moviment fluid, tres nivells progressius, sistema de fruites, sistema de vida, interfície completa i efectes de so. També hem après a utilitzar Godot i GDScript, a organitzar un projecte complet i a treballar de manera col·laborativa. El resultat final és un joc senzill però complet que compleix els objectius plantejats.

Paraules clau:

Han de servir per a trobar el projecte si fem servir un cercador

- Fruit Quest
- videojoc 2D plataformes
- Godot engine
- GDScript
- Pixel art
- Plataformes

Abstract:

Fruit Quest is a 2D platformer video game with a pixel art style. The protagonist, a fun character similar to a cat with a space armor, explores different levels filled with collectible fruits. The main objective of the game is to collect all the fruits, avoid traps, and complete progressively more difficult levels.

Our goal is to create an accessible, entertaining, and easy-to-understand game, but one that still requires skill, speed, and good reflexes. It is designed for casual players and retro game lovers (from 8 years old), without the need for complex tutorials, using classic platforming mechanics and progressive challenges that encourage continued play.

We have followed an agile and progressive methodology. The development has been carried out using the Godot Engine and the GDScript programming language. We have also used GitHub for version control and external resources for sprites, sounds, and music. The development has been divided into iterative phases: requirements analysis, design of levels and mechanics (movement, jumping, collisions, fruit collection, scoring), integration of sound effects and music, creation of menus and HUD, continuous testing, and bug fixing.

We have successfully developed a functional video game with smooth movement, three progressive levels, a fruit collection system, a life system, a complete interface, and sound effects. We have also learned how to use Godot and GDScript, how to organize a complete project, and how to work collaboratively. The final result is a simple but complete game that fulfills the initial objectives.

Keywords (entre 4 i 8):

- Fruit Quest
- 2D platformer
- Godot engine
- GDScript
- Pixel art
- Platforms

Índex

<u>1</u>	<u>Introducció</u>	<u>12</u>
<u>1.1</u>	<u>Context</u>	<u>12</u>
<u>1.2</u>	<u>Justificació</u>	<u>12</u>
<u>1.3</u>	<u>Objectius</u>	<u>13</u>
<u>1.4</u>	<u>Estratègia i planificació del projecte</u>	<u>14</u>
<u>2</u>	<u>Descripció del projecte</u>	<u>19</u>
<u>2.1</u>	<u>Anàlisi de requisits</u>	<u>19</u>
<u>2.1</u>	<u>Previsió de tasques d'investigació</u>	<u>20</u>
<u>2.2</u>	<u>Tecnologies</u>	<u>22</u>
<u>2.3</u>	<u>Estructura del projecte</u>	<u>24</u>
<u>2.4</u>	<u>Descripció dels components</u>	<u>28</u>
<u>2.5</u>	<u>Definició de les tasques</u>	<u>32</u>
<u>2.5</u>	<u>Definició de les funcionalitats</u>	<u>35</u>
<u>3</u>	<u>Altres capítols</u>	<u>39</u>
<u>4</u>	<u>Conclusions</u>	<u>41</u>
<u>4.1</u>	<u>Conclusions generals del projecte</u>	<u>41</u>
<u>4.2</u>	<u>Consecució dels objectius</u>	<u>42</u>
<u>4.3</u>	<u>Valoració de la metodologia i planificació</u>	<u>43</u>
<u>4.4</u>	<u>Visió de futur</u>	<u>44</u>
<u>5.</u>	<u>Glossari</u>	<u>45</u>
<u>6.</u>	<u>Bibliografia</u>	<u>47</u>
<u>7</u>	<u>Annexos</u>	<u>50</u>

