Contents

[Acknowledgements 2](#_Toc75004397)

[I. Introduction: 3](#_Toc75004398)

[1. I want to build an RPG game : 3](#_Toc75004399)

[2. Game Engine: 3](#_Toc75004400)

[3. Methodology : 3](#_Toc75004401)

[ Pre-production: 3](#_Toc75004402)

[ Production: 3](#_Toc75004403)

[ Post-production: 4](#_Toc75004404)

[ Operations: 4](#_Toc75004405)

[4. Unity Learn: 4](#_Toc75004406)

[5. Why Unity game engine: 6](#_Toc75004407)

[6. Workstation and tools: 6](#_Toc75004408)

[II. Pre-production: 7](#_Toc75004409)

[1. BrainStorming: 7](#_Toc75004410)

[2. Scooping a similar game: Eternium 8](#_Toc75004411)

[3. Filtering the brainstorming ideas based on available resources and scooping: 9](#_Toc75004412)

[4. Story+plot: 10](#_Toc75004413)

[5. Core mechanics: 10](#_Toc75004414)

[6. Game Play: 10](#_Toc75004415)

[7. Music/sound: 11](#_Toc75004416)

[8. Art style: 11](#_Toc75004417)

[9. Project Concept 11](#_Toc75004418)

[10. Project Timeline 12](#_Toc75004419)

[11. Use Case diagram: 14](#_Toc75004420)

[12. Class Diagram: 15](#_Toc75004421)

[13. Optimization: 15](#_Toc75004422)

[14. Chosen assets: 16](#_Toc75004423)

[15. Initial BackLog: Sprints and tasks in priority order: 18](#_Toc75004424)

[III. Production: 18](#_Toc75004425)

[1. Introduction :unity editor 18](#_Toc75004426)

[2. Introduction: Scripting with visual Studio 19](#_Toc75004427)

[3. Assets: Package Manager 20](#_Toc75004428)

[4. User Interface: The starting scene 20](#_Toc75004429)

[5. User Interface: The game scene 22](#_Toc75004430)

[6. 23](#_Toc75004431)

[IV. Post-production: 23](#_Toc75004432)

[V. Operations: 24](#_Toc75004433)

[VI. Conclusion: 25](#_Toc75004434)

# Acknowledgements

Mini-RPG game

# Introduction:

## I want to build an RPG game :

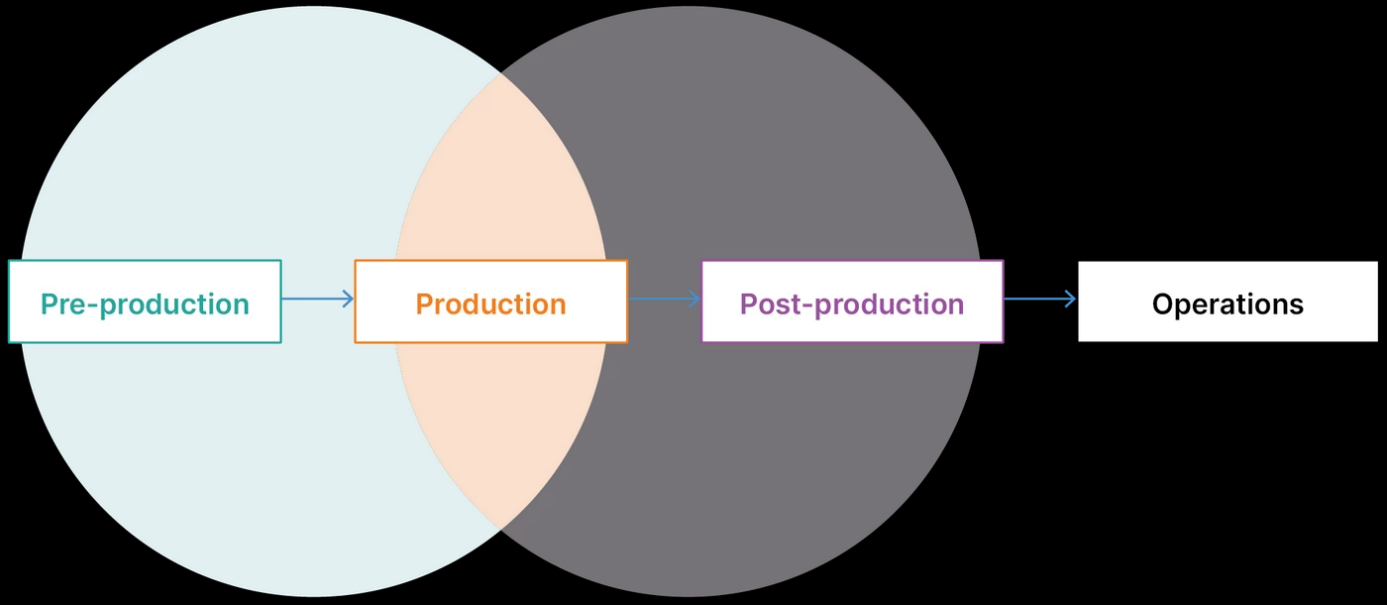
RPG stands for role-playing-game , a game in which players take on the roles of imaginary characters and go for adventures.In my game dev project I will start from ‘I want to build an RPG game’ a high level statement.a project design document is provided as well(HeroDesignPrim.docs).

## Game Engine:

Unity will be the choice of 3d real-time game engine.

* Loads of documentations and support threads.
* Loads of free assets for commercial use.
* Free\* (with a yearly revenue cap).