Llista de figures

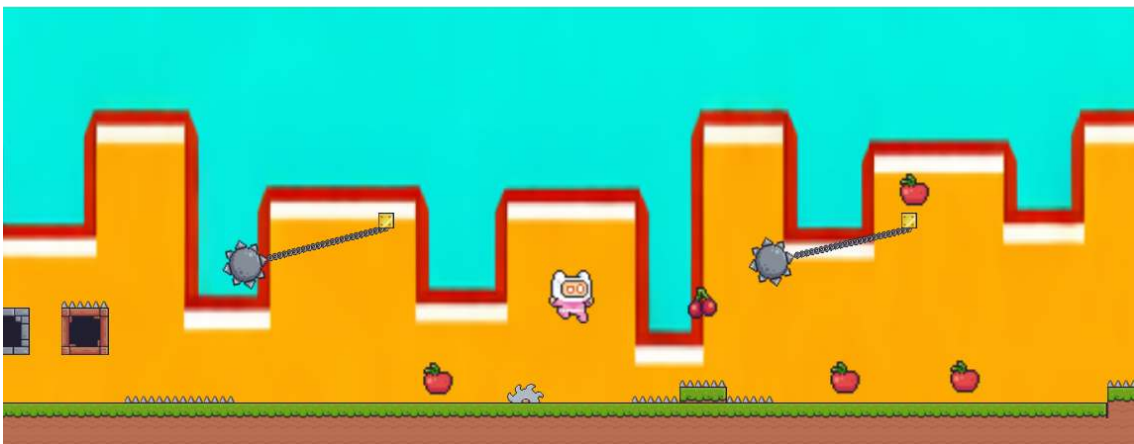
1.MENU PRINCIPAL



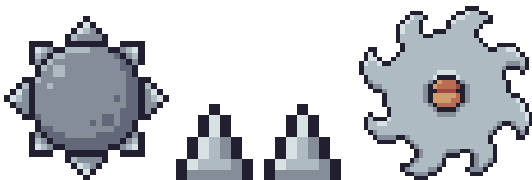
2.PERSONATGE



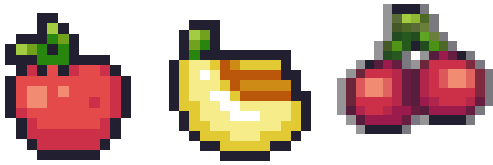
3. NIVELLS



4. TRAMPES



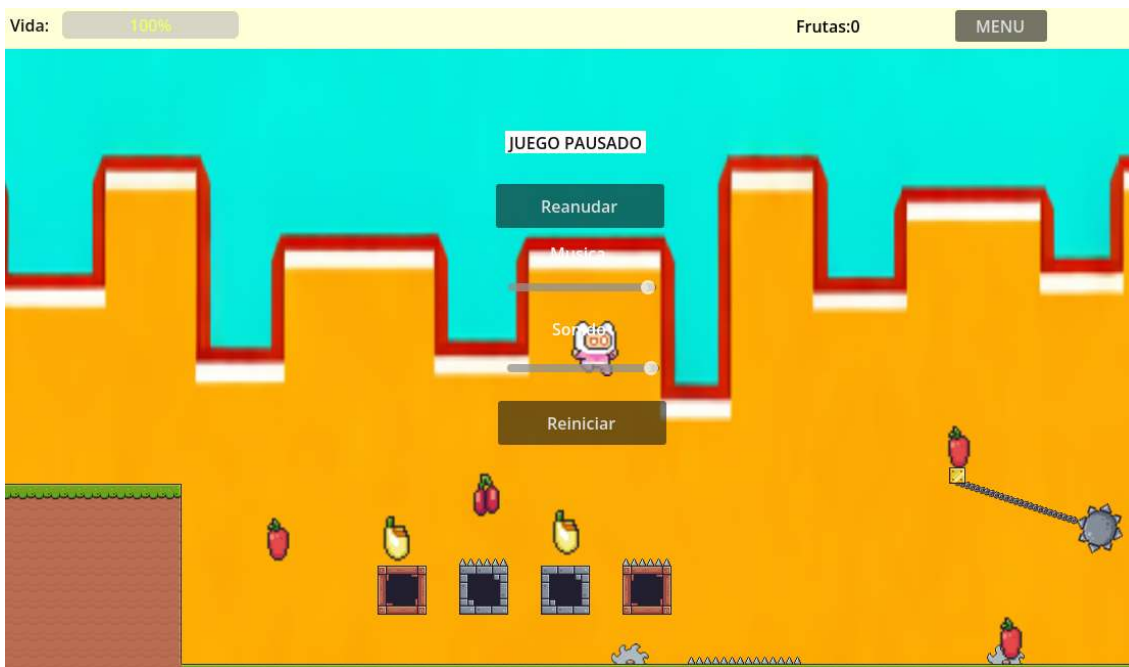
5. FRUITES



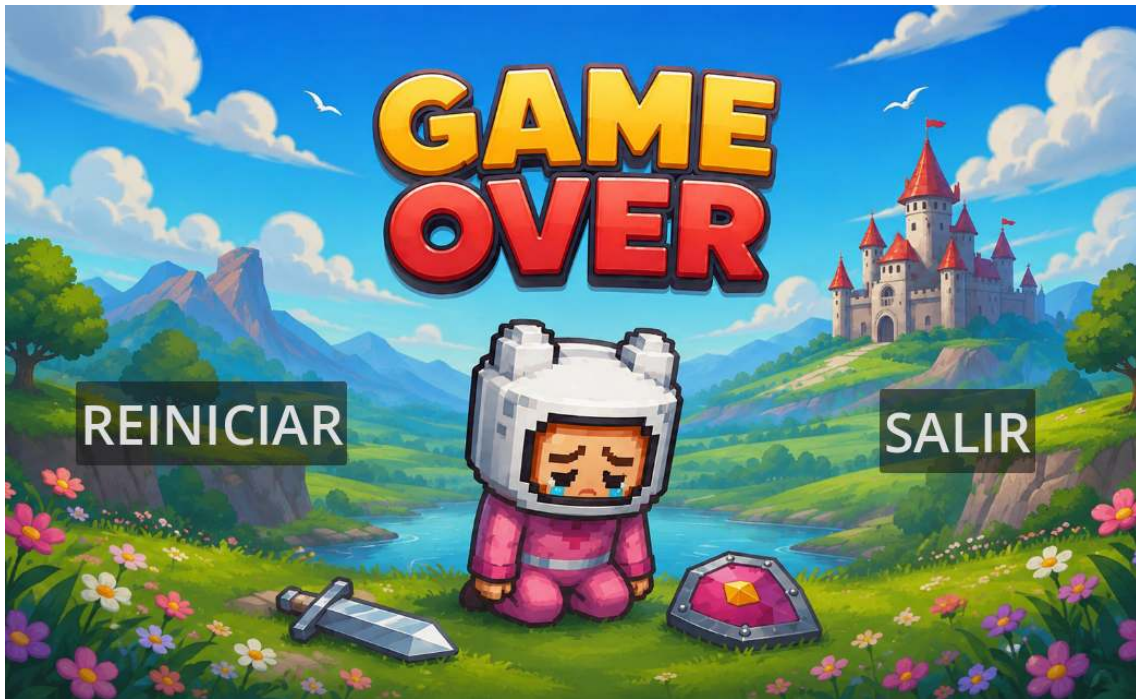
5. BARRA DE VIDA I COMPTADOR DE FRUITES



8. MENU PAUSA



9. PANTALLA GAME OVER



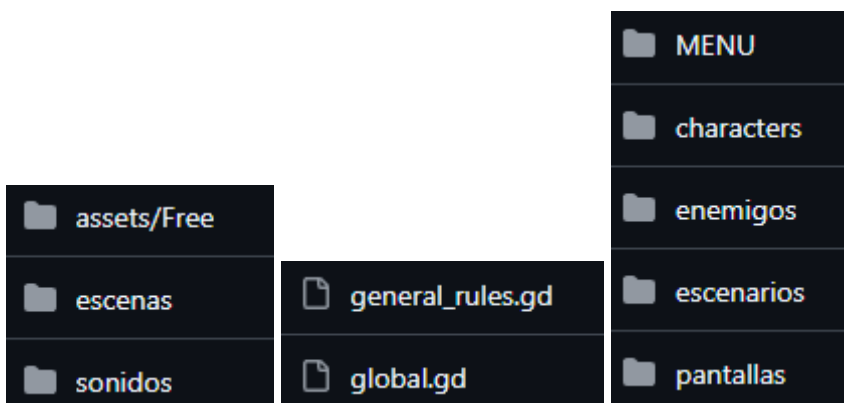
10. PANTALLA NIVELL COMPLETAT



11. PANTALLA VICTORIA



12. ESTRUCTURA DE CARPETES DEL PROYECTO EN GODOT



13. EXEMPLE SCRIPTS

```

func _physics_process(delta):
    if is_on_floor():
        leaved_floor = false
        had_jump = false
        count_jumps = 0

    if not is_on_floor():
        if not leaved_floor:
            $coyote_timer.start()
            leaved_floor = true
            velocity.y += gravity * delta

        if Input.is_action_just_pressed("ui_accept") and right_to_jump():
            if count_jumps == 1:
                allow_animation = false
                $animaciones.play("double_jump")
                $audioblesalto.play()
            else:
                $audio_jump.play()

            count_jumps += 1
            velocity.y = JUMP_VELOCITY

        var direction = Input.get_axis("ui_left", "ui_right")

    if direction:
        velocity.x = direction * SPEED
    else:
        velocity.x = move_toward(velocity.x, 0, SPEED)

    move_and_slide()
    
```

Aixo es una part del script principal, que gestiona la gravetat, el moviment lateral amb el tectat i el sistema dels tipus de salts, implementant el doble salt y el salt coyote. Finalment el script també activa les animacions i els efectes de so corresponents segons les accions del jugador.

```

112 func _on_damage_detector_area_shape_entered(area, self, _area, _area_shape_index, _local_body_index):
113     if not can_take_damage:
114         return
115
116     self.take_damage = false
117     $audio_damage.play()
118     $anim_ondamage.animation = false
119     velocity.y = -100
120     $animaciones.play("hit")
121
122     health -= 10
123
124     if health <= 0:
125         Global.current_scene = get_tree().current_scene.scene_file_path
126         $animaciones.play("gameover")
127         $asset_get_tree().create_timer(0.5).start()
128         self.take_damage = true
    
```

Aquesta part, controla el sistema de danys, quan el personatge col·lisiona amb una trampa, el sistema comprova si por rebre dany. Despres fa el efecte de so, l'animacio de l'impacte i li resten vida. Si la vida arriba a 0, automaticament es carrega la pantalla de game over. També apliquem la vulnerabilitat temporal per evitar rebre multiples danys seguits.

```

20 func _on_area_collect_body_entered(body):
21
22     if collected:
23         return
24
25     if not body.is_in_group("player"):
26         return
27
28     collected = true
29
30     Global.collected_fruits += 1
31     print("Frutas recogidas: ", Global.collected_fruits)
32
33     if body.has_method("collectFruit"):
34         body.collectFruit(fruitType)
35
36     Global.puntuacion = body.fruitCount
37     Global.frutas_recogidas = Global.collected_fruits
38     Global.current_scene = get_tree().current_scene.scene_file_path
39
40     $animaciones.play("collected")
    
```

En aquesta part del script de fruites, controla l'interacció entre el jugador i les fruites de cada nivell, quan el personatge toca una fruita seguidament la fruita desapareix amb una animació de recollecció i s'actualitza el recompte de les fruites, guardant l'estat actual de la partida.

1 Introducció

1.1 Context

Els videojocs son molt populars al dia de avui, i més si son jocs que de veritat sigui facil d'aprendre però que sigui entretingut i divertit de jugar. Els jocs actuals la majoria o són complicats de jugar o per poder continuar avançant, necesites pagar.

Aquest el nostre projecte neix per ser diferent, ja que es un joc senzill, gratuït, entretingut i divertit, on només necessites les teves habilitats i reflexos per passar els nivells. El joc està fet amb eines totalment gratuïtes per a que qualsevol dels usuaris pugui jugar sense cap inconvenient ni gastar-se diners.

1.2 Justificació

Ens va sortir l'idea de fer aquest projecte per unes simples raons:

- Aprendre a programar de veritat, encara que sigui molt basic.
- Treballar en equip i poder organitzar-nos com a professionals.
- Per crear alguna cosa divertida que pugui jugar la gent.
- Per poder demostrar les coses que hem après durant els dos cursos de grau mig.

A més d'aixó, pensem que el projecte respon a la necessitat en el mercat de jocs casuals, oferint sessions curtes, intenses pero divertides. Amb aquest projecte hem creat un producte complet utilitzant tecnologies lliures.

1.3 Objectius

Perquè aquest projecte sigui un èxit, hem d'assegurar-nos que anem per bon camí, per això ens hem d'aplicar unes metes que fins a que no les aconseguim no parem, ja que ens servirà de guia durant tot el desenvolupament i saber quan finalitzem cada part del joc.

1.3.1 Objectiu general

El nostre objectiu és crear un joc 2D que sigui divertit i entretingut, on el nostre personatge, pugui ser controlat a través del 3 nivells diferents creats per nosaltres, on haura de recollir fruites per poder sumar punts i evitar trampes mortals que resten vida. Amb l'ajuda de Godot i un programari basic, desde uns manuals vistos per nosaltres, assegurem que l'experiència del joc sigui fluida, divertida i visualment coherent.

1.3.2 Objectius específics

Per poder aconseguir l'objectiu general, hem de primer complir amb les següents fites concretes:

- **Moviment del personatge:** Programar que el gat espacial es pugui moure de forma fluida a la dreta i l'esquerra.
- **Salt normal i salts avançats:** No només implementar el salt normal basic, sinó també el doble salt i el salt coyote que deixa saltar altre vegada als instants de caire.
- **Creació de nivells:** Dissenyar i crear els nivells utilitzant assets, amb fons interactius i col·lisions configurades per nosaltres per a que el personatge no pugui atravesar aquests assets.
- **Recollida de les fruites:** Posarem diferents fruites per tot el mapa, per a que al tocar cada fruita puguin guanyar i acumular fruites.
- **Sistema de perills diferenciats:** Hi ha 2 tipus de perills, la sierra i la bola de pintxos, on si et toquen perds 10 de vida, però també estan les files de pintxos, on si caus en ells morts directament.

- **Interfície d'usuari:** Volem posar un menú d'inici amb la portada de fons i 2 botons per jugar i sortir, un menu de pausa per poder parar el joc en qualsevol moment, una barra de vida que mostra quan de vida li queda al jugador i al costat un recompte de punts que té cada fruita.
- **Transició entre nivells:** Amb el script, afegir que detecti quan agafa totes les fruites carregui automàticament el següent nivell.
- **Efectes de so:** Integrar efectes de so quan el personatge fa algun moviment, una música de fons, també en les diferents pantalles de **game over**, **nivel completado** i **victoria** amb un so corresponent.
- **Contador de fruites:** Fer un script que comti els punts les fruites que s'han recollit per saber si s'ha completat totes les fruites que és el que trata el joc.
- **Scripts individuals:** Cada element (personatge, trampa, fruita, enemic, pantalles) tindrà el seu propi script per gestionar la seva interacció entre tots.

1.4 Estratègia i planificació del projecte

L'estratègia seguida en aquest projecte ha consistit en desenvolupar un joc completament nou des de zero. Hem triat aquesta opció perquè ens dona un control total sobre tot el desenvolupament del videojoc, des de les mecàniques bàsiques fins als detalls més avançats. A més, ens permet entendre millor com funciona internament un motor de videojocs com Godot, així com millorar les nostres habilitats en programació amb GDScript i en disseny de videojocs.

Aquest enfocament també ens ha ajudat a afrontar problemes reals que poden aparèixer durant el desenvolupament, com errors en el codi, problemes de moviment o dificultats en la implementació de mecàniques. D'aquesta manera, no només hem creat un joc sinó que també hem après a analitzar, solucionar problemes i prendre decisions tècniques de manera autònoma.

La planificació del nostre projecte s'ha organitzat per fases per tal de treballar de forma ordenada:

Fase 1: Implementar el moviment del personatge, ja que és la base del joc. Això inclou el desplaçament, el salt, doble salt, salt coyote i les col·lisions amb les plataformes.

Fase 2: Desenvolupar tots els nivells del joc, definint la seva estructura, distribució de plataformes i dificultat progressiva.

Fase 3: Afegir elements com són les trampes (pinxes, serres, boles) que aporten dificultat i fan el joc més dinàmic.

Fase 4: Implementar els menús del joc, la barra de vida, el menú de pausa, de game over, de nivel completat i de victòria, el sistema de recollida de fruites i puntuació.

1.5 Metodologia de treball

Per dur a terme el desenvolupament del joc, hem decidit seguir una metodologia àgil i col·laborativa. Aquesta elecció respon la necessitat de crear un producte funcional pas a pas, permetent-nos provar cada part del joc abans de continuar amb la següent. A continuació, detallem l'organització del nostre treball, les eines utilitzades i la dinàmica del grup.

A diferència de les altres metodologies, nosaltres hem escollit una metodologia àgil i iterativa. Això és que hem dividit el nostre projecte en petites fases, en cadaescuna d'aquestes fases provavem una funcionalitat en concret, una vegada comprovada, pasavem a la següent tasca.

Amb aquesta metodologia, podíem arreglar al instant les coses que no terminaven de funcionar, sense tindre que esperar el final del joc. També, tenim flexibilitat durant el desenvolupament, així, poder incorporar millores sense tindre que desorganitzar tot lo avançat. Cada vegada que veíem el progressos constants y ben fets, ens va tenir bastants motivats.

Hem utilitzat diferents eines per poder gestionar el projecte d'una manera més comoda, que ens va permetre organitzar-nos millor i no perdre cap procés. Les eines digitals són les següents:

Github: Cada vegada que fèiem un canvi al codi, el "pujàvem" al repositori a través de la terminal. Això ens ha servit de còpia de seguretat i per evitar que el treball d'un esborri el de l'altre.

Trello: Hem fet servir un tauler amb columnes "Pendent, En Procés, Fet" per saber què tocava fer cada setmana i marcar les tasques com a completades.

Godot Engine: És on hem muntat el joc complet, amb les seves escenes, nivells, els scripts i la logica completa del joc.

Nosaltres, hem decidit que per treballar no és necessari dividir les tasques, sino, que és millor treballar conjuntament en tots els apartats.

Els avantatges que hem vist a l'hora de fer aquesta decisió, es que els dos tenim el mateix coneixement de les coses del joc, això és vital per a la defensa del projecte, ja que qualsevol de nosaltres pot respondre preguntes sobre qualsevol aspecte del joc.

Quan trobavem un error complicat, entre els dos sempre ha sigut més facil de resoldre'ls, ja que els dos podem pensar solucions diferents.

Cada setmana, feiem petites reunions abans de continuar amb el projecte, per revisar el que ja s'havia fet y planificar el que feriem en aquella sessió.

Vam seguir la estructura del cronograma establert al Document funcional, en diferents fases:

Octubre	Investigació i configuració	Instal·lació de Godot, i creació del repositori a GitHub.
Novembre - Desembre	Disseny i Prototip	Creació dels primers esbossos dels nivells i selecció dels gràfics
Gener - Febrer	Programació de Mecàniques	Implementació del moviment, salts i col·lisions.
Març	Implementació de nivells i assets	Construcció dels 3 nivells definitius, col·locació de fruits, trampes.
Abril	Poliment i Àudio	Afegir sons, música, menús i la barra de vida
Abril - Maig	Proves Documentació	Correcció d'errors finals, redacció d'aquesta memòria i preparació de la presentació.