## Methodology :



It is consisted mainly of 4 phases of production:

### Pre-production:

planning, prototyping and initial designs. Scrum Style: backlog +sprints planning.

### Production:

creation of the product and assets ,user interfaces ,gameplay ,it is the important and largest part. It will be based on Kanban agile method.

### Post-production:

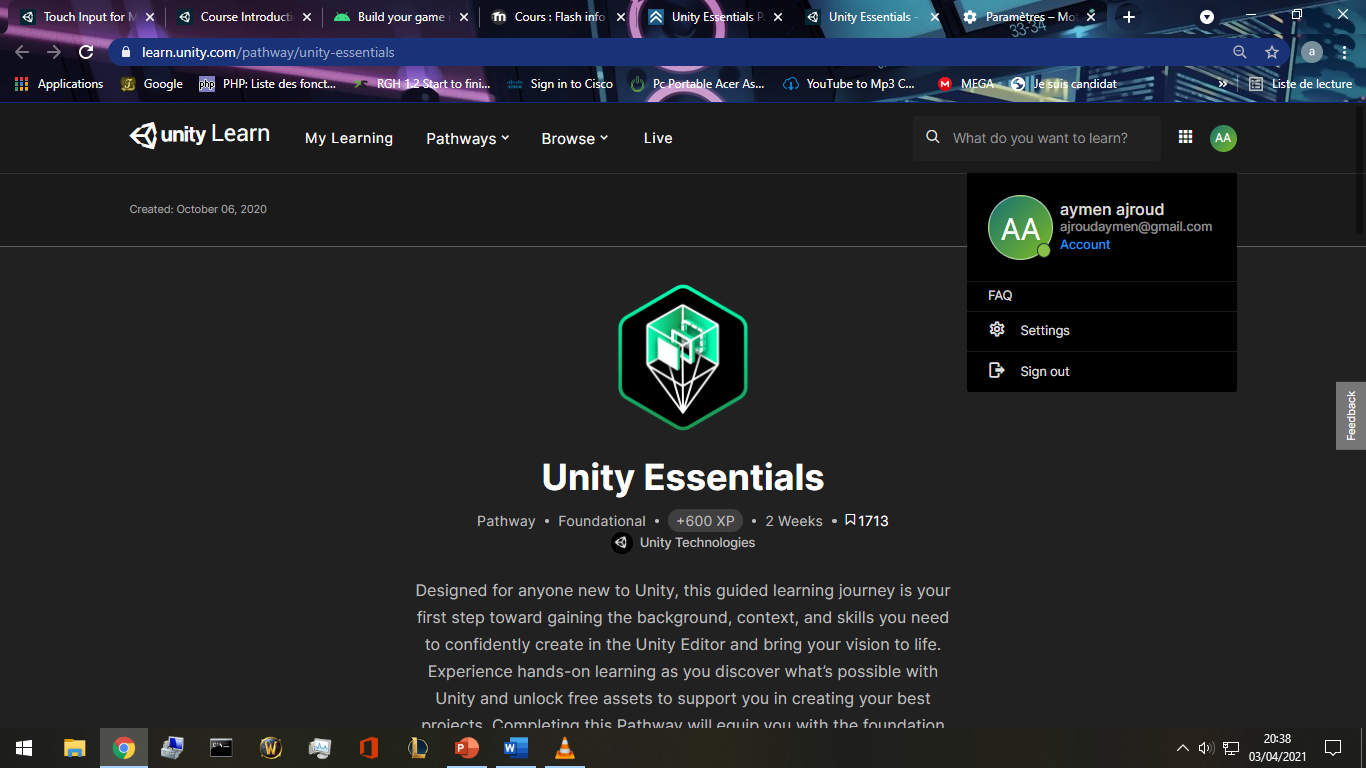
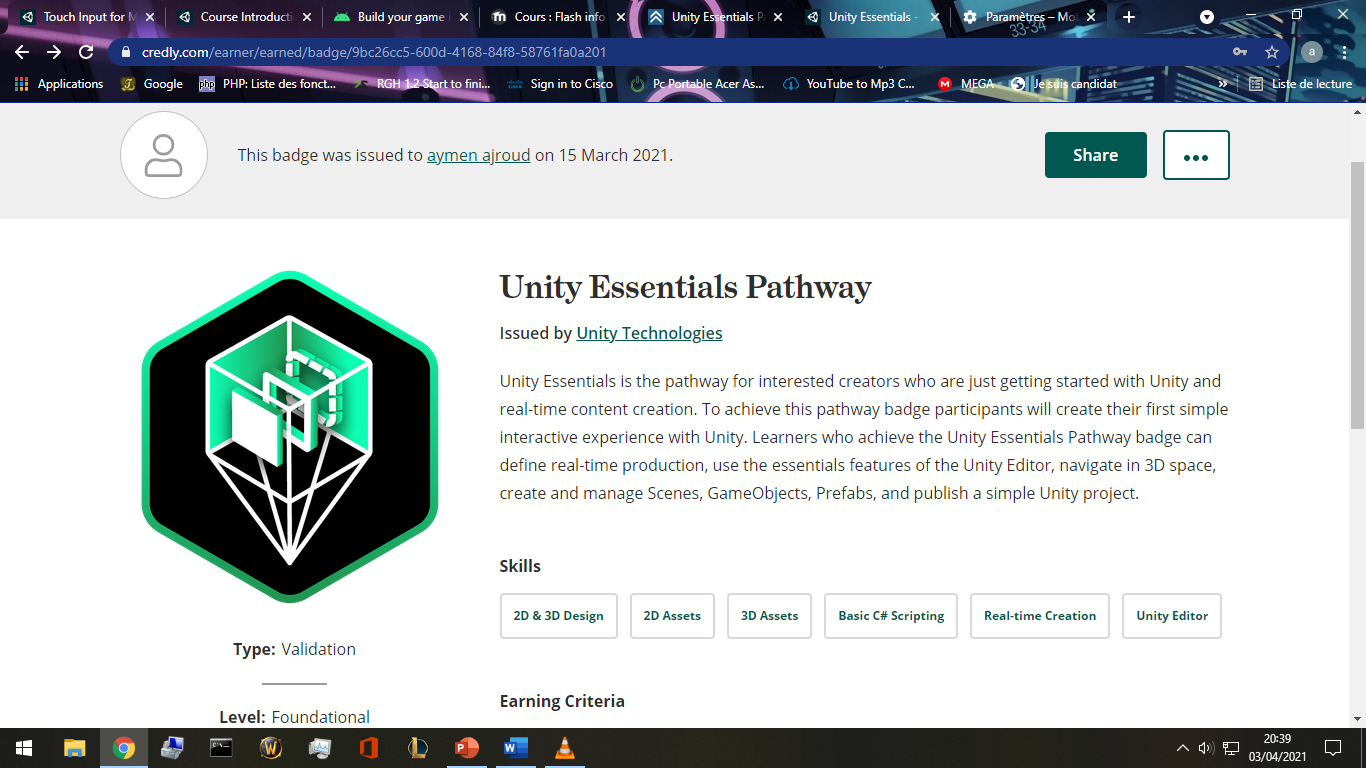
quality assurance (QA), editing, testing, bug fixing, and final polishing.