Amb aquesta metodologia, també ens ha ajudat a prevenir diversos problemes, com la perduda de dades, ja que amb les pujades diàries al repositori de github sempre hem tingut el joc actualitzat fins l'última modificació.

També al dividir el projecte en l'aplicació trello, hem pogut veure la manca de temps de cada fase, veiem si estavem enrederits o teniem que ajustar el ritme.

1.6 Estudi econòmic i pressupostari

Fent aquest estudi, hem vist que gràcies a l'ús de les aplicacions amb totes les funcions lliures, no hem tingut que aportar cap diners del nostre compte, així que el cost econòmic d'aquest projecte es de 0,00€.

El cost que podem dir que ens ha costat ha sigut el temps i esforç invertit, que en un entorn professional això tindria un cost econòmic, segons la planificació del document funcional, les hores invertides valorades en cost economics serien les següents:

Període total	8 mesos (Octubre - Maig)
Octubre - abril	3 hores a la setmana = 78 hores
Abril - Maig	6 hores a la setmana = 48 hores
Total 1 persona	126 hores
Total l'equip de treball	252 hores

Si aquest projecte es fes dins d'una empresa, les hores de treball tindria un cost. Per a un desenvolupador principiant tindria un cost de 20€ l'hora, que en total amb les hores invertides tindria un cost de 5.040€.

Si després volguéssim millorar el joc després d'acabar aquest projecte, hem de tenir en compte els següents costos potencials:

- Adaptar el joc a noves versions del motor.
- Creació de nous nivells (cost de temps).
- Publicació a plataformes com Steam (cost de 100€ aprox).

2 Descripció del projecte

2.1 Anàlisi de requisits

2.1.1 Requisits funcionals

A continuació, estan els requisits funcionals del projecte que considerem que amb tot això, el projecte està finalitzat.

- RF1: El jugador pot moure el personatge a dreta i esquerra amb les fletxes del teclat.
- RF2: El jugador pot saltar, fer salt doble i aprofitar el salt coyote amb la tecla Espai.
- RF3: El menú principal permet iniciar el joc (carrega nivell 1) o sortir del joc.
- RF4: El joc té 3 nivells amb col·lisions configurades correctament en cada bloc.
- RF5: Hi ha fruites repartides pel mapa que sumen punts al tocar-les.
- RF6: Hi ha trampes (pinxes, serres, boles) que resten vida i eliminen el jugador a l'instant.
- RF7: El joc detecta la recol·lecció total de les fruites i carrega el següent automàticament.
- RF8: El joc té efectes de so per a salts, danys, recollida i música de fons en diferents pantalles.
- RF9: El joc mostra una barra de vida visible que s'actualitza en temps real.
- RF10: Al costat de la barra de vida estara el recompte de fruites.
- RF11: El jugador pot pausar el joc en qualsevol moment amb la tecla ESC.
- RF12: Cada element (personatge, trampa, fruita, pantalla) té el seu propi script per gestionar la interacció.

2.1.2 Requisits no funcionals

- RNF1: El joc ha de funcionar amb fluïdesa, sense tirades i en ordinadors normals.
- RNF2: Els controls han de respondre ràpid sense retard.
- RNF3: L'estètica ha de ser coherent gràcies als assets utilitzats.
- RNF4: El codi ha d'estar ordenat per poder-ho modificar en un futur, si és necessari.
- RNF5: El joc ha de ser jugable només amb teclat (fletxes, espai, etc).

2.1 Previsió de tasques d'investigació

Abans de començar el desenvolupament del videojoc, ha sigut molt important i necessari realitzar tasques d'investigació amb l'objectiu d'entendre les eines que anaven a utilitzar com poden ser les tecnologies i mecàniques que volíem afegir al projecte. Aquestes tasques d'investigació ens han ajudat a tenir una bona base del projecte i han influït en les decisions que hem tingut durant el procés del treball.

Principalment hem investigat el funcionament del motor de videojocs Godot. Aquesta tasca ha consistit en comprendre l'estructura basada en escenes i nodes, així com el funcionament en entorns 2D i la gestió de col·lisions. Aquest coneixement ha estat essencial per poder començar a construir el joc des de zero.

En segon lloc, hem pogut analitzar l'ús del llenguatge de programació GDScript que està dins de Godot. Hem pogut aprendre com vincular scripts als objectes del joc i com gestionar la lògica, com ara el moviment del personatge, el control de variables, etc. Aquesta investigació ha estat molt important per desenvolupar correctament el comportament del joc.

Una altra tasca important ha estat la investigació dels sistemes de moviment típics en jocs de plataformes. Hem analitzat com implementar el moviment lateral, el salt, el doble salt i el salt coyote. Aquest últim proporciona un petit marge de temps per saltar després d'haver abandonat una plataforma, millorant la jugabilitat i fent-la més emocionant per al jugador.

També hem treballat en el sistema de col·lisions. Ha sigut necessari determinar com evitar que el personatge atravesi elements, com detectar interaccions amb objectes com les fruites i com gestionar el contacte amb les trampes. Aquesta part ha sigut molt important per tenir un bon funcionament del joc.

El disseny de nivells, s'ha investigat com crear escenaris amb una dificultat cada cop més elevada. Hem tingut que anar analitzant diferents aspectes com la distribució de plataformes, la col·locació de trampes i la ubicació de les fruites. L'objectiu ha estat aconseguir nivells amb un nivell equilibrat i que el jugador pugui divertir-se jugant al joc.

Seguidament també hem après l'ús d'assets externs, com poden ser sons descarregats en format de carpeta comprimida. Hem après a importar aquests sons al projecte de GODOT, sense cap problema i disfrutar d'aquests efectes.

Una altra cosa que hem investigat i hem implementat al joc, ha estat la gestió del progrés entre nivells. S'ha analitzat com detectar quan el personatge menja totes les fruites i carregar automàticament el següent nivell, després de la pantalla de "nivel completado".

Finalment, s'ha implementat la creació d'usuari i la integració d'efectes d'àudio i so. En aquest punt, s'ha estudiat com afegir efectes de so i música de fons que responguin a les accions del jugador.

Hem arribat a la conclusió, que en conjunt totes aquestes tasques d'investigació han estat molt importants per al desenvolupament del projecte, ja que ens han permès comprendre millor els elements necessaris per crear el videojoc i ens ha ajudat a poder orientar-nos millor.

2.2 Tecnologies

2.2.1 Comparativa de les tecnologies valorades

Abans de començar el projecte hem comparat diferents tecnologies per decidir quines ens convenien més. La idea era escollir eines que siguin fàcils d'utilitzar, però que també ens permeten fer el joc sense problemes.

Primer hem comparat dos motors de videojocs: Unity i Godot. Unity és molt potent i molt utilitzat, però també és més complicat i pesat, sobretot per fer un joc 2D senzill com el nostre. En canvi Godot és menys pesat que Unity, més fàcil d'aprendre i està molt ben preparat per fer jocs en 2D. Per això ens ha semblat millor opció per aquest projecte.

Després hem comparat dos llenguatges: GDScript i C#. GDScript és més fàcil perquè està fet per Godot, però ja que el joc es realitza en godot hem aprofitat i utilitzat GDScript.

També hem mirat altres eines que ens podien ajudar. Per guardar el projecte i treballar en equip, hem utilitzat GitHub, que ens permet no perdre el codi i tenir-lo sempre guardat.

En resum, hem triat Godot i GDScript perquè són una bona combinació: fàcils d'utilitzar, però al mateix temps útils per aprendre i fer un projecte complet.

2.2.2 Tecnologies escollides

Després de comparar les diferents opcions disponibles, vam decidir utilitzar tecnologies que fossin fàcils d'aprendre i adequades per al desenvolupament d'un videojoc 2D complet i funcional. L'objectiu principal era treballar amb eines senzilles però potents, que ens permetessin desenvolupar el projecte de manera ordenada i eficient.