### Operations:

sales, monetization, updates (going through pre-production again), and continued maintenance.

## Unity Learn:

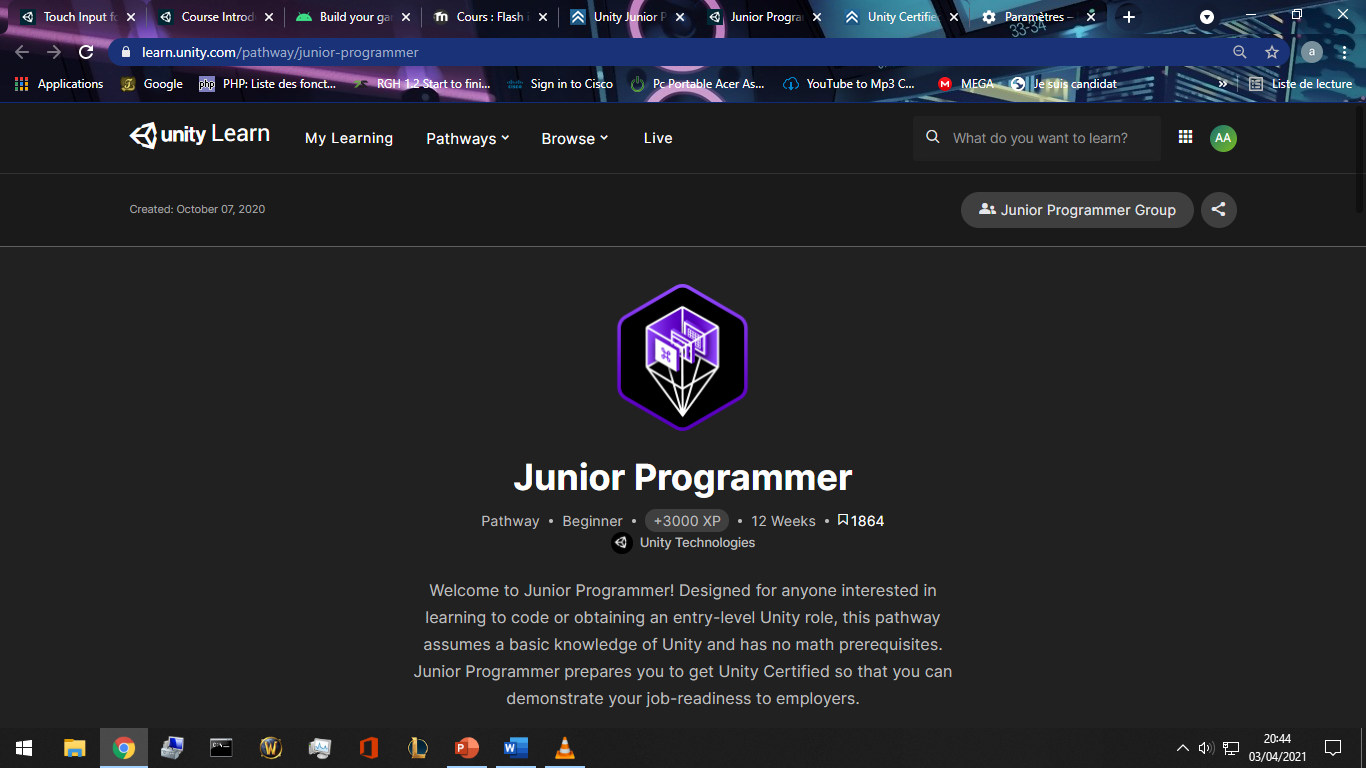
Unity Essential Pathway: 2 weeks duration: 1🡪15 Mars 2021