El motor gràfic escollit ha estat Godot Engine, ja que és gratuït, lleuger i molt recomanable per a videojocs 2D. A més, el seu sistema basat en nodes i escenes facilita molt l'organització del projecte i la creació de mecàniques interactives.

Pel que fa al llenguatge de programació, hem utilitzat GDScript, que és el llenguatge propi de Godot. Hem escollit aquesta opció perquè està totalment integrada dins del motor gràfic, és fàcil d'entendre i té una sintaxi semblant a Python. Gràcies a això hem pogut programar els moviments del personatge, les trampes, els menús, els sons i totes les funcionalitats del videojoc de manera més senzilla.

Tot el desenvolupament del codi s'ha realitzat directament des de l'editor integrat de Godot, sense necessitat d'utilitzar programes externs. Això ens ha permès treballar de forma més còmoda i tenir tots els scripts i escenes organitzats dins del mateix projecte.

Finalment, per al control de versions i el treball en equip hem utilitzat GitHub, que ens ha servit per guardar còpies de seguretat del projecte, controlar els canvis realitzats i compartir el treball entre els integrants del grup.

En resum, les tecnologies principals utilitzades han estat Godot Engine, GDScript i GitHub, ja que ens han permès desenvolupar el videojoc de manera estable, organitzada i sense grans complicacions.



2.3 Estructura del projecte

El projecte està organitzat mitjançant diferents carpetes i escenes dins de Godot Engine. Cada element del videojoc disposa de la seva pròpia escena i el seu propi script, cosa que facilita molt l'organització i el desenvolupament del joc.

L'estructura principal del projecte està dividida en diferents carpetes segons la seva funcionalitat i relació.

Carpeta “Menu”:

Aquesta carpeta conté totes les escenes, scripts i imatges relacionades amb els diferents menús del joc.

Imatges dels menús

- fruitquest.png → Fons del menú principal.
- gameover.png → Fons de la pantalla Game Over.
- nivelcompletado.png → Fons de nivell completat.
- victoria.png → Fons de la pantalla de victòria.

Escenes dels menús

- menu.tscn → Escena del menú principal.
- nivelcompletado.tscn → Escena quan el jugador completa un nivell.
- gameover.tscn → Escena de Game Over.
- victoria.tscn → Escena final de victòria.

Scripts dels menús

- menu.gd → Control del menú principal.
- nivelcompletado.gd → Control de la pantalla de nivell completat.
- gameover.gd → Control de la pantalla de derrota.
- victoria.gd → Control de la pantalla de victòria.

Carpeta “Characters”:

Aquesta carpeta conté el personatge principal del videojoc.

Fitxers

- ninjacat.tscn → Escena principal del personatge.
- ninjacat.gd → Script amb tota la lògica del jugador.

Aquest script controla:

- Moviment.
- Salt.
- Doble salt.
- Coyote jump.
- Animacions.
- Vida del jugador.

Carpeta “Escenarios”:

Aquesta carpeta conté els mapes, els objectes interactius i els fons del videojoc.

Fons dels mapes

- fonomapa1.png
- fonomapa2.png

Aquestes imatges s'utilitzen com a background dels nivells.

Fruites

- fruit.tscn → Escena de les fruites.
- fruit.gd → Script de recollida de fruites.

Mapes

- mapa1.tscn
- mapa2.tscn
- mapa3.tscn

Aquests fitxers contenen l'estructura principal de cada mapa.

Carpeta “Pantallas”:

Aquesta carpeta conté les escenes principals dels nivells i els seus scripts.

Escenes

- pantalla1.tscn → Nivell 1.
- pantalla2.tscn → Nivell 2.
- pantalla3.tscn → Nivell 3.

Scripts

- pantalla1.gd
- pantalla2.gd
- pantalla3.gd

Aquests scripts controlen:

- El funcionament general del nivell.
- El canvi entre pantalles.
- La detecció de victòria.
- La interacció amb els objectes.

Interfície del jugador

Fitxers

- `playerinfo.tscn` → Escena de la interfície.
- `playerinfo.gd` → Script de la barra de vida i recompte de fruites.

Aquest sistema mostra durant tota la partida:

- Vida restant.
- Fruitess recollides.
- Informació del jugador.

Scripts addicionals

`següentnivell.gd`

Script encarregat de carregar automàticament el següent nivell quan el jugador completa l'actual.

`audio_fondo.gd`

Script encarregat de controlar la música de fons del joc.

`global.gd`

Script global carregat automàticament durant tota la partida. S'utilitza per guardar informació general del joc.

`general_rules.gd`

Script amb regles generals compartides entre diferents parts del videojoc.

Carpeta “Sonidos”

Aquesta carpeta conté tots els efectes de so i música utilitzats al videojoc.

Exemples:

- So de salt.
- So de fruites.
- Música de fons.
- Sons dels menús.
- Sons de derrota i victòria.

2.4 Descripció dels components

2.4.1 PLAYER (gat)

És el cor del joc i té un script que controla la física.

- Moviment: Es mou amb les fletxes del teclat.
- Salt Normal: Salta quan pressionem la tecla espai estant a terra.
- Salt Doble: Pot tornar a saltar mentre està a l'aire amb el doble espai.
- Salt Coyote: Si cau d'una plataforma, té un segon extra per poder saltar encara que ja no estigui al terra.
- Vida: Té una variable de vida que es connecta amb la barra de vida de la interfície i va perdent vida quan es choca amb alguna trampa o mort quan toca els pintxos.

2.4.2 ELS NIVELLS

Hem creat 3 mapes diferents.

- Disseny: Hem fet servir els assets d'una carpeta zip per construir les plataformes i els blocs dels mapes ràpidament.
- Col·lisions: Cada bloc té configurat la seva col·lisió personalitzada.
- Fons: Hem posat imatges de fons dels nivells “**backgrounds**” per donar més bona visibilitat al jugar i que no sigui tot negre.

2.4.3 Trampes

Les trampes són obstacles distribuïts pels diferents nivells amb l'objectiu d'augmentar la dificultat del videojoc i posar a prova els reflexos del jugador.

Al nostre joc hi ha tres tipus principals de trampes:

- **Serres en moviment:** es desplacen horitzontalment pel mapa i, quan toquen el jugador, li resten 10 punts de vida.
- **Bola de punxes:** disposa d'un moviment tipus pèndol amb cadenes. Quan col·lisiona amb el jugador també li resta 10 punts de vida.
- **Files de pintxos:** són trampes estàtiques col·locades principalment al terra. Si el jugador cau sobre elles, mor instantàniament i ha de reiniciar el nivell.

Cada trampa disposa del seu propi script per controlar el moviment, les col·lisions i el dany aplicat al jugador.

2.4.4 INTERFÍCIE

- Menú Principal:

El joc disposa d'un menú principal amb un fons personalitzat utilitzant la portada del videojoc. Aquest menú conté dos botons principals:

- **Jugar:** carrega directament el primer nivell.
- **Sortir:** tanca el videojoc.

- Barra de Vida:

Durant la partida el jugador pot visualitzar una barra de vida que mostra la salut restant del personatge.

Quan el jugador col·lisiona amb trampes com les serres o el spikeball, la vida disminueix progressivament. Si la vida arriba a zero, apareix la pantalla de Game Over.

- Comptador de fruites:

El joc també disposa d'un comptador de fruites visible durant tota la partida. Aquest sistema mostra la quantitat de fruites recollides ja que l'objectiu principal del videojoc és aconseguir totes les fruites per poder avançar al següent nivell.

- Menú de pausa:

Hem implementat un menú de pausa que permet al jugador:

- Reprendre la partida.
- Reiniciar el nivell actual.
- Controlar el volum de la música i dels efectes de so.

Aquest menú es pot obrir durant la partida.

- Pantalla de nivell completat:

Quan el jugador aconsegueix totes les fruites d'un nivell, apareix una pantalla de "Nivell Completat".

Aquesta pantalla mostra:

- El nivell superat.
- La puntuació de fruites.
- Botó per continuar.
- Botó per sortir del joc.

- Pantalla Game Over:

Si el jugador perd tota la vida o mor instantàniament amb les punxes, apareix la pantalla de Game Over.

Aquesta pantalla permet:

- Reiniciar el nivell.
- Sortir del joc.

- Pantalla de victòria:

Quan el jugador supera el nivell final, apareix una pantalla de victòria.

Aquesta pantalla dona l'opció de:

- Reiniciar el joc des del nivell 1.
- Sortir del joc.

2.5 Definició de les tasques

2.5.1 Sistema de Moviment Avançat

Hem programat un script específic per poder gestionar tots els salts que tenim configurats. No hem posat que salta, sinó que comprova si ja ha fet un salt per permetre i que pugui fer el doble salt i també tenim com un temporitzador intern per al salt coyote, ja que es un salt que dona 1 segon extra després de caure de la plataforma.

- Lògica Coyote: Quan el jugador deixa de tocar el terra, s'inicia un comptador d'un segon extra. Si el jugador prem salt mentre el comptador està actiu el salt es permet, però si ja ha passat temps de més, el jugador caurà cap a baix.

2.5.2 Gestió de Nivells i Transicions

El sistema de progressió del joc està basat en la recollida de fruites. Cada nivell conté diferents fruites distribuïdes pel mapa i el jugador les ha de recollir totes per poder avançar.

Hem implementat un script que controla el nombre de fruites recollides durant la partida. Quan el jugador aconsegueix recollir totes les fruites del nivell, el sistema detecta automàticament que el nivell està completat i carrega la pantalla de "Nivell Completat".

Des d'aquesta pantalla, el jugador pot continuar cap al següent nivell o sortir del joc.

En completar el nivell final, apareix directament la pantalla de victòria final.

2.5.3 Sistema de Danys i Vida

Per fer el joc més divertit i amb una mica més dificultat el que hem fet ha sigut crear un sistema de vida i danys. D'aquesta manera el jugador no únicament ha de saltar i recollir fruites, sinó que també ha d'evitar trampes durant els nivells.

Hem diferenciat dos tipus de danys:

1. Les trampes com la serra i la bola de pintxos, aquestos resten 10% de vida al jugador quan hi col·lisiona. Aquests danys es veuen reflectats en la barra de vida quan recibeix el dany.
2. Les files de pintxos provoquen una eliminació immediata del jugador, auqn el jugador cau sobre aquestes trampes, apareix directament la pantalla de "game over" amb dues opcions, sortir del joc o reiniciar el nivell.

2.5.4 Scripts individuals

Una de les decisions més importants durant la creació del joc ha sigut separar el codi de script en diferents scripts segons la funció que tinguin cadascun. En lloc de posar tot el codi dins d'un únic script, hem dividit cada funcionalitat en diferents fitxers .gd. Això ens ha ajudat a mantenir el projecte més net, ordenat i fàcil de modificar.

El script principal que tenim es ninjacat.gd que controla tot el comportament del personatge principal del nostre joc. Aquest script gestiona el moviment, els salts, el doble salt, el salt coyote i també el sistema de vida del jugador. Es el script més important del joc, ja que controla gran part de les mecàniques principals del videojoc.

Després tenim el script de les Fruites, que es l'encarregat de les fruites del joc. Quan el jugador toca una fruita, aquest script detecta la col·lisió i suma els punts especialitzats de cada fruita al marcador i fa desaparèixer la fruita del mapa, també aquest script es important perque es necessita recollir totes les fruites per pasar el nivell. També pot reproduir un efecte de so perquè el jugador sàpiga que l'ha recollit.

Després tenim els scripts de trampes. La seva funció és detectar quan el jugador toca una zona en la cual hi hagi una trampa com per exemple la fila de pintxos que et treu tota la vida de cop, una bola de pintxos o una serra que t'anirà treient 10% de vida.

Continuem amb els scripts de las pantallas (nivells), que ocnroles el funcionament general de cada nivell. Gestiones el recompte de fruites, el canvi de nivell, les transicions i la carrega de les pantalles finals.

Cada menu, també té el seu propi script, que controlen els botons de reinici i sortir, el canvi al següent nivell i en la pantalla final controla el reinici del joc.

Tenim uns scripts globals (`general_rules.gd`, `global.gd`), que s'executen automàticament en tot el joc i permeten compartir informació entre les diferents parts del joc.

En l'interfície del jugador, també té el seu script (`playerinfo.gd`), que permet controlar la barra de vida, el recompte de fruites, el progrés del nivell i també controlar el volum dels diferents audios, el de la música de fons i el de els moviments del personatge.

Treballar amb scripts separats ens ha ajudat molt durant el projecte, perquè si alguna cosa fallava només havíem de revisar el script concret d'aquell objecte i no tot el codi del joc.

2.5.5 Menús i audio

Per fer el joc més complet també hem implementat diferents menús i efectes de so. Això fa que l'experiència sigui més agradable i dona una sensació més professional al projecte.

El menú principal és la primera pantalla que veu el jugador quan obre el joc. Té un fons amb la portada del joc i dos botons principals: “**Jugar**” i “**Sortir**”. El botó de jugar carrega directament el primer nivell, mentre que el de sortir tanca el joc. Aquests botons funcionen gràcies a la configuració que li hem donat, ja que detecten quan el jugador fa clic amb el ratolí al botó.

També hem creat un menú de pausa que surt a la part superior dreta de la pantalla. Quan el jugador prem la tecla ESC, el joc es pausa i apareix el menú de pausa a la pantalla, també es pot fer clic en el botó “MENU”. Això ho hem fet utilitzant la funció `get_tree().paused = true`, que pausa tot el joc menys la interfície. D'aquesta manera, el jugador pot parar la partida quan vulgui, baixar la música o el audio o pujar-ho i tornar a reanudar el joc.

En la part de l'audio, hem afegit diferents efectes de so per donar més vida al joc i que sigui més atractiu. Per exemple, hi ha sons quan el jugador salta, quan recull una fruita o rep dany. Aquests sons s'activen amb codi utilitzant funcions com `AudioPlayer.play()`. També hem afegit música de fons als nivells perquè el joc no se senti buit o silenciós mentre jugues.

En general, els menús i els efectes de so ajuden molt a millorar la sensació del joc i fan que l'experiència del jugador sigui molt més divertida i completa.

2.5 Definició de les funcionalitats

2.5.1 Sistema de moviment del personatge

Una de les funcionalitats més importants del joc és el control del personatge. El jugador pot moure al personatge cap a l'esquerra i la dreta utilitzant el teclat. El moviment s'ha programat perquè sigui fluid i ràpid, donant una sensació còmoda mentre es juga.

A més del moviment bàsic, també s'ha implementat un sistema de salts avançat. El personatge pot fer un salt normal, un doble salt mentre està a l'aire i el conegut "salt coyote". Aquesta mecànica dona al jugador un petit marge de temps per tornar a saltar just després de caure d'una plataforma, fent que el joc sigui més just i menys frustrant.

Aquest sistema funciona gràcies a scripts en GDscript que controlen la física del personatge i detecten quan està tocant el terra o quan es troba a l'aire.

Estat de la funcionalitat: Implementada completament.

2.5.2 Sistema de nivells

El joc està format per 3 nivells diferents amb dificultat progressiva. Cada nivell té plataformes, trampes i fruites distribuïdes de manera diferent per augmentar la dificultat a mesura que el jugador avança.

Els nivells s'han construït utilitzant tilesets/tilemaps i assets externs importats a Godot. També s'han afegit backgrounds diferents per donar més ambientació al nivell i profunditat visual a cada mapa.

Quan el jugador agafa totes les fruites, el joc detecta que el recompte està completat i carrega automàticament el següent nivell. Això permet una transició ràpida i fluida entre escenes.

Estat de la funcionalitat: Implementada completament.