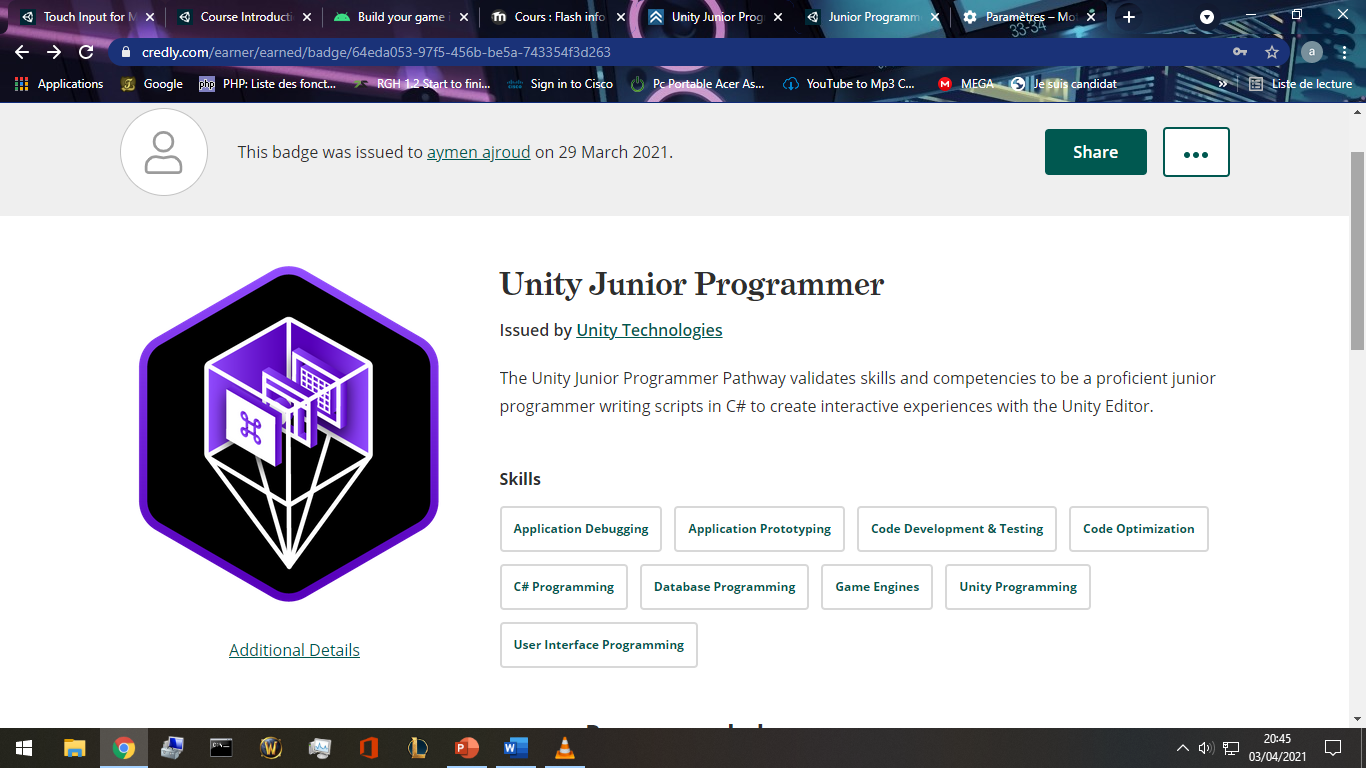
<https://learn.unity.com/pathway/unity-essentials>

[https://www.credly.com/badges/9bc26cc5-600d-4168-84f8- 58761fa0a201/public\_url](https://www.credly.com/badges/9bc26cc5-600d-4168-84f8-%2058761fa0a201/public_url)

* Unity Junior Programmer: 15🡪29 Mars 2021:

<https://learn.unity.com/pathway/junior-programmer>



<https://www.credly.com/badges/64eda053-97f5-456b-be5a-743354f3d263/public_url>

## Why Unity game engine:

* Full of license-free assets : <https://assetstore.unity.com/account/assets>.
* Rich documentation :a huge online manual, <https://docs.unity3d.com/Manual/index.html>
* a huge communities with tons of threads and debugging: forum.unity.com ,
* blogs.unity3d.com, reddit.com.
* C# base coding language.
* Real-Time testing :I can run the game whenever I want using the editor built in
* real-time emulator along with a “Profiler” that calculate frames, resources consumption, all technical info , all for the cause of optimization.
* An impressive link with microsoft visual studio for advanced scripting and easy documentation.

## Workstation and tools:

* Motherboard: H310M PRO-VD PLUS
* CPU: intel i5 9400F 2.9 GHz
* GPU: Nvidia GeForce GTX 1650
* Screen: 32” Vega Led
* Microsoft office suite validated with the institute office 365 email [aymen.ajroud@sousse.r-iset.tn](mailto:aymen.ajroud@sousse.r-iset.tn) .
* Umlet: a powerful diagram designer and freeware.
* SketchBook: a freeware and easy to use image editor.
* Microsoft windows Paint.(who doesn’t know Paint)
* Unity editor 2020.3.5f1 LTS
* Blender
* GitHub student pack: GitHub pro.activated by applying with institute email. With GitHub pro student pack I have access to more then hundred hosting/storage/databases and many more services on 3rd party providers like Microsoft azure student pack, canva.com , digital ocean,… <https://education.github.com/pack>
* Firebase.google.com , free basic plans for start ups, cloud storage, Realtime database NoSQL , statistics , web hosting, user authentication .Easy and documented implementation with unity projects.
* Hundreds of free assets in unity assets store for commercial use.