2.5.3 Sistema de col·lisions

El sistema de col·lisions és el que permet que el joc funcioni correctament físicament. S'ha configurat perquè el personatge no pugui travessar parets, plataformes o objectes sòlids del mapa.

També s'utilitza per detectar quan el jugador toca **fruites**, **pintxos**, **serres** o les **boles de pintxos**. Cada objecte té una col·lisió configurada segons la seva funció dins del joc.

Aquest sistema ha estat molt important durant el desenvolupament, ja que qualsevol error en les col·lisions podia afectar directament la jugabilitat.

Estat de la funcionalitat: Implementada completament.

2.5.4 Sistema de fruites i puntuació

Durant els nivells, el jugador pot recollir fruites repartides pel mapa. Quan el personatge toca una fruita, aquesta desapareix i s'afegeixen punts al marcador. La puntuació es diferent dependent de la fruita que hi agafem:

Pomes=1punt **Bananas**=2Punts **Cireres**=3punts

Aquest procés es controla amb un script que hem creat, el qual detecta la col·lisió entre el jugador i la fruita. A més, també es reproduïx un efecte de so per donar més experiència visual i sonora al jugador.

El sistema permet comptar quantes fruites s'han recollit durant la partida i mostrar aquesta informació a la pantalla. I el jugador necessita recollir totes les fruites per poder completar el nivell, ja que si et deixes una sola fruita, no podràs passar .

Estat de la funcionalitat: Implementada completament.

2.5.5 Sistema de danys i vida

Per augmentar la dificultat del joc, s'ha creat un sistema de vida i danys per trampes. El jugador pot perdre vida quan toca la bola de pintxos/serres o morir directament si cau sobre una fila de pintxos.

Hem diferenciat dos tipus de danys:

- Les files de pintxos provoquen mort instantània i reinicien el nivell.
- Las boles de pintxos i les serres només resten una part de la vida del jugador. Exactament 10% de vida si et donen un cop.

La vida restant es mostra amb una barra de vida visible a la pantalla, que s'actualitza automàticament cada vegada que el jugador rep dany.

Estat de la funcionalitat: Implementada completament.

2.5.7 Interfície d'usuari (UI)

El joc disposa d'una interfície senzilla perquè el jugador pugui entendre fàcilment tota la informació important.

S'ha implementat:

- Barra de vida
- Comptador de fruites o puntuació
- Menú principal
- Menú de pausa
- GameOver
- Pantalla de nivell completat
- Pantalla de victòria final.

El menú principal permet iniciar el joc o sortir-ne. El menú de pausa es pot obrir amb la tecla ESC o prement en el botó de menú de la part superior de la pantalla i congela el joc temporalment fins que el jugador ho reanudi.

Seguidament el menú de game over surt quan el jugador ha mort en qualsevol nivell i per últim tenim el menú quan es passa el nivell que pot anar al següent nivell o tancar el joc i el menú final que surt quan t'has pasat els 3 nivells, que podràs reiniciar el joc un altre cop o sortir.

Estat de la funcionalitat: Implementada completament.

2.5.8 Organització mitjançant scripts

Per mantenir el projecte ordenat, cada element important del joc té el seu propi script independent. Això facilita tant la programació com la correcció d'errors.

Alguns exemples són:

- **ninjacat.gd**: controla el moviment i la vida del jugador.
- **Fruit.gd**: controla les fruites i la puntuació.
- **spikes.gd**: gestiona les trampes i la mort instantània.
- **spikeball.gd**: gestiona el moviment de pèndol del spikeball i el dany.
- **serra.gd**: gestiona el moviment horitzontal i el dany.

Aquesta manera de treballar fa que el codi sigui més net, més fàcil d'entendre i més senzill de modificar en el futur.

Estat de la funcionalitat: Implementada completament.

3 Altres capítols

Aquesta memòria s'ha estructurat seguint una organització pensada per explicar de manera clara tot el procés que hem fet tots dos en el desenvolupament del projecte Fruit Quest. Tot i seguir la guia proporcionada, alguns apartats s'han adaptat segons les necessitats del projecte i els aspectes que hem considerat més importants durant el desenvolupament.

L'objectiu principal de la memòria és mostrar no és únicament mostrar el resultat final del videojoc , sinó també tot el procés que hi ha darrere: les investigacions realitzades, les decisions tècniques, els problemes trobats i les solucions aplicades durant el desenvolupament.

Durant el projecte s'han hagut de prendre diferents decisions tècniques després de comparar diverses opcions. Per exemple, abans de començar vam valorar diversos motors de videojocs i diferents llenguatges de programació.

Finalment, es va escollir Godot i GDscript perquè ofereixen una bona combinació entre facilitat d'ús, rendiment i aprenentatge útil per al futur. A més, el sistema basat en nodes i escenes de Godot facilita molt l'organització del projecte i la creació de videojocs 2D.

També s'han hagut de prendre decisions relacionades amb el disseny dels nivells, les mecàniques del jugador i la manera d'organitzar els codis. En alguns casos, hem fet proves amb diferents opcions fins trobar la que funcionava millor dins del joc. Un exemple és el sistema de moviment, on es van provar diferents configuracions de salts fins aconseguir una jugabilitat més fluida gràcies al doble salt i al salt coyote.

Pel que fa a l'organització dels codis, vam decidir separar cada funcionalitat en scripts independents perquè fos més fàcil corregir errors i mantenir el projecte ordenat. Aquesta decisió també facilita futures ampliacions del joc.

A més, durant el desenvolupament han aparegut problemes tècnics, especialment relacionats amb les col·lisions, la física del personatge i la sincronització entre animacions i moviments. La majoria d'aquests problemes es van solucionar mitjançant proves constants i modificacions dels scripts fins obtenir un resultat estable.

En general, tota la memòria està enfocada a explicar les decisions importants del projecte, els motius pels quals s'han pres i el procés seguit per arribar al resultat final. D'aquesta manera, es pot entendre millor tot el treball realitzat durant el desenvolupament de Fruit Quest.

4 Conclusions

4.1 Conclusions generals del projecte

El desenvolupament de Fruit Quest ha estat una experiència molt positiva tant a nivell acadèmic com personal. Durant el projecte hem pogut aplicar molts dels coneixements apresos al cicle d'SMX.

A nivell tècnic, hem après a utilitzar el motor Godot i a programar en GDscript dins d'un projecte real. També hem entès millor com funcionen les col·lisions, la física en videojocs, els sistemes de vida, les animacions i la gestió de nivells.

Moltes coses que al principi ens semblaven difícils, com el doble salt o el salt coyote, finalment les hem pogut implementar després de fer proves i corregir molts errors.

El projecte també ens ha ajudat a millorar la nostra manera de treballar en equip. Hem après a repartir tasques, comunicar-nos millor i ajudar-nos quan apareixien problemes tècnics. L'ús de GitHub ens ha ensenyat a treballar de manera més professional i ordenada.

A nivell personal, aquest projecte ens ha fet veure que crear un videojoc és molt més complex del que sembla, però també molt satisfactori quan les coses funcionen correctament. En general, Fruit Quest ens ha servit per guanyar experiència, confiança i nous coneixements que ens poden ser útils en futurs projectes o treballs relacionats amb la informàtica.

4.2 Consecució dels objectius

La majoria dels objectius definits al principi del projecte s'han assolit correctament.

L'objectiu principal, que era desenvolupar un videojoc de plataformes 2D funcional amb diferents nivells i mecàniques pròpies, s'ha complert satisfactòriament. El joc és jugable, estable i permet completar els nivells.

Pel que fa als objectius específics:

- El sistema de moviment del personatge s'ha implementat completament, incloent moviment lateral, salt, doble salt i salt coyote.
- Els 3 nivells del joc han estat creats i funcionen correctament, tot i que encara es podrien millorar alguns aspectes visuals.
- El sistema de col·lisions funciona correctament tant amb plataformes com amb trampes i fruites.
- El sistema de puntuació i recollida de fruites està completament implementat.
- Els menús principals i de pausa funcionen correctament.
- La barra de vida i el sistema de danys també s'han implementat sense problemes.
- Els efectes de so i la música s'han integrat al joc, encara que es podrien afegir més sons en futures versions.

En general, es considera que els objectius principals del projecte s'han assolit gairebé al complet, i el resultat final compleix amb la idea inicial plantejada.

4.3 Valoració de la metodologia i planificació

La metodologia de treball utilitzada durant el projecte ha estat adequada i ens ha ajudat molt a mantenir una bona organització. Hem treballat seguint una metodologia àgil, desenvolupant el joc per parts petites i fent proves constants després de cada canvi important.

La planificació inicial s'ha seguit bastant bé, encara que en alguns moments hem hagut d'adaptar el temps previst perquè certes parts han resultat més difícils del que semblava. Per exemple, la implementació del sistema de salts avançats i alguns problemes amb les col·lisions van requerir més temps de proves i correccions.

Tot i això, dividir el projecte en petites tasques ens ha ajudat a detectar errors ràpidament i evitar problemes més grans al final. L'ús d'eines com GitHub i Trello també ha estat molt útil per treballar en equip i tenir controlat què estava fet i què faltava.

Una altra cosa positiva ha estat fer proves constants del joc. Això ens ha permès anar millorant la jugabilitat i corregir errors abans que afectessin altres parts del projecte.

En general, la metodologia aplicada ha funcionat correctament i ha permès completar el projecte de manera ordenada i bastant propera a la planificació inicial.

4.4 Visió de futur

Tot i que el projecte actual ja és funcional i jugable, encara hi ha moltes idees que es podrien afegir en futures versions per millorar el joc.

Una de les possibles millores seria crear més nivells amb nous obstacles, trampes i mecàniques diferents per fer el joc més llarg i variat. També es podria afegir un “boss final” al final de l’últim nivell per donar més emoció al joc.

Una altra ampliació interessant seria implementar un sistema de guardat de partides i puntuacions màximes perquè els jugadors poguessin guardar el seu progrés i competir entre ells.

També es podria millorar l’apartat visual, afegint més animacions, efectes especials i backgrounds més detallats. A nivell sonor, es podrien incorporar més efectes de so i diferents músiques per cada nivell.

Finalment, una de les idees més interessants seria publicar el joc en plataformes com Itch.io perquè altres persones el poguessin descarregar i provar. Això permetria rebre opinions de jugadors reals i continuar millorant el projecte en el futur.

5. Glossari

Godot Engine	Motor de videojocs gratuït i de codi obert utilitzat per crear el joc Fruit Quest.
GDscript	Llenguatge de programació propi de Godot utilitzat per programar la lògica i les funcionalitats del joc.
Script	Fitxer de codi que controla el comportament d'un objecte dins del joc, com el moviment, les col·lisions o les interaccions.
Node	Element bàsic dins de Godot. Cada objecte del joc, com el jugador o una fruita, és un node.
Escena	Conjunt de nodes agrupats per formar una part del joc, com un nivell o un menú.
Sprites	Imatge 2D que representa un personatge, objecte o element visual del joc.
HUD	Informació que apareix a la pantalla mentre es juga, com la barra de vida o els punts.
FPS (Frames Per Segon)	Fotogrames per segon. Indica la fluïdesa que té el joc. Com més FPS hi ha, més suau es veu i més fluït anirà.
Col·lisions	Sistema que detecta quan dos objectes del joc es toquen entre ells.
Salt Doble	Mecànica que permet al jugador tornar un altre cop a saltar mentre està a l'aire.
Salt Coyote	Mecànica que dona un petit marge de temps per saltar després de caure d'una plataforma.
Assets	Recurs extern utilitzat al joc, com imatges, sons o animacions descarregades d'internet.
Tileset	Conjunt de blocs o textures utilitzats per construir els nivells del joc.

Background	Imatge de fons que dona ambientació als nivells del joc.
GitHub	Plataforma online utilitzada per guardar el codi del projecte i treballar en equip.
UI	Interfície d'usuari del joc, com els menús, botons o barres de vida i de puntuació.
Gameplay	Forma en què es juga el videojoc, incloent controls, dificultat i mecàniques.
Requisit Funcional	Funció o acció que el joc ha de poder fer obligatòriament.
Requisit No Funcional	Característica relacionada amb el rendiment, la qualitat o el funcionament general del joc.

6. Bibliografía

A continuació es mostren totes les fonts utilitzades durant el desenvolupament del projecte.

[1] Godot Engine Documentation

Godot Engine. *Official Documentation*.

<https://docs.godotengine.org/en/stable/>

Consultat: Setembre - octubre 2025.

[2] Itch.io (assets i recursos)

Itch.io. *Indie game assets and resources*.

<https://itch.io/>

Data de consulta: Octubre 2025.

[3] Patreon – Fertakus (audio)

Fertakus. *Patreon posts and audio resources*.

<https://www.patreon.com/c/fertakusM/posts>

Data de consulta: Abril 2026.

[4] Pixabay (efectes de so)

Pixabay. *Video game sound effects library*.

<https://pixabay.com/es/sound-effects/search/videojuego/>

Data de consulta: Abril 2026.

[5] Tutorial de YouTube 1

<https://youtu.be/R0EAD-exbmk>

Data de consulta: Noviembre 2025.

[6] Tutorial de YouTube 2

<https://youtu.be/R6bgruMmF6Y>

Data de consulta: Noviembre 2025.

[7] Tutorial de YouTube 3

<https://youtu.be/0clM5t5dfgg>

Data de consulta: Diciembre 2025.

[8] Tutorial de YouTube 4

https://youtu.be/Y_B6j15ITHU

Data de consulta: Diciembre 2025.

[9] Tutorial de YouTube 5

<https://youtu.be/2zZRQOzcw7Y>

Data de consulta: Enero 2026.

[10] Tutorial de YouTube 6

<https://youtu.be/cPHcBSzmf64>

Data de consulta: Enero 2026.

[11] Tutorial de YouTube 7

<https://youtu.be/l-qNLWPppnU>

Data de consulta: Febrero 2026.

[12] Tutorial de YouTube 8

<https://youtu.be/VxpUIp-ieUM>

Data de consulta: Febrero 2026.

[13] Tutorial de YouTube 9

https://youtu.be/hyTu5ZTR5Yk?si=jGX_ztyHxz43RSGi

Data de consulta: Febrero 2026.

[14] Tutorial de YouTube 10

<https://youtu.be/0cT507Cr6j4>

Data de consulta: Marzo 2026.

[15] Tutorial de YouTube 11

<https://youtu.be/rE8lZmVp5BA>

Data de consulta: Marzo 2026.

[16] Tutorial de YouTube 12

<https://youtu.be/yG7qg4uOQto>

Data de consulta: Abril 2026.

[17] Tutorial de YouTube 13

<https://youtu.be/qtF9ZTGilmo>

Data de consulta: Abril 2026.

[18] Tutorial de YouTube 14

https://youtu.be/C_GODDa9V0s

Data de consulta: Abril 2026.

[19] Tutorial de YouTube 15

https://youtu.be/JAe9j-9n0Tw?si=u7EhlseAlrq_De9p

Data de consulta: Mayo 2026.

7 Annexos

Annex 1. Enllaç al repositori del projecte

Fruit Quest està disponible públicament a GitHub. En aquest repositori es poden trobar diferents coses importants del nostre projecte i explicades detalladament com ho hem utilitzat tot dins el nostre projecte.

Repositori oficial del projecte:

<https://github.com/st3venn/FruitQuest.git>

El repositori inclou:

- Descripció del projecte
- Especificacions tècniques
- Controls del sistema
- Estructura de directoris
- Eines utilitzades
- Objectius del projecte
- Estat del projecte
- Licència