# Pre-production:

## BrainStorming:

I’ve spent 3 days writing down any idea I think of,at any time, and anywhere.

* Level system with experience points.
* With/without level cap.
* Stamina (hp), scalable with items/level
* Multiplayer
* Offline solo
* Openworld
* Stages , each stage have small map, stage=scene.
* Classes like warrior rogue mage priest
* Resources like mana rage energy
* Casting skills/spells
* Talent points tree
* Achievements system
* Google play implementation
* Guest login/ linking account
* Skills and spells upgradable
* Money system (gold , coins, gems,)
* Storyboard (implementing a manga/anime…)
* Characters to start with(female/male) race humain/other imaginairies.
* Quest/reward system.
* Npc (NonPlayerCharacters) quests, trading (sell buy).
* Inventory system
* Character equip system
* Character stats interface
* Enemies to kill (drop items/money/experience/quests requirements).
* Moving touch pad, spells/skills button list
* Free look button
* Swipe to fast cast spells/skills
* duel with other players with pvp reward system
* daily + weekly quests
* 2 or more factions to join
* Guild system
* Messaging system/mail box
* Compare achievements with others in a scoreboard
* Items with levels and rarity
* Fixed view
* Customized view
* Options interface (menu quit volume graphic quality)
* Mini map
* Customizing the controls size/visibility/position
* Pets that follow you and help attack
* Compagnions that can equip items and use skills.
* Crafting tab: salvage items and create higher lvl ones.
* Tap enemies to attack
* Payment ui.
* Googleplay plugin
* Global/guild/party chat system
* Party with friends and play together

## Scooping a similar game: Eternium

* I really like the gameplay but it feels like I hit the Wall where the Grind begins and nothing else other than running endless Trials is progress. Crafting Items isnt worth it cause i need Set/Named items. And there is nothing else to farm other than CL, everything else will drop or wont. *Reddit.com/*
* **Eternium** is an amazingly fun and beautifully crafted Action RPG, reminiscent of the **great** classics. **Eternium** stands out from other mobile Action RPGs by its effortless “tap to move” and innovative “swipe to cast” controls, and its player-friendly “no paywalls, never pay to win” philosophy. *Amazon.in/*

*\*\*google playstore reviews\*\**

* Stats decrease when u lvl up , why not increased? lacking joystick support.
* Upgrade skills by gems instead of skill points. Sorta of pay to win.
* No stamina/energy to limit gameplay.
* Old graphics
* The concept to draw lines to use items/cast spell or skill is innovative.
* Low experience gain rate.
* Grinding for items.
* Free play, don’t have to daily check in.
* No blood or gore so all ages are welcome.
* Small font and bad resolution compatibility on some phones.
* No too easy where it becomes boring.
* Returning to a level and grinding again and again for experience with same story dialogues showing is boring.
* Bad item drop rate
* Progression for new characters are slow.
* Problem with the crafting tab user interface UI.
* Pvp(player-vs-player) is unbalanced, matching players gear is bad.
* Interactive support in google playstore review section.

## Filtering the brainstorming ideas based on available resources and scooping:

* Level system with experience points.
* With~~/without~~ level cap.
* Stamina (hp), scalable with items/level
* ~~Multiplayer~~
* Offline solo
* OpenWorld
* Stages , each stage have small map, stage=scene.
* ~~Classes like warrior rogue mage priest~~
* Resources like mana rage energy
* Casting skills/spells
* Talent points tree
* Achievements system
* Google play implementation
* Guest login/ linking account
* Skills and spells upgradable and unlockable by lvling up
* Money system (gold , coins, gems,)
* Storyboard (implementing a manga/anime…)
* Characters to start with(female/male) race humain~~/other imaginairies.~~
* Quest/reward system.
* Npc (NonPlayerCharacters) quests, trading (sell buy).
* Inventory system
* Character equip system
* Character stats interface
* NPC Enemies to kill (drop items/money/experience/quests requirements).
* Moving touch pad, spells/skills button list
* Free look button
* Swipe to fast cast spells/skills
* ~~duel with other players with pvp reward system~~
* daily + weekly quests
* ~~2 or more factions to join~~
* ~~Guild system~~
* Messaging system/mail box
* Compare achievements with others in a scoreboard
* Items with levels and rarity
* Fixed view
* ~~Customized view~~
* Options interface (menu quit volume graphic quality)
* Mini map
* Customizing the controls size/visibility/position
* ~~Pets that follow you and help attack~~
* ~~Compagnions that can equip items and use skills.~~
* ~~Crafting tab: salvage items and create higher lvl ones.~~
* ~~Tap enemies to attack~~
* Payment ui.
* Google play plugin
* Global/guild/party chat system
* ~~Party with friends and play together~~

## Story+plot:

A hero came to save the village from skeletons, take quests, gather, kill , find, and return for exp and glory.

## Core mechanics:

* player can move freely.
* Talk to npcs in the village for questing.
* Attack goblins.
* Find stolen items.
* Gather herbs.

## Game Play:

1st u choose between a male/female character,then u spawn beside a village. You can move freely in the open world. Kill threatening skeletons ,find rare items, collect herbs for quests to earn exp and lvl up.

## Music/sound:

A calm music running in background.

## Art style:

Medieval style of sword and magic.free assets from unity strore.

## Project Concept

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1  **Player Control** |  | You control a   |  | | --- | | *A male/female character* | | in this   |  |  | | --- | --- | | *Top down view* | game | |
|  | where   |  | | --- | | *The touch pad+buttons* | | makes the player   |  | | --- | | *move run attack* | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 2  **Basic**  **Gameplay** |  | During the game,   |  |  | | --- | --- | | *Enemies/friendly npcs* | appear | | from   |  | | --- | | *Moving around the open world stage* | |
|  | and the goal of the game is to   |  | | --- | | *Complete missions, earn money, upgrade skills and leveling up* | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 3  **Sound**  **& Effects** |  | There will be sound effects   |  | | --- | | *when moving, attacking, dying, enemy dying, friendly npcs talk* | | and particle effects   |  | | --- | | *when Dying, moving, finishing quests, leveling up* | |
|  | [*optional*] There will also be   |  | | --- | | *All characters have basic animation effect.* | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4  **Gameplay**  **Mechanics** |  | As the game progresses,   |  | | --- | | *Enemy level increases, quests rewards increases, experiencee needed to lvl up increases, talent points* | | making it   |  | | --- | | *Scallable* | |
|  | [*optional*] There will also be   |  | | --- | | *when you die, you need to watch an ad video or wait ? minutes to respawn* | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 5  **User**  **Interface** |  | The   |  | | --- | | *Money* | | will   |  | | --- | | *Increase* | | whenever   |  | | --- | | *loot enemies/sell items* | |
|  | At the start of the game, the title   |  |  | | --- | --- | | *hero* | will appear | | | and the game will end when   |  | | --- | | *Doesn't end,level cap, dangeons, bosses, loot, items.* | |

## Project Timeline

|  |  |  |
| --- | --- | --- |
| Milestone | Description | Due |
| **#1** | |  | | --- | | * *Menu/start/quit ui + 1st stage+controls+characters+npc+ennemies+deploy to googleplay store* | | |  | | --- | | *29/04* | |
| **#2** | |  | | --- | | * *log in functionality+google play integration+back end server+databases* | | |  | | --- | | *29/05* | |
| **#3** | |  | | --- | | * *optimization+adding more features +ads+billing system* | | |  | | --- | | *15/06* | |
| **#4** | |  | | --- | | * *more code optimization+release of more stages.+facebook page+advertizing* | | |  | | --- | | *30/06* | |

## Use Case diagram:Diagram Description automatically generated

## Class Diagram:

Diagram

Description automatically generated

## Optimization:

When developing, optimization should come naturally while coding,here are some tips for RPG kind of games:

* Reduce the music/audios format to keep it light.
* Compress/reuse textures.
* Remove unused assets items.
* Smaller games are faster to download and require less data on initial install, and eventually higher install conversions.Texture compression formats.
* Reduce events number in a scene.
* Reduce/re-use effect plugins (lightning, sounds, particles [or anything that includes "advanced" physics], etc.).
* Support 64-bits architectures.
* Use unity profiler and analyze game performance/requirements.
* Clean code.
* Always add comments.
* Keep it simple.
* Static batching (static objects reduce hugely resource consumption, so whenever an object doesn’t move, make it static).
* Dynamic batching (combine meshes ).
* Culling to limit what’s rendering (Imagine standing in front of a door and still rendering all objects behind that door. This is where Occlusion Culling comes in).
* Reducing physics computing (It's important to note that having too many Rigidbody objects within your game will affect performance negatively.)
* Object Pooling(Pooling commonly used objects allows you to reuse them over and over again without destroying them).
* Coroutines/update(update will happen every frame, so coroutines sometimes reduce a lot of work).
* Cashing Components(for example Don't use GetComponent multiple times in a method when you can cache it once and use that variable throughout the script.).
* Better to manually assign objects in the script component then using the gameobject.find().
* if you're not using something, turn it off.
* in general you should use the minimum amount of lights necessary to achieve your desired style.
* Enable Instancing option within unity.

## Chosen assets:

|  |  |  |
| --- | --- | --- |
| A group of mannequins  Description automatically generated with low confidence | <https://assetstore.unity.com/packages/3d/characters/humanoids/girl-with-clothes-worker-set-162925> | A female villager with idle animation only |
| screenshot | <https://assetstore.unity.com/packages/3d/characters/humanoids/fantasy/oriental-fantasy-character-berserker-grade-0-171147> | Female hero with animations |
| screenshot | <https://assetstore.unity.com/packages/3d/characters/humanoids/humans/fantasy-chess-rpg-character-arthur-160647> | Male Hero with animations |
| screenshot | <https://assetstore.unity.com/packages/2d/gui/fantasy-wooden-gui-free-103811> | Ui materials |
| screenshot | <https://assetstore.unity.com/packages/3d/environments/nature-starter-kit-2-52977> | Nature kit 3d |
| A picture containing text, indoor  Description automatically generated | <https://assetstore.unity.com/packages/3d/characters/humanoids/fantasy/rpg-skeleton-35463> | Skeleton 3d with animations |
| A picture containing graphical user interface  Description automatically generated | <https://assetstore.unity.com/packages/2d/gui/icons/basic-rpg-icons-181301> | Basic rpg game icons |
|  | <https://assetstore.unity.com/packages/3d/environments/fantasy/fantasy-forest-environment-free-demo-35361> | Trees/bush 3d prefabs |
| screenshot | <https://assetstore.unity.com/packages/tools/version-control/github-for-unity-118069> | GitHub unity plugin |

## Initial BackLog: Sprints and tasks in priority order:

* UI: user interface, login/link interface (dead buttons), support, faq,
* Building initial map
* Graphic settings interface
* Inventory system
* Equip system
* Controls: player input, touch pad.
* Hero stats
* Villager npcs,quest system,and enemy skeletons.
* Environment details,map details.

# Production:

## Introduction :unity editor

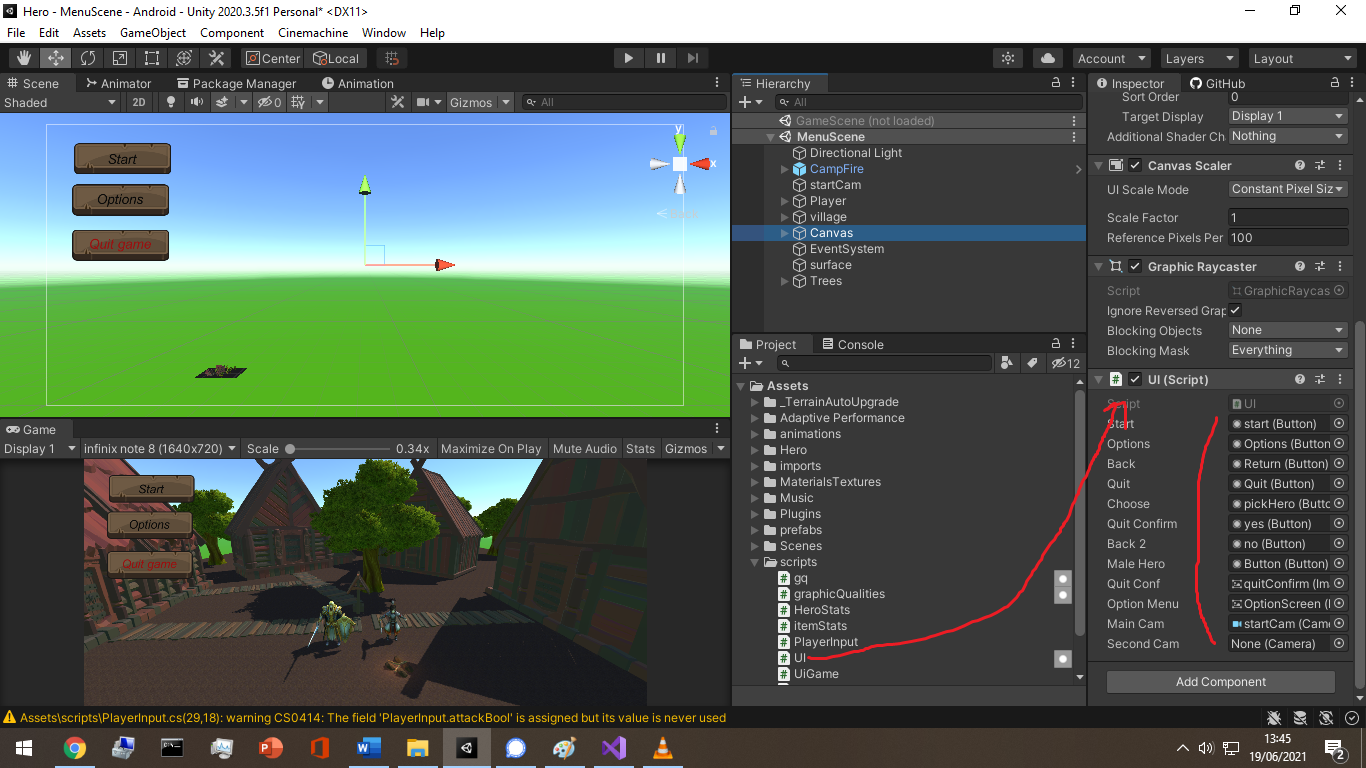
A screenshot of a computer

Description automatically generated with medium confidence

## Introduction: Scripting with visual Studio

A picture containing text, electronics, monitor, screenshot

Description automatically generated



When declaring an object as a public variable inside a script, I need to drag and drop it in the inspector otherwise It gives a null pointer reference if I try to access an empty object.

Example of treatment on a text object : change it value, turn it to inactive state, change its parameters , all through scripting.

## Assets: Package Manager

The built in unity 2020 packageManager can directly download and import assets from my owned assets related to my unity account. Graphical user interface, website

Description automatically generated

All assets are free to use commercially.

Exterior websites for other assets: <https://www.cgtrader.com/> <https://soundimage.org/>

A picture containing text, screenshot, monitor, indoor

Description automatically generated

## User Interface: The starting scene

all the canvas children are attached and declared in its script as variables to do what ever is needed along the scripting.

Text

Description automatically generated

In the case of a button/image or any object that should start an event when triggered can extend the onclick events, or I can simply use the built in onclick trigger in the unity inspector.

That’s almost how ui objects event system works.All the treatments and the events are inside the called method in the onClick Panel.

A screenshot of a computer

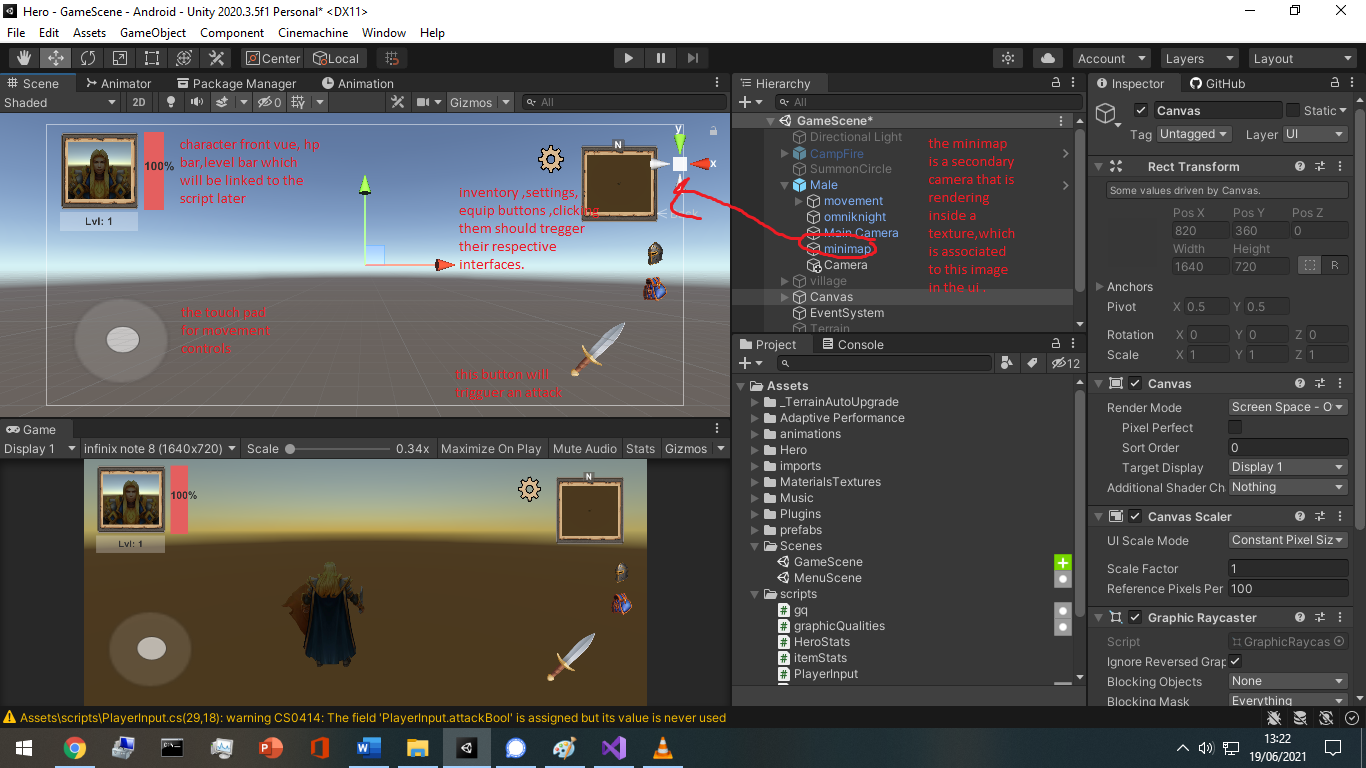
Description automatically generated with medium confidence

As for saving the settings stats and players preferred settings, there exists a static class in unity named PlayerPrefs that can save data as a key-value pairs.and this is almost the same concept for every other settings.

Graphical user interface

Description automatically generated

## User Interface: The game scene



When building ui interface, I found that the order in the hierarchy matters a lot since ui objects in the upper order are rendered on top of the lower level ones.

The input key “escape” is the same as the android return button , so I need to implement this input in order to interact with the interface as well.

The input system should always be in the update() method in a script.

Update() is called every frame, and every device screen have a frame rate and is calculated as fps (frame per second).a frame is simply an image.

## Inventory/equip system:

A picture containing text, indoor, screenshot, electronics

Description automatically generated

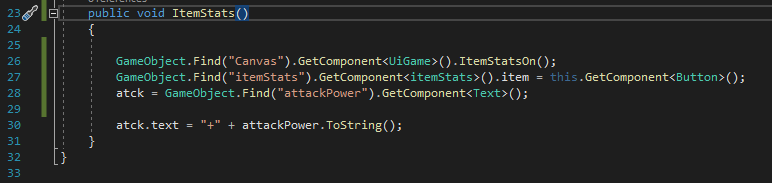
An item is built with a ui button interface, it has an icon and on click, it shows its stats in a panel.this is what we call in unity , a prefab.

Here an example of a sword level 1 with 100 attack power and ready for adding more stats and other info and constrains.

Graphical user interface

Description automatically generated

Here is an example pf script.



This method is called when the item is clicked on and this is where the item stats will be transferred to the item stats panel on the left side.

Text

Description automatically generated

This is called when the destroy item is clicked.

## Touch input for smartphones:

Smartphone devices are capable of tracking up to five fingers touching the screen simultaneously. You can retrieve the status of each finger touching the screen during the last frame by accessing the *Input.touches* property array.

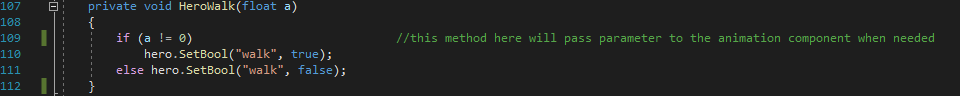
The most important property parts are the position, id, and the phase.

The phase have 5 states: Began, Moved, Stationary, ended and canceled.

## Player Input:

Text

Description automatically generated

When dealing with player moving object, animations and FX either visual or sound. 

A picture containing text

Description automatically generated

Graphical user interface

Description automatically generated

# Post-production:

# Operations:

# Conclusion: