

Contents

Acknowledgements	2
I. Introduction:.....	3
1. I want to build an RPG game :	3
2. Game Engine:.....	3
3. Methodology :	3
• Pre-production:.....	3
• Production:.....	4
• Post-production:	4
• Operations:	4
4. Unity Learn:	4
5. Why Unity game engine:.....	6
6. Workstation and tools:	Error! Bookmark not defined.
II. Pre-production:	6
1. BrainStorming:.....	6
2. Scooping a similar game: Eternium.....	8
3. Filtering the brainstorming ideas based on available resources and scooping:	8
4. Story+plot:	10
5. Core mechanics:.....	10
6. Game Play:.....	10
7. Music/sound:	10
8. Art style:.....	10
9. Project Concept	11
10. Project Timeline	13
11. Use Case diagram:.....	14
12. Class Diagram:	15
13. Chosen assets:	15

14. Initial BackLog: Sprints and tasks in priority order:	19
III. Production:	20
1. Introduction :unity editor.....	20
2. Introduction: Scripting with visual Studio	20
3. Assets: Package Manager	22
4. User Interface: The starting scene	23
5. User Interface: The game scene	25
6. Inventory/equip system:	26
7. Touch input for smartphones:	28
8. Player Input:	28
9. Hero stats:	29
10. Unity Mechanim:	30
11. Unity Rigging:	30
12. NPC prefabs:	31
13. Terrain building tools:.....	32
IV. Post-production:	33
1. Unity Profiler:	33
2. Optimisation and code review:	34
3. Bug fixing:.....	35
4. Version Control and data safety:.....	36
5. Polish:.....	37
V. Operations:	37
1. Google/iOS developer account:	37
2. Unity monetizing:	38
3. Reviews and updates:.....	38
4. Updates and continued maintenance:.....	38
VI. Conclusion:	39

Acknowledgements

First and foremost, praises and thanks to the God, the Almighty, for His showers of blessings throughout my work to complete the project² successfully.

I would like to express my deep and sincere gratitude to my family who supported me from the start, Habib and Samia, Ahmed Yassine my little brother, Roua, Mehdi, Lotfi.

A special thank you to Randa for her huge support and tips with the paperwork and testing.

I am extremely grateful for their love, prayers, caring and sacrifices for educating and preparing me for my future.

I hold a huge respect and gratitude to all my teachers and supervisors through my 3 year studies at ISET Sousse, Prof Adnen ROUATBI, Prof Nizar ROUATBI, Prof Moez ZOUARI, Prof Wissem ELTAIEF.

Also I express my thanks especially to my supervisor Prof Adel LAARIF who guided me well and gave me precious tips and wise guidance that helped me advance and learn a lot through this project.

Finally, my thanks go to all the people who have supported me to complete the research work directly or indirectly.

Mini-RPG game

I. Introduction:

1. I want to build an RPG game :

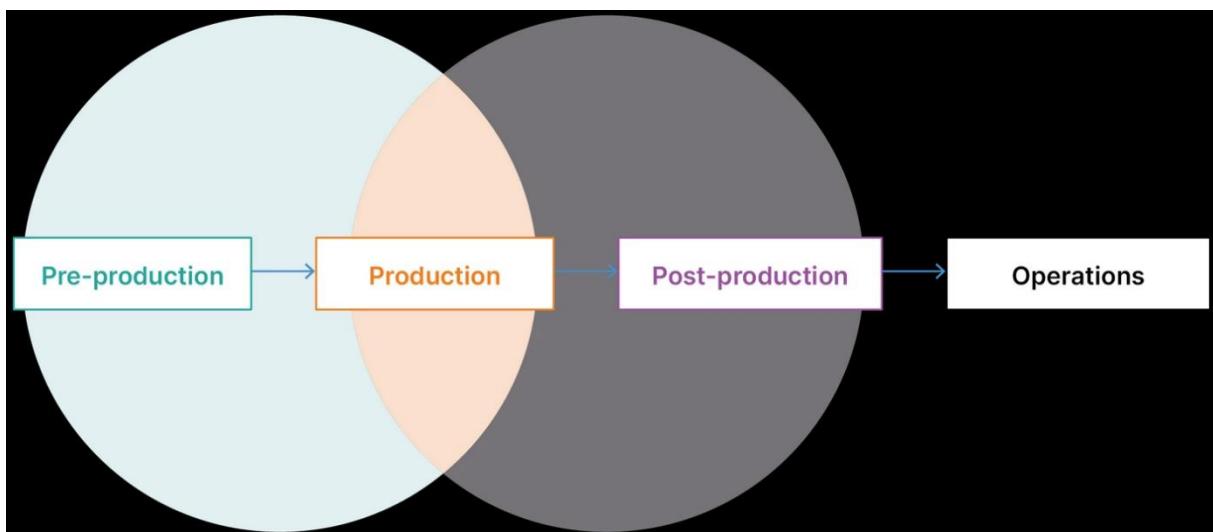
RPG stands for role-playing-game , a game in which players take on the roles of imaginary characters and go for adventures. In my game dev project I will start from 'I want to build an RPG game' a high level statement.

2. 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).

3. 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.

4. Unity Learn:

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

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

The screenshot shows a web browser displaying the Unity Learn platform. The main navigation bar includes 'My Learning', 'Pathways', 'Browse', and 'Live'. A search bar at the top right asks 'What do you want to learn?'. On the left, a sidebar shows the user's account information: 'aymen ajroud' with the email 'ajroud.aymen@gmail.com' and an 'Account' button. Below this are links for 'FAQ', 'Settings', and 'Sign out'. The central content area features a large green hexagonal badge for the 'Unity Essentials Pathway'. To the left of the badge is a circular profile picture placeholder. To the right, there are buttons for 'Share' and '...' (more options). Below the badge, the text 'Unity Essentials Pathway' is displayed, followed by 'Issued by [Unity Technologies](#)'. A detailed description of the pathway is provided: 'Unity Essentials is the pathway for interested creators who are just getting started with Unity and real-time content creation. To achieve this pathway badge participants will create their first simple interactive experience with Unity. Learners who achieve the Unity Essentials Pathway badge can define real-time production, use the essentials features of the Unity Editor, navigate in 3D space, create and manage Scenes, GameObjects, Prefabs, and publish a simple Unity project.' Underneath the description, a section titled 'Skills' lists several categories: '2D & 3D Design', '2D Assets', '3D Assets', 'Basic C# Scripting', 'Real-time Creation', and 'Unity Editor'. At the bottom of the badge area, it says 'Type: Validation' and 'Level: Foundational'. The bottom of the screen shows a Windows taskbar with various pinned icons and the date/time '03/04/2021 20:39'.

https://www.credly.com/badges/9bc26cc5-600d-4168-84f8-58761fa0a201/public_url

- Unity Junior Programmer: 15→29 Mars 2021:

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

The screenshot shows the Unity Learn website with the 'Junior Programmer' pathway selected. The page features a large purple hexagonal badge icon at the top. Below it, the title 'Junior Programmer' is displayed in bold white text. Underneath the title, it says 'Pathway • Beginner • +3000 XP • 12 Weeks • 1864' and 'Unity Technologies'. A descriptive paragraph explains the pathway is designed for beginners interested in learning to code or obtaining an entry-level Unity role, assuming basic knowledge of Unity and no math prerequisites. It prepares users for Unity Certification. The bottom of the screenshot shows a Windows taskbar with various icons.

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

The screenshot shows the Credly badge details page for the 'Unity Junior Programmer' badge. It features a large purple hexagonal badge icon. To the left of the badge, there is a circular profile picture placeholder. Below the badge, the text 'This badge was issued to [aymen ajroud](#) on 29 March 2021.' is displayed. On the right side, there are two buttons: 'Share' and '...' (more options). Below the badge, the title 'Unity Junior Programmer' is shown in bold, followed by 'Issued by [Unity Technologies](#)'. A descriptive paragraph states that the pathway validates skills and competencies for a proficient junior programmer writing scripts in C# to create interactive experiences with the Unity Editor. A 'Skills' section lists several categories: Application Debugging, Application Prototyping, Code Development & Testing, Code Optimization, C# Programming, Database Programming, Game Engines, Unity Programming, and User Interface Programming. At the bottom, there is a link 'Additional Details'.

5. 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.

II. Pre-production:

1. 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

2. 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 isn't 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.

3. 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~~

4. Story+plot:

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

5. Core mechanics:

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

6. Game Play:

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

7. Music/Sound:

A calm music running in background.

8. Art style:

Medieval style of sword and magic.

9. Project Concept

1	You control a <i>A male/female character</i>	in this <i>Top down view</i>	game
Player Control	where <i>The touch pad+buttons</i>	makes the player <i>move run attack</i>	
2	During the game, <i>Enemies/friendly npcs</i>	from <i>appear</i>	<i>Moving around the open world stage</i>
Basic Gameplay			
and the goal of the game is to			<i>Complete missions, earn money, upgrade skills and leveling up</i>
3	There will be sound effects <i>when moving, attacking, dying, enemy dying, friendly npcs talk</i>	and particle effects <i>when Dying, moving, finishing quests, leveling up</i>	
Sound & Effects			

[optional] There will also be

All characters have basic animation effect.

4

As the game progresses, making it

Gameplay

Mechanics

Enemy level increases, quest rewards increases, experience needed to lvl up increases, talent points

Scallable

[optional] There will also be

when you die, you need to watch an ad video or wait ? minutes to respawn

5

The will whenever

User

Interface

Money

Increase

loot enemies/sell items

At the start of the game, the title and the game will end when

hero

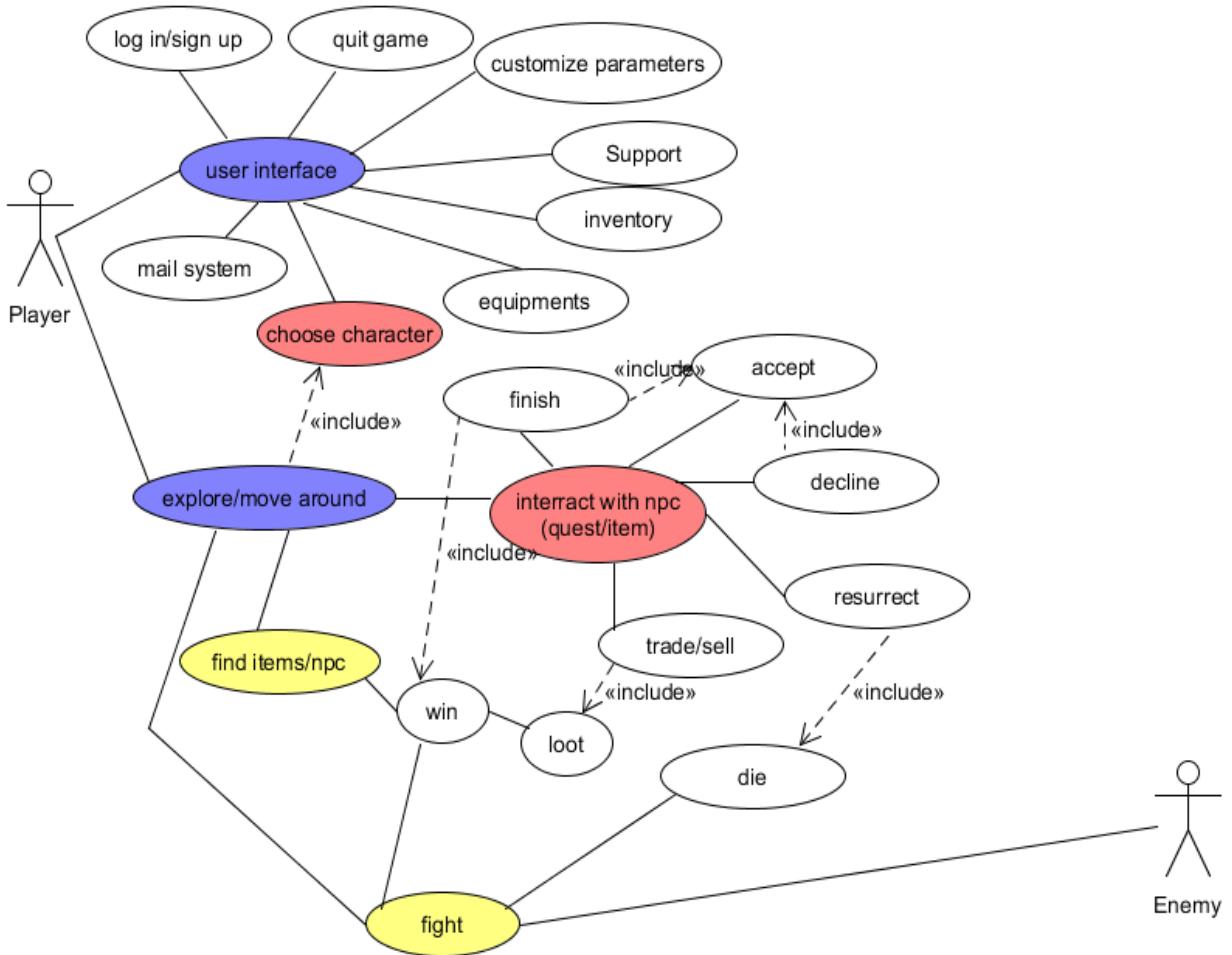
will appear

Doesn't end, level cap, dungeons, bosses, loot, items.

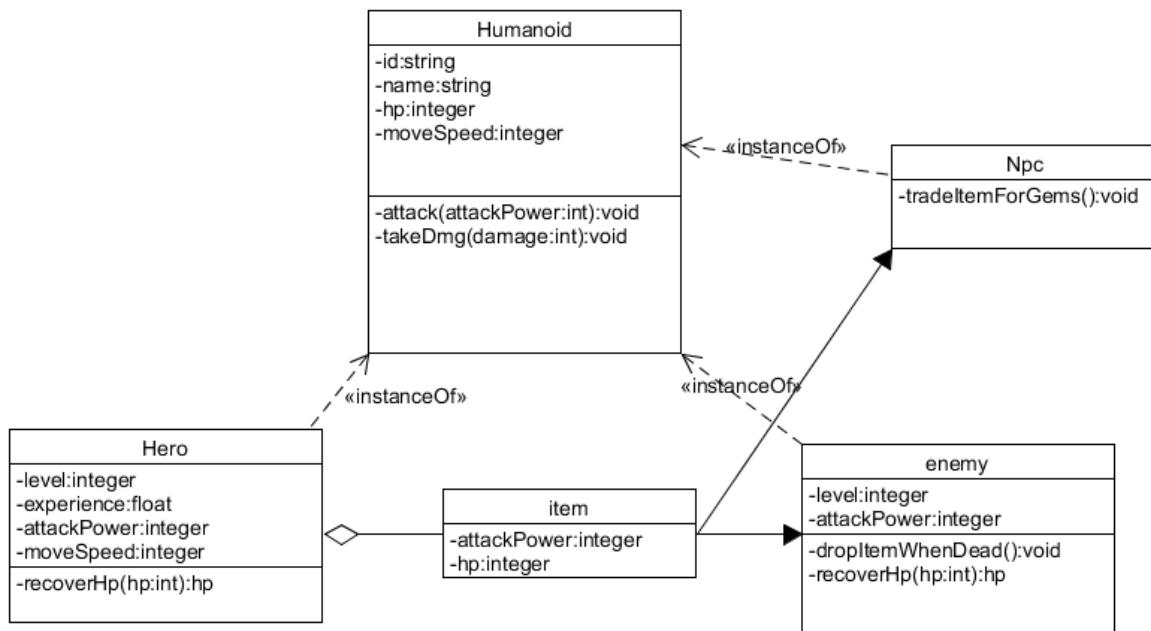
10. Project Timeline

Milestone	Description	Due
#1	<ul style="list-style-type: none">- Menu/start/quit ui + 1st stage+controls+characters+npc+ennemis+deploy to googleplay store	29/04
#2	<ul style="list-style-type: none">- log in functionality+google play integration+back end server+databases	29/05
#3	<ul style="list-style-type: none">- optimization+adding more features +ads+billing system	15/06
#4	<ul style="list-style-type: none">- more code optimization+release of more stages.+facebook page+advertisizing	30/06

11. Use Case diagram:

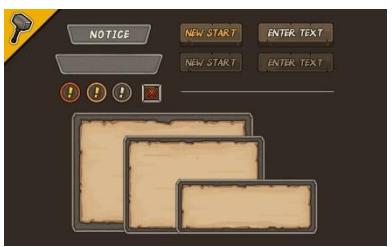
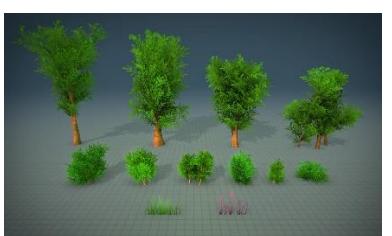


12. Class Diagram:

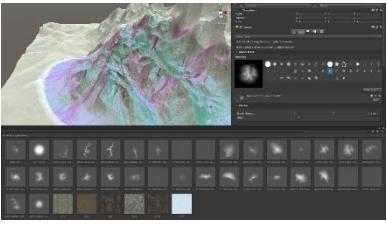


13. Chosen assets:

	https://assetstore.unity.com/packages/3d/characters/humanoids/girl-with-clothes-worker-set-162925	A female villager with idle animation only
	https://assetstore.unity.com/packages/3d/characters/humanoids/fantasy/oriental-fantasy-character-berserker-grade-0-171147	Female hero with animations

	https://assetstore.unity.com/packages/3d/characters/humans/humans/fantasy-chess-rpg-character-arthur-160647	Male Hero with animations
	https://assetstore.unity.com/packages/2d/gui/fantasy-wooden-gui-free-103811	Ui materials
	https://assetstore.unity.com/packages/3d/environments/nature-starter-kit-2-52977	Nature kit 3d
	https://assetstore.unity.com/packages/3d/characters/humans/fantasy/rpg-skeleton-35463	Skeleton 3d with animations
	https://assetstore.unity.com/packages/2d/gui/icons/basic-rpg-icons-181301	Basic rpg game icons

	<p>https://assetstore.unity.com/packages/3d/environments/fantasy/fantasy-forest-environment-free-demo-35361</p>	Trees/bush 3d prefabs
	<p>https://assetstore.unity.com/packages/tools/version-control/github-for-unity-118069</p>	GitHub unity plugin
	<p>https://assetstore.unity.com/packages/3d/animations/villager-animations-free-157920</p>	Villager animation pack :mining , fishing Works on humanoid unity rig.
	<p>https://assetstore.unity.com/packages/vfx/particles/fire-explosions/fire-spell-effects-36825</p>	Fire & Spell Effects
	<p>https://assetstore.unity.com/packages/3d/characters/humanoids/character-elf-114445</p>	Character elf

	<p>https://assetstore.unity.com/packages/3d/characters/humans/humans/basic-bandit-89978</p>	<p>Basic bandit character</p>
	<p>https://assetstore.unity.com/packages/3d/animations/rpg-character-mecanim-animation-pack-free-65284</p>	<p>RPG Character Mecanim Animations</p>
	<p>https://assetstore.unity.com/packages/2d/textures-materials/nature/terrain-tools-sample-asset-pack-145808</p>	<p>Terrain Tools Sample Asset Pack</p>
	<p>https://assetstore.unity.com/packages/3d/environments/landscapes/rpg-poly-pack-lite-148410</p>	<p>RPG Poly Pack - Lite</p>
	<p>https://assetstore.unity.com/packages/audio/soundfx/footstep-snow-and-grass-90678</p>	<p>Footstep(Snow and Grass)</p>
	<p>https://soundimage.org/fantasywonder/</p>	<p>Royalty free music and SFX</p>

14. Workstation and tools:

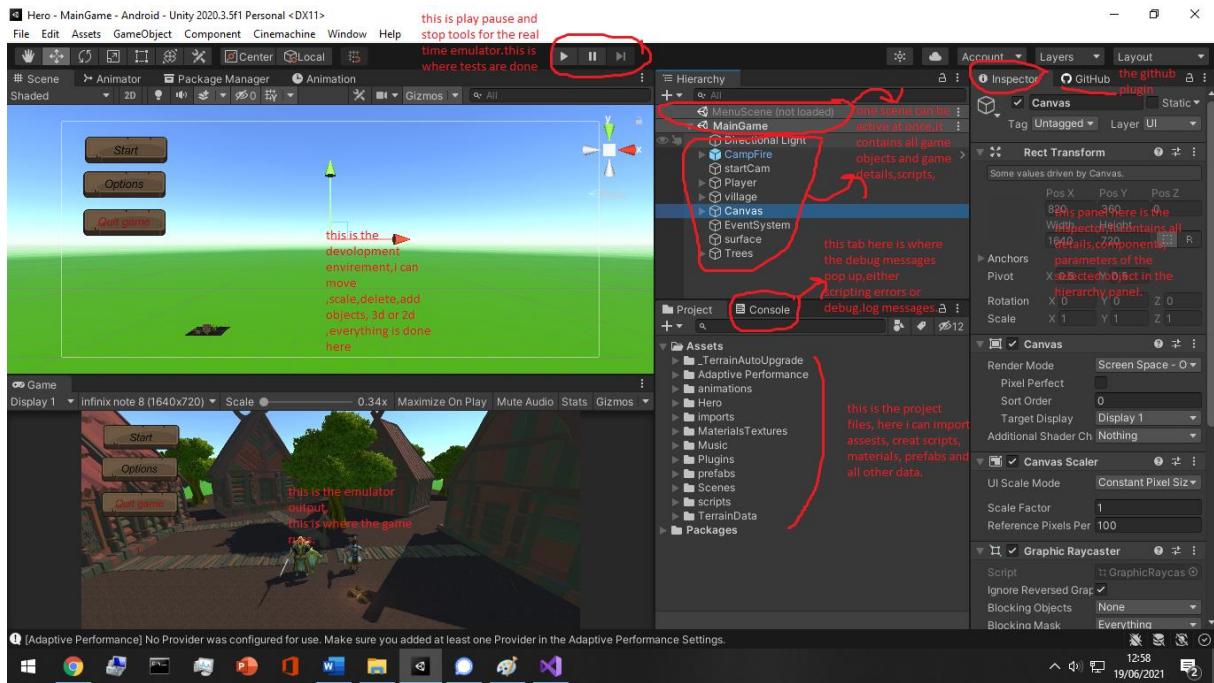
- Motherboard: H310M PRO-VD PLUS
- CPU: intel i5 9400F 2.9 GHz
- GPU: Nvidia GeForce GTX 1650
- Screen: 32" Vega Led
- Umlet: a powerful diagram designer and freeware.
- SketchBook: a freeware and easy to use image editor.
- Microsoft windows Paint.(who doesn't know Paint)
- Unity editor 2020.3.5f1 LTS
- Blender
- Microsoft office suite validated with the institute office 365 email aymen.ajroud@sousse.r-iset.tn .
- 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.

15. 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.

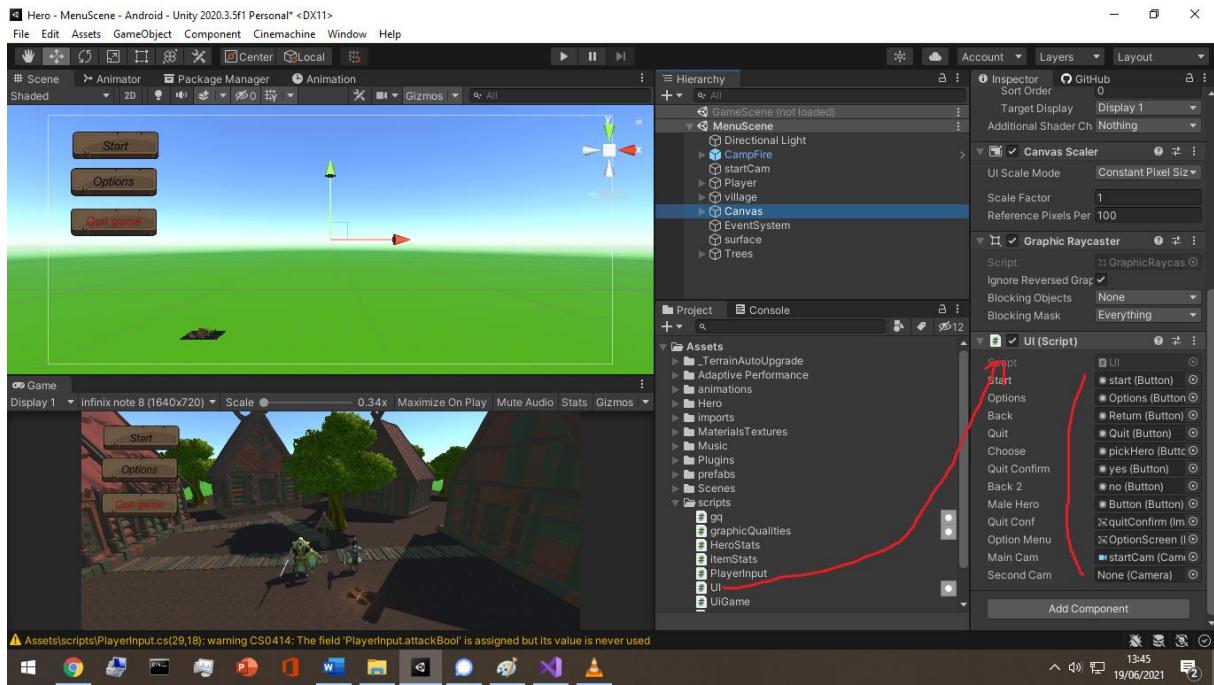
III. Production:

1. Introduction :unity editor



2. Introduction: Scripting with visual Studio

```
1  using System.Collections;
2  using System.Collections.Generic; //this is the imports section , all the unity built in references and libraries are declared here
3  using UnityEngine;
4
5  @Script Unity | Références
6  public class gg : MonoBehaviour
7  {
8      //the declaration of any method or variable is done here, inside the class,
9      //public or serialized variables can be seen inside the inspector when targeting the script or attaching it to a game object.
10     // Start is called before the first frame update
11     void Start()
12     {
13     }
14
15     // Update is called once per frame
16     void Update()
17     {
18     }
19
20     ;
21 }
22 }
```

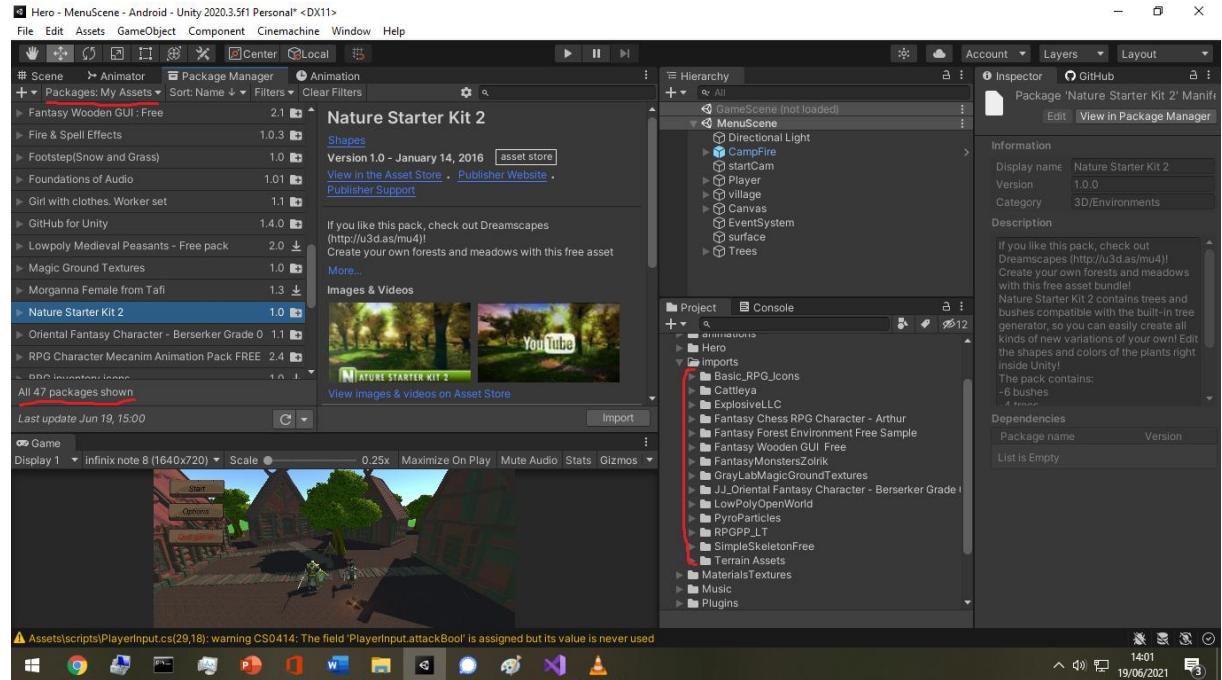


When declaring an object as a public variable inside a script, I need to drag and drop it in the inspector otherwise It gives a null pointer reference if I try to access an empty object.

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

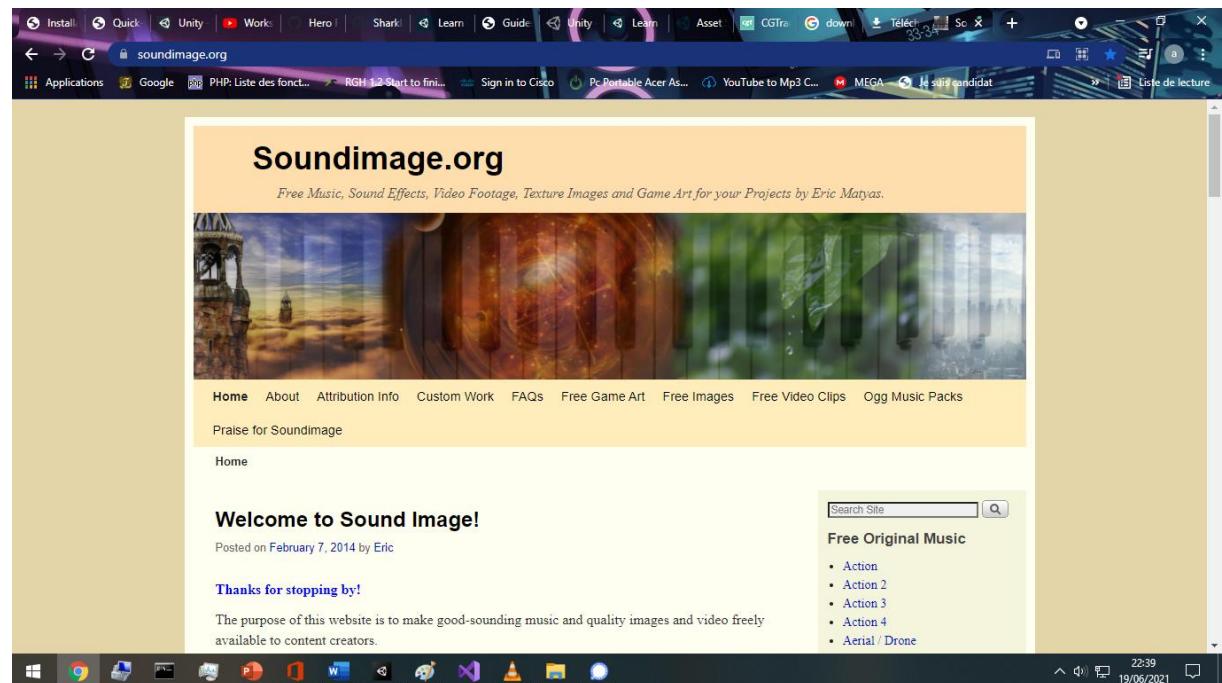
3. Assets: Package Manager

The built in unity 2020 packageManager can directly download and import assets from my owned assets related to my unity account.

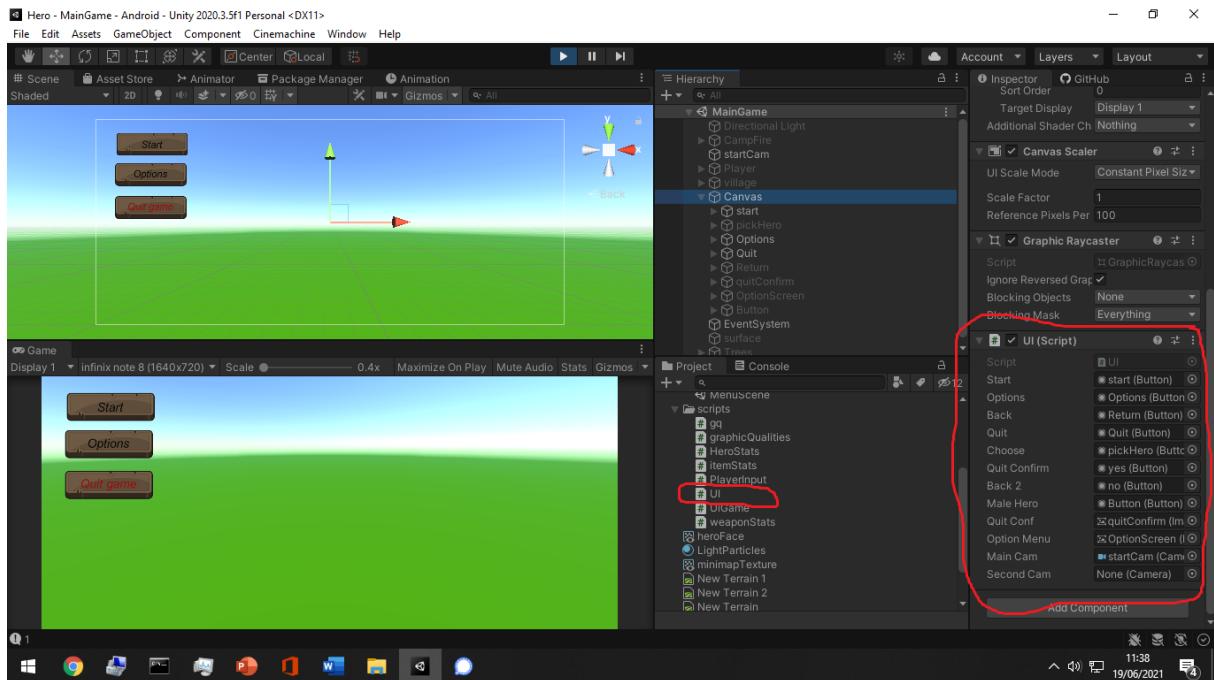


All assets are free to use commercially.

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



4. User Interface: The starting scene



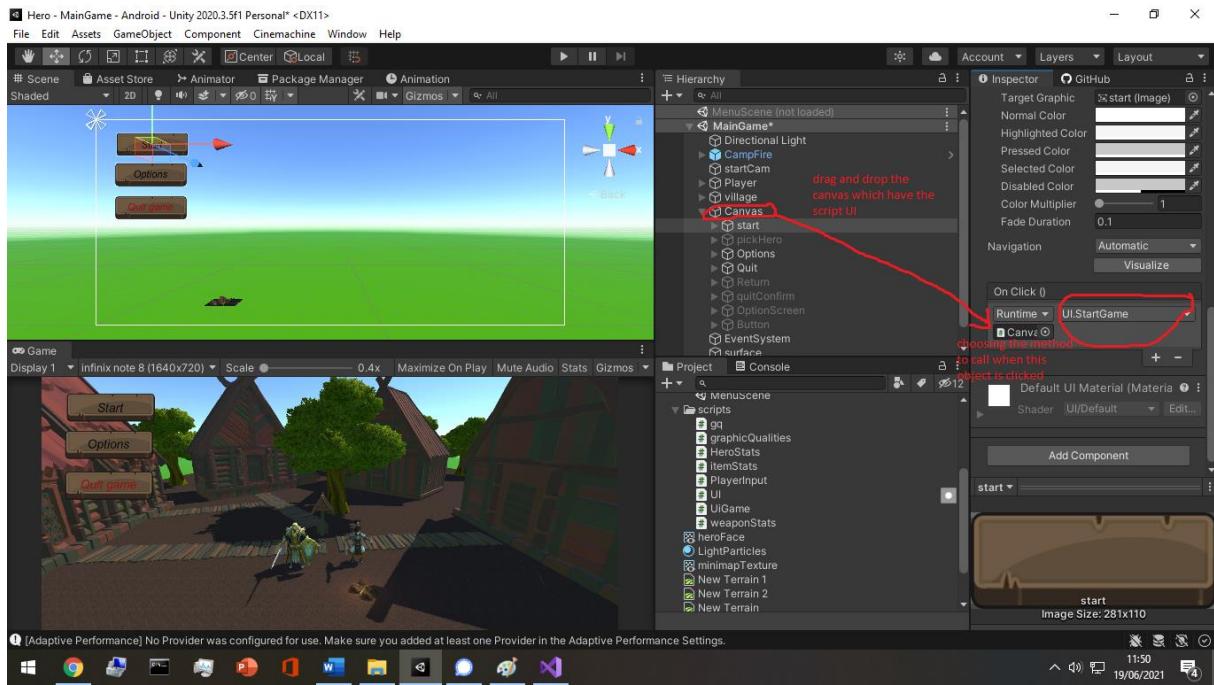
all the **canvas** children are attached and declared in **its** script as variables to do what ever is needed along the scripting.

```
① Script Unity | O références
public class UI : MonoBehaviour
{
    3 références
    public static string Hero { get; set; }
    [SerializeField] Button start, options, back, quit, choose, quitConfirm, back2, maleHero;
    [SerializeField] Image quitConf, optionMenu;
    [SerializeField] Camera mainCam, secondCam;

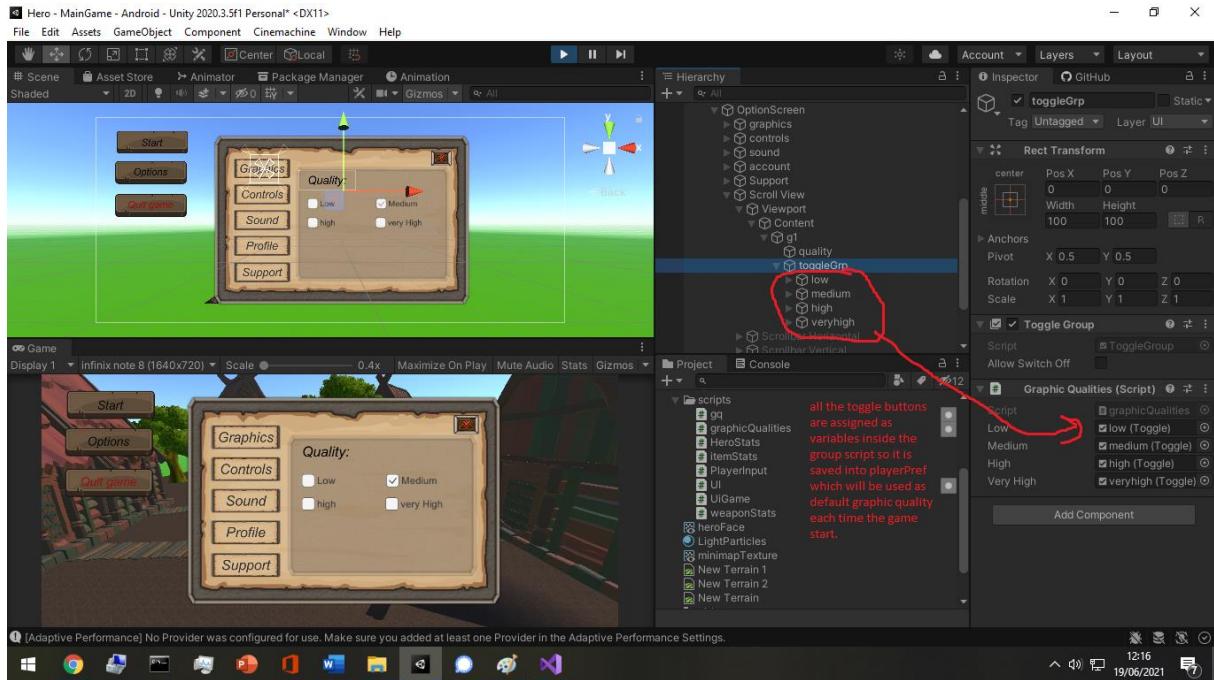
    private bool startGameBo;
    private Transform posCamStart;
    // Start is called before the first frame update
}
```

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.

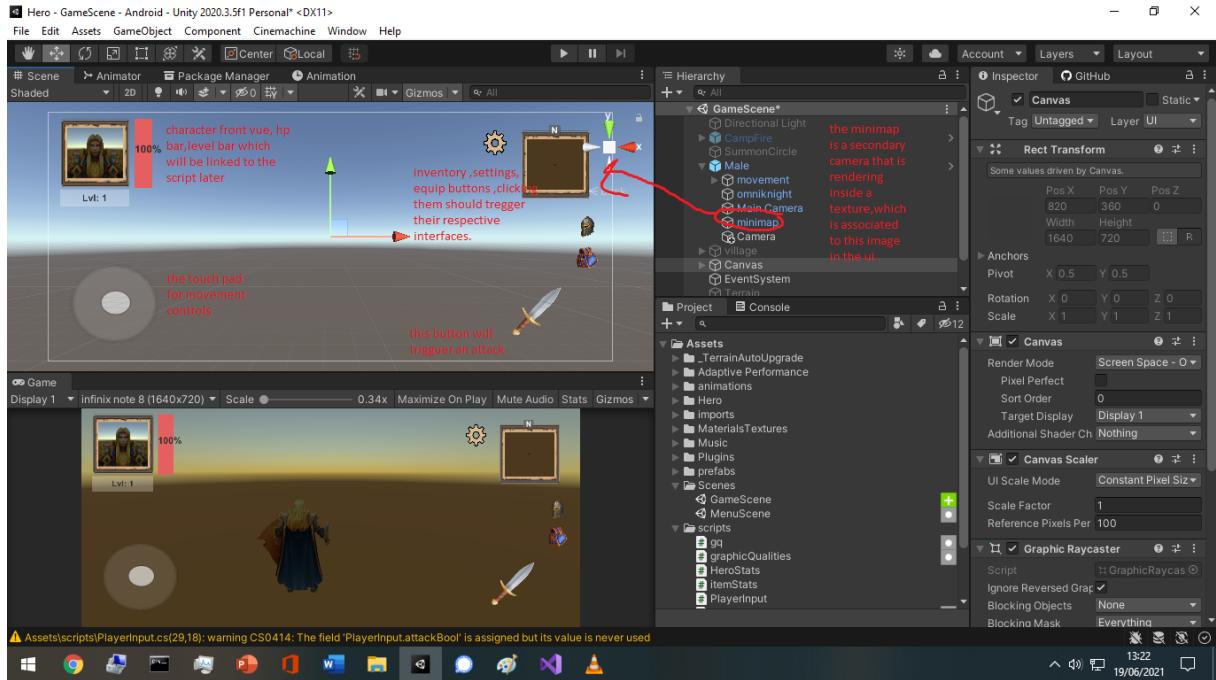
That's almost how ui objects event system works. All the treatments and the events are inside the called method in the onClick Panel.



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. This is almost the same concept for every other settings.



5. User Interface: The game scene



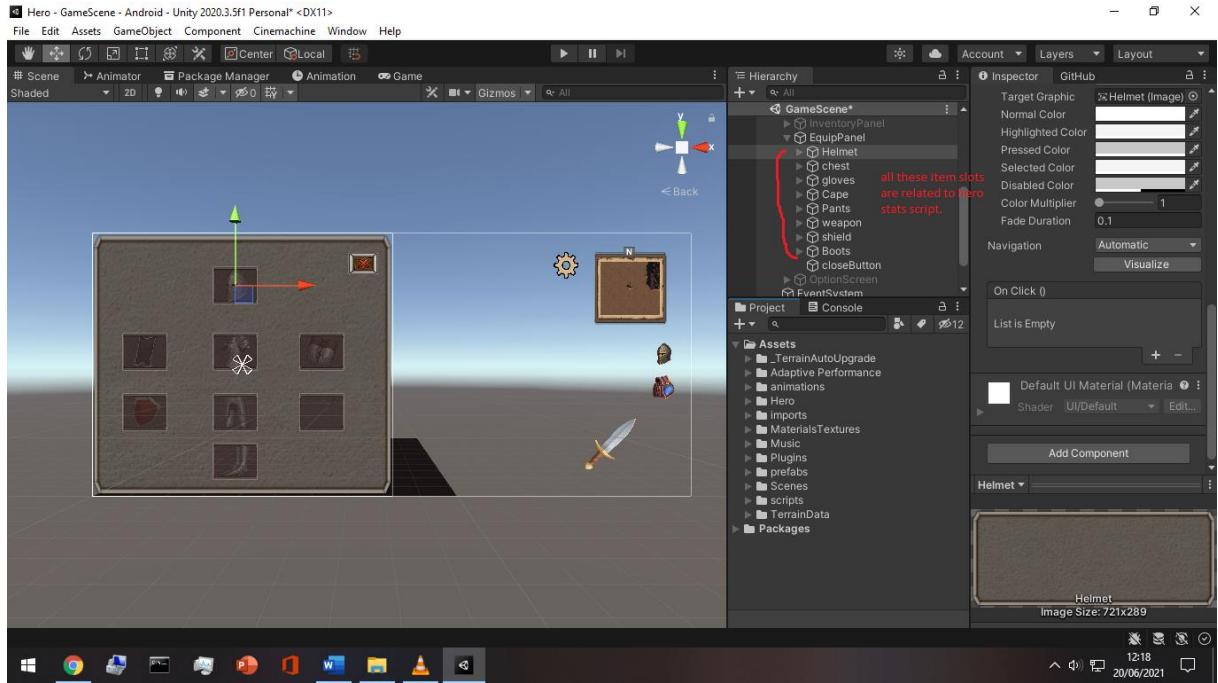
When building ui interface, I found that the order in the hierarchy matters a lot since ui objects in the upper order are rendered on top of the lower level ones.

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.

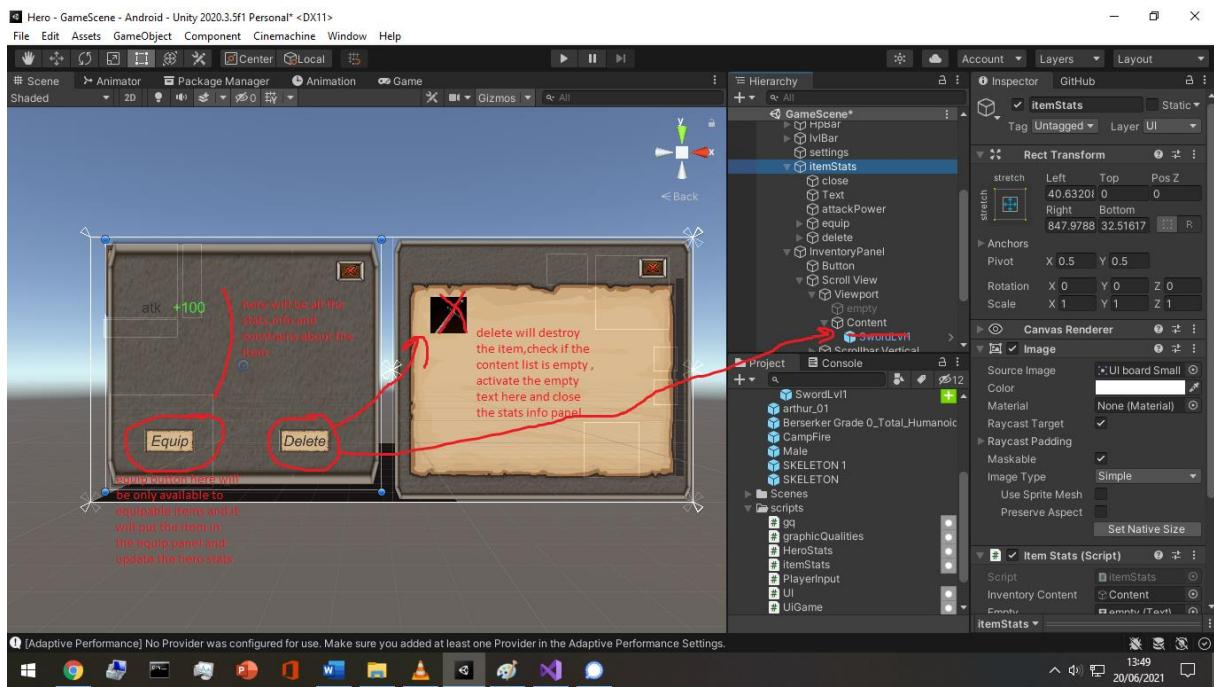
Update() is called every frame, and every device screen have a frame rate and is calculated as fps (frame per second).A frame is simply an image.

6. Inventory/equip system:



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.

The text for UI elements and text meshes can incorporate multiple font styles and sizes. Rich text is supported both for the UI System and the legacy GUI system. The Text, GUIStyle, GUIText and TextMeshPro classes have a Rich Text setting which instructs Unity to look for markup tags within the text.



Here is an example pf script.

```

23 public void ItemStats()
24 {
25
26     GameObject.Find("Canvas").GetComponent<UiGame>().ItemStatsOn();
27     GameObject.Find("itemStats").GetComponent<itemStats>().item = this.GetComponent<Button>();
28     atck = GameObject.Find("attackPower").GetComponent<Text>();
29
30     atck.text = "+" + attackPower.ToString();
31 }
32
33

```

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.

```

36     public void DestroyItem()
37     {
38         Destroy(item.gameObject);
39         if (inventoryContent.transform.childCount == 1) empty.gameObject.SetActive(true);
40         this.gameObject.SetActive(false);
41     }
42
43

```

Annotations in the code editor:

- Line 38: `Destroy(item.gameObject);`
- Line 39: `if (inventoryContent.transform.childCount == 1) empty.gameObject.SetActive(true);`
- Line 40: `this.gameObject.SetActive(false);`
- Line 43: `ActivatesDeactivates the GameObject, depending on the given true or false/ value.`

This is called when the destroy item is clicked.

7. Touch input for smartphones:

Smartphone devices can track 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.

8. Player Input:

```
for (int i = 0; i < Input.touchCount; i++)
{
    Touch touch = Input.GetTouch(i);
    if (touch.phase == TouchPhase.Began)
    {
        touchStart = touch.position;
        if (touch.position.x <= width / 2 && touch.position.y < height) phase Began
        {
            movePos.position = touch.position;
            moveBool = true;
        }
        else if (touch.position.y >= height) rotateCamBool = true;
    }
    else if (touch.phase == TouchPhase.Moved || touch.phase == TouchPhase.Stationary)
    {
        touchEnd = touch.position;
        if (moveBool) phase moved/stationary
        {
            x1 = (touchEnd.x - touchStart.x) / (width / 8);
            y1 = (touchEnd.y - touchStart.y) / (height / 4);
            if (touchEnd.x < touchPadPanelRect.rect.xMax * 2 && touchEnd.x > touchPadPanelRect.rect.xMin * 2 && touchEnd.y > touchPadPanelRect.rect.yMin * 2 && touchEnd.y < touchPadPanelRect.rect.yMax * 2)
            {
                if (rotateCamBool)
                {
                    if (touch.phase == TouchPhase.Stationary)
                    {
                        x2 = 0;
                        touchStart = touchEnd;
                    }
                    else x2 = touchEnd.x - touchStart.x;
                }
            }
        }
        else if (touch.phase == TouchPhase.Ended) phase ended
        {
            if (moveBool)
        }
    }
}
```

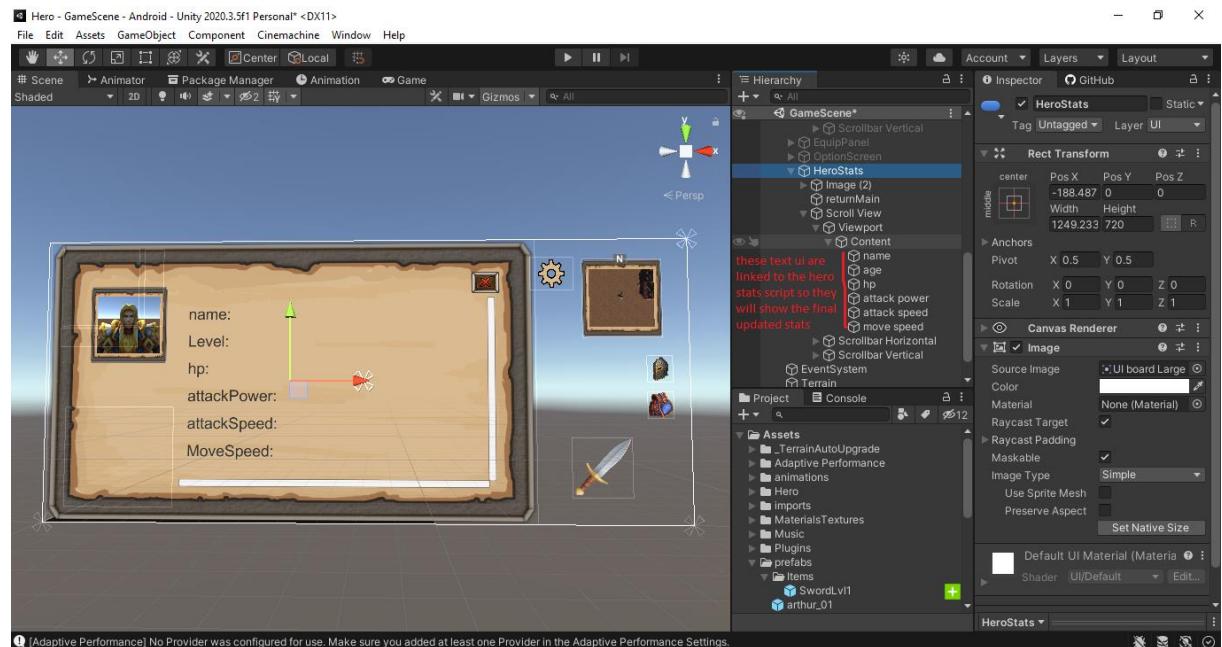
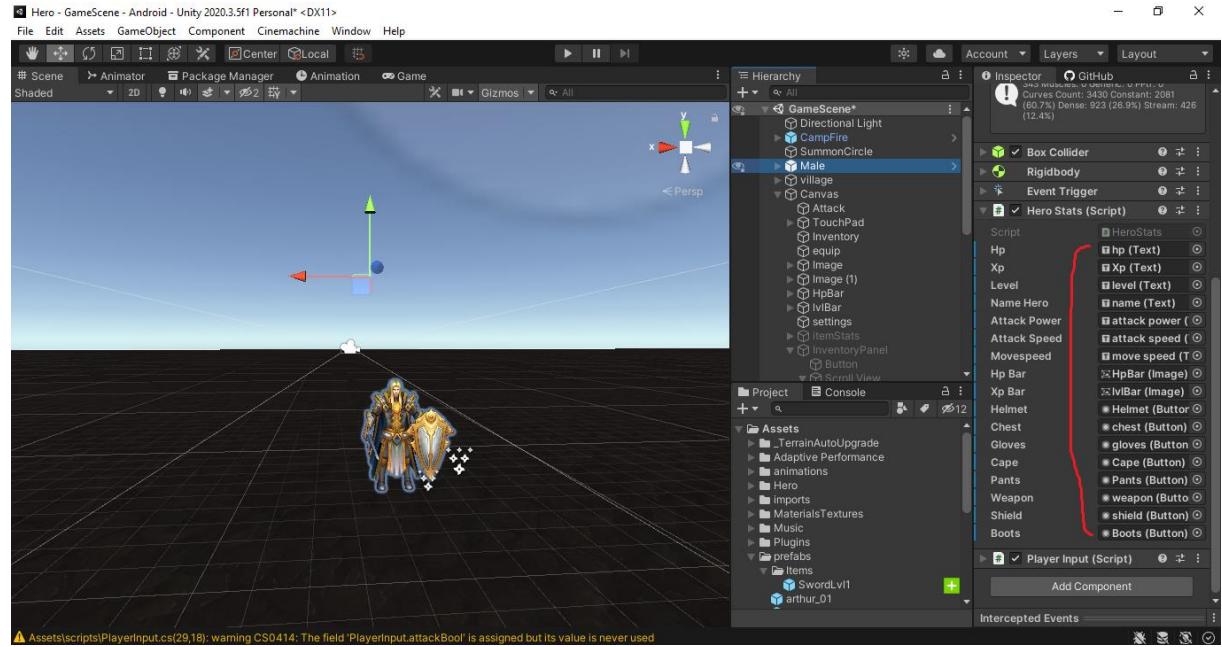
When dealing with player moving object, animations and FX either visual or sound.

```
private void HeroWalk(float a)
{
    if (a != 0) //this method here will pass parameter to the animation component when needed
        hero.SetBool("walk", true);
    else hero.SetBool("walk", false);
}
```

```
public void Attack()
{
    if (timer > 2) //this timer system is implemented to prevent attack spamming, this is basicly the attack speed and it will be a variable for future manipulations
    {
        hero.Play("arthur_attack"); //this method here will pass parameter of attacking to the animation component
        sword.Play();
        shield.Play(); // these sword and shield are particles that are played on the moment of attacking.particle is a VFX
        attackBool = true;
        timer = 0;
    }
    else attackBool = false;
}
```

9. Hero stats:

The hero object will have heroStats.cs script attached to it and from there all ui texts will be referenced here.



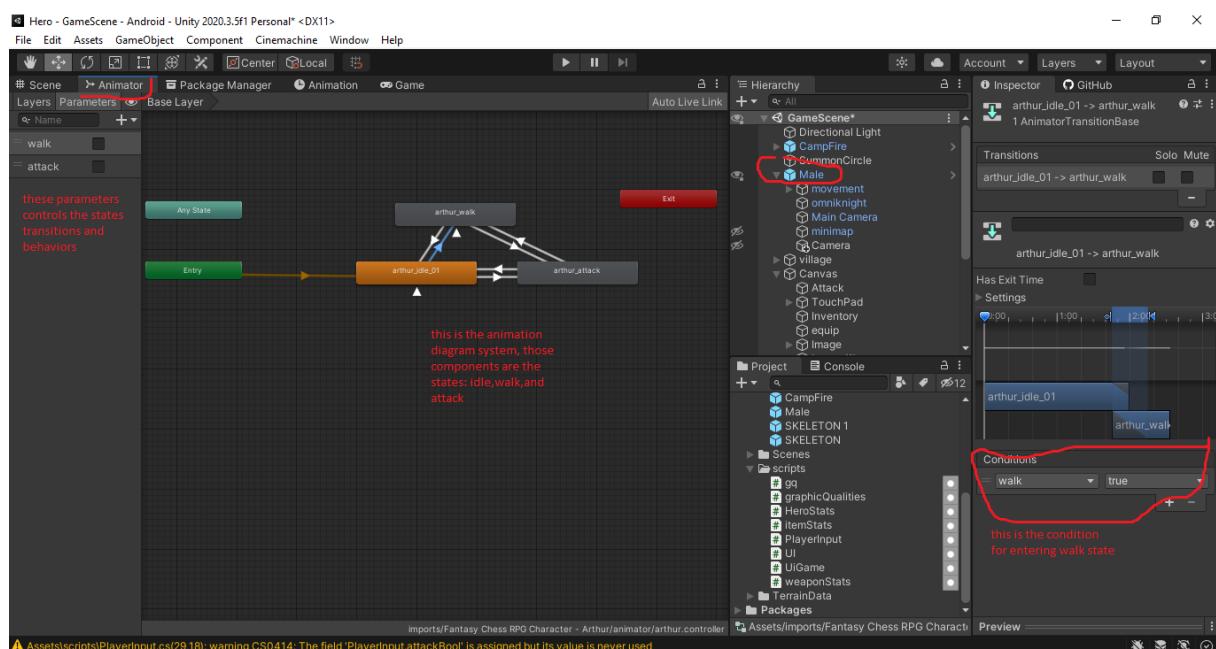
10. Unity Mechanim:

Unity has a rich and sophisticated animation system called Mecanim. Mecanim provides:

- Easy workflow and setup of animations on humanoid characters.
- Animation retargeting - the ability to apply animations from one character model onto another.
- Simplified workflow for aligning animation clips.
- Convenient preview of animation clips, transitions and interactions between them. This allows animators to work more independently of programmers, prototype and preview their animations before gameplay code is hooked in.
- Management of complex interactions between animations with a visual programming tool.
- Animating different body parts with different logic.

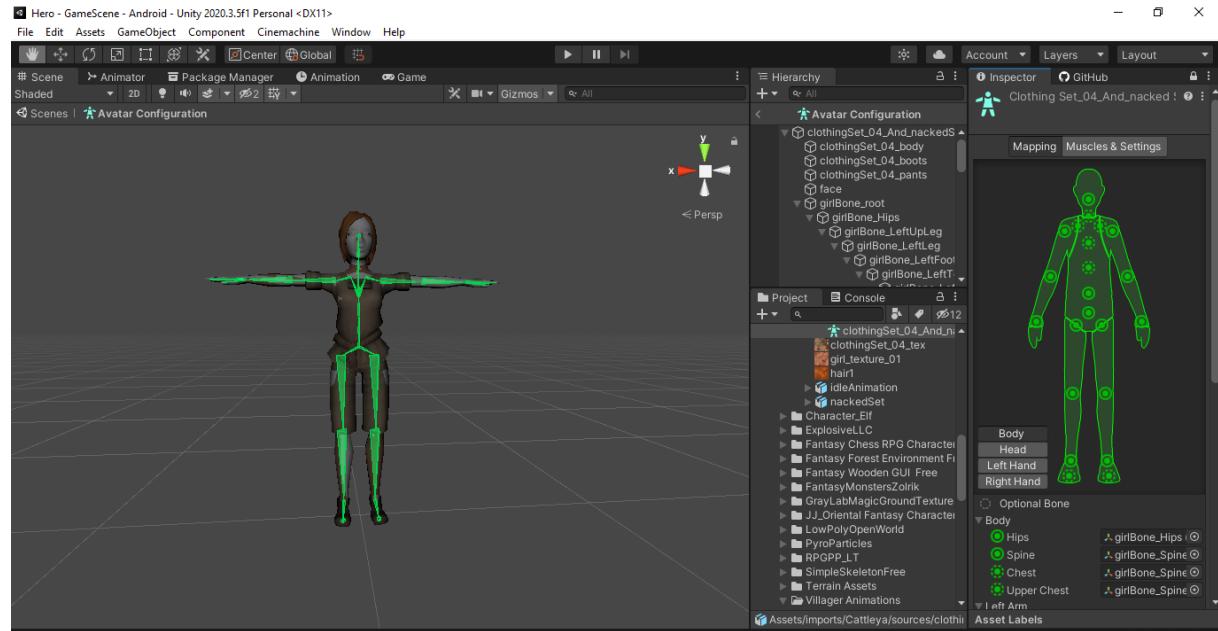
11. Unity Rigging:

The Animation **Rigging** package for **Unity** enables users to set up procedural motion on animated skeletons at runtime. This makes it possible to do powerful things during gameplay such as world interactions, skeletal deformation **rigging**, and physics-based secondary motion. Most of the free assets are “rigged” and ready for mechanim.

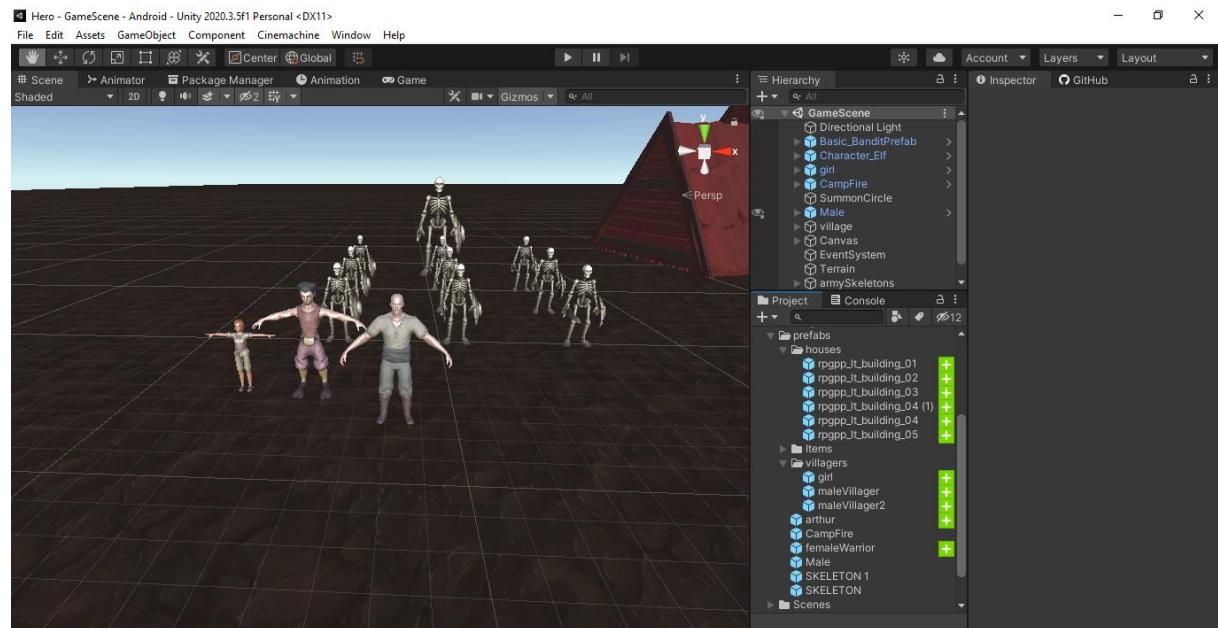


This is unity Mecanim tool for controlling animation flow.

And this one below for rigged humanoid characters.

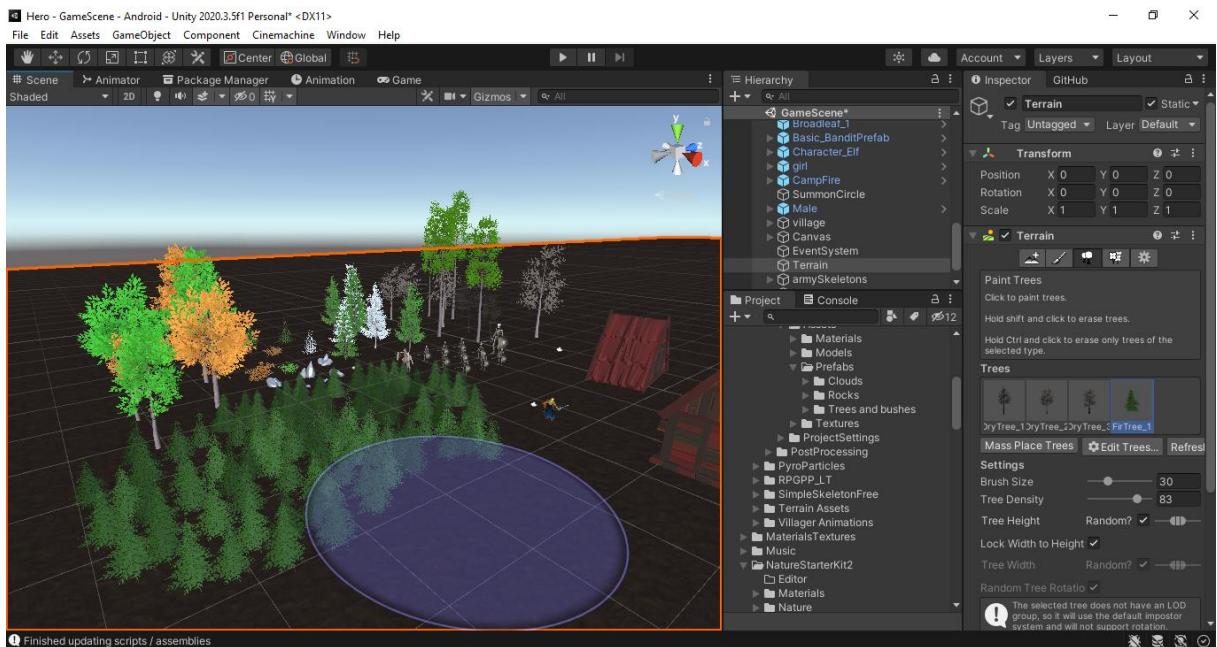
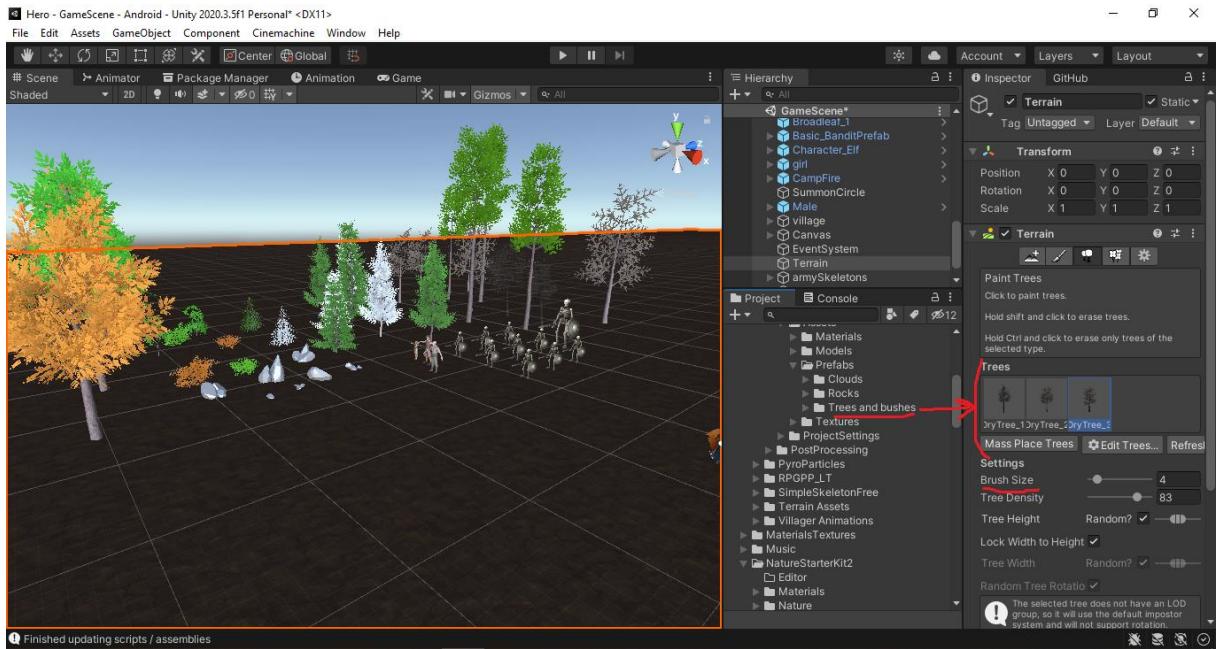


12. NPC prefabs:



Unity's Prefab system allows you to create, configure, and store a GameObject complete with all its components, property values, and child GameObjects as a reusable Asset. The Prefab Asset acts as a template from which you can create new Prefab instances in the **Scene**.

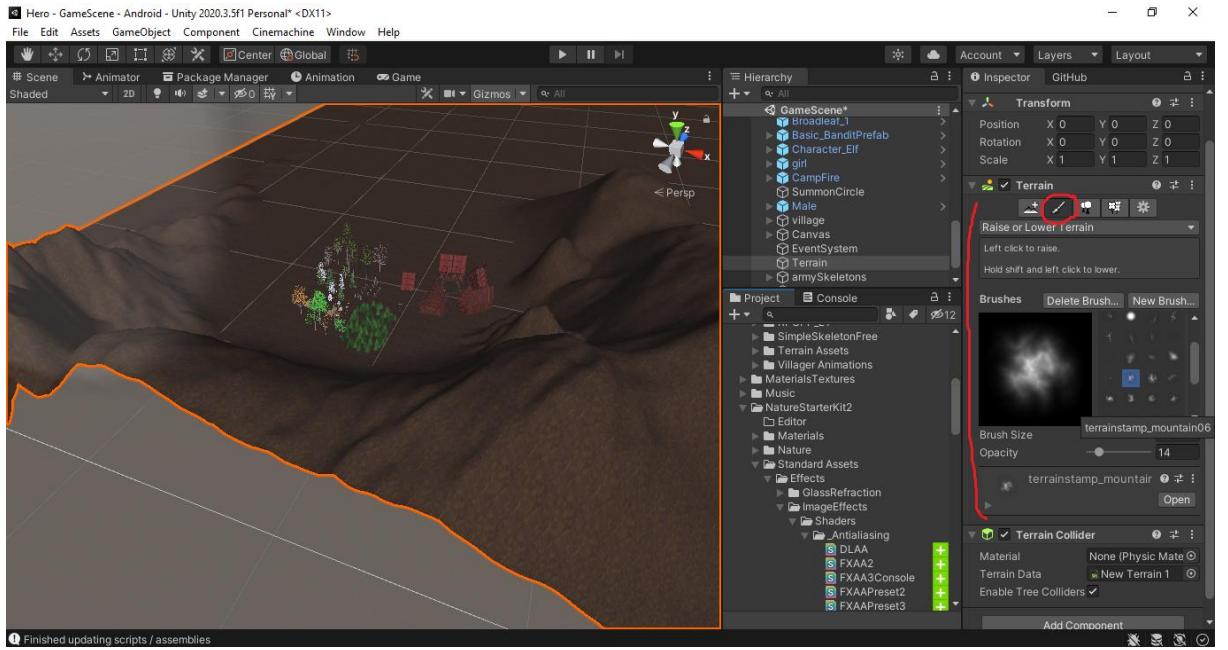
13. Terrain building tools:



The terrain tool panel makes building terrain vegetation very quick with the use of a brush that have a dynamic size from 1 tree to hundreds.

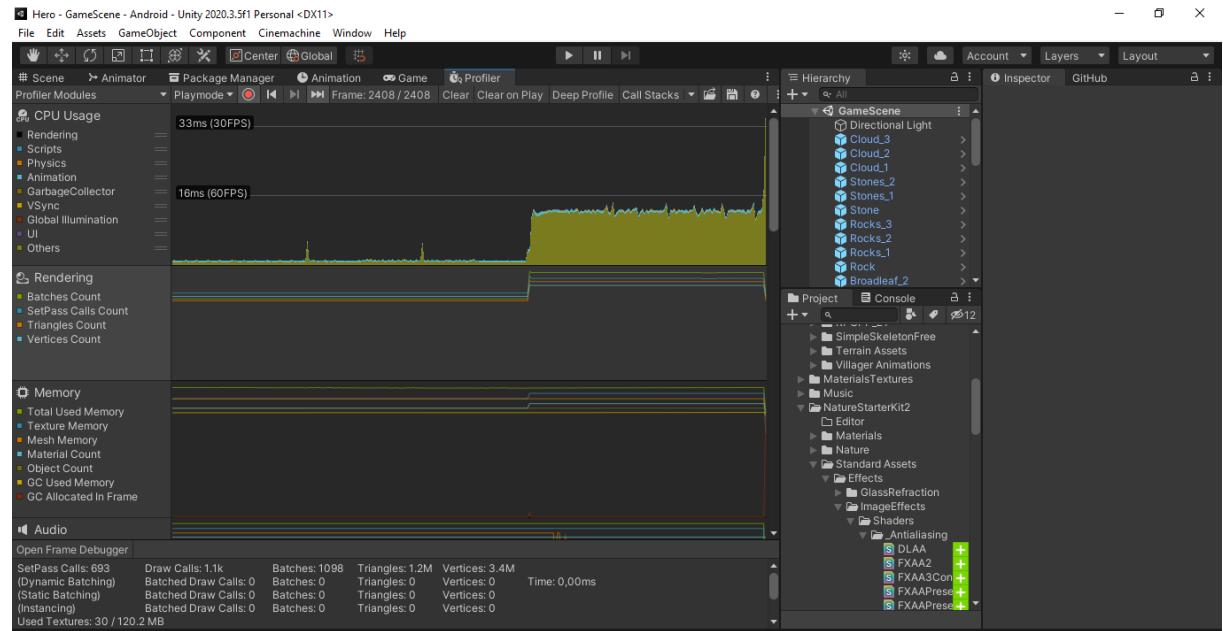
There is also the terrain painting:

- Raise or lower terrain to build rivers, mountains, hills.
- Paint holes(remove part from terrain).
- Paint textures (dirt, snow...).



IV. Post-production:

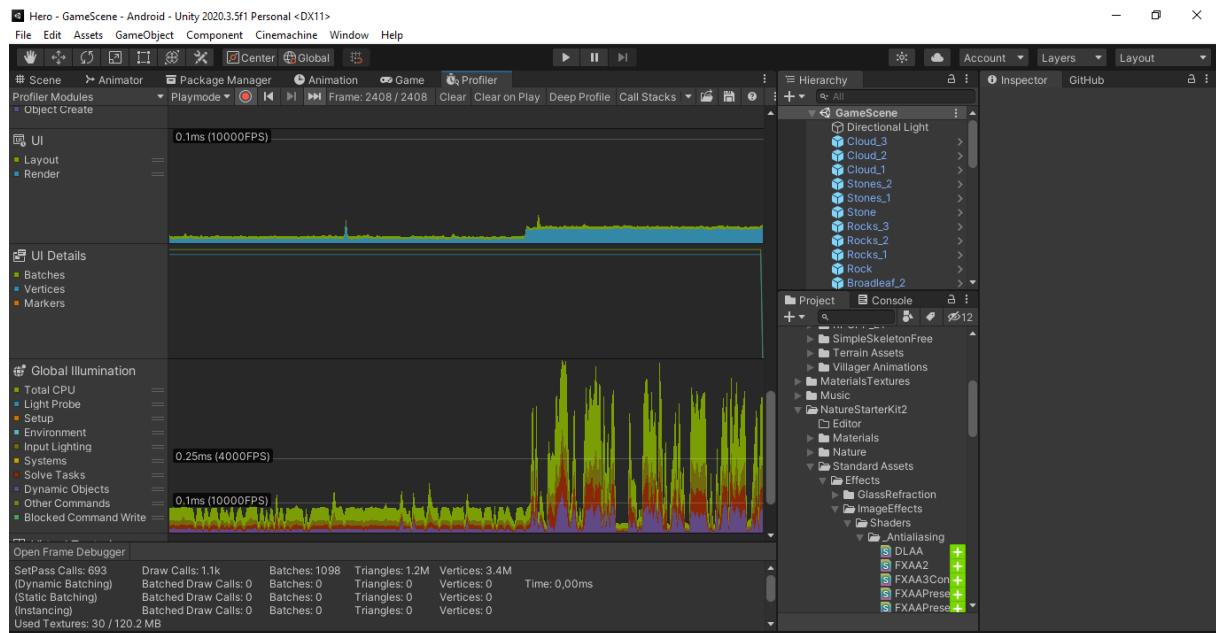
1. Unity Profiler:



The Unity Profiler is a tool you can use to get performance information about your application. You can connect it to devices on your network or devices connected to your machine to test how your application runs on your intended release platform.

You can also run it in the Editor to get an overview of resource allocation while you are developing your application.

The Profiler gathers and displays data on the performance of your application in areas such as the CPU, memory, renderer, and audio. It's a useful tool to identify areas for performance improvement in your application and iterate on those areas. You can pinpoint things like how your code, assets, scene settings, camera rendering, and build settings affect your application's performance. It displays the results in a series of charts, so you can visualize where spikes in your application's performance happen.



2. Optimization and code review:

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 repeatedly without destroying them).
- Coroutines/update (update will happen every frame, so coroutines sometimes reduce a lot of work).
- Caching 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 are not using something, turn it off.
- in general, you should use the minimum amount of light necessary to achieve your desired style.
- Enable Instancing option within unity.

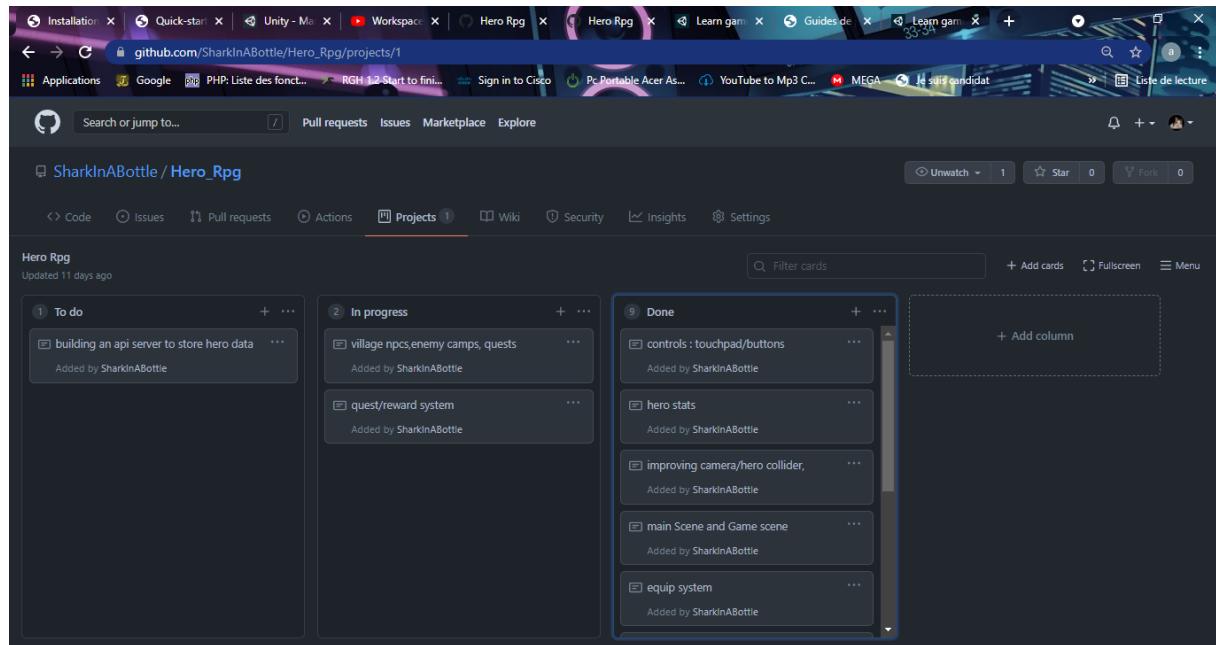
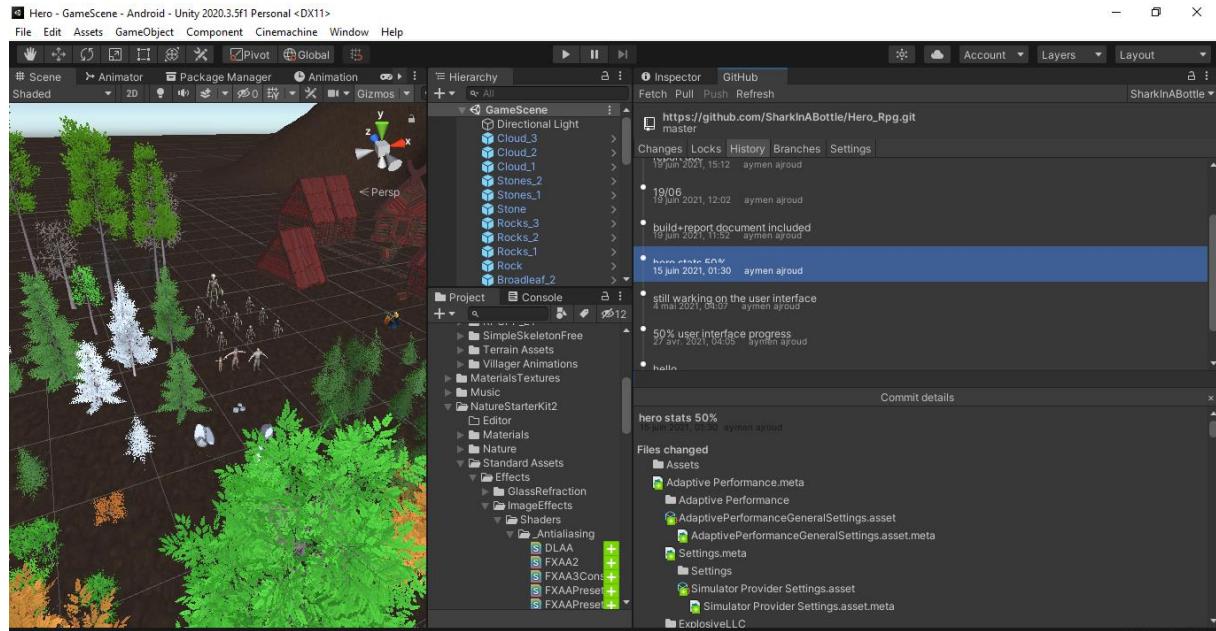
3. Bug fixing:

The game will have bugs. Every game has bugs and this one will not be an exception. The question is, how do you deal with bugs. If you know your bugs are getting out of hand it might be a good idea to stop working on new features and focus on stabilizing the build until things are back on track.

If you identify a system that is constantly causing problems, it might be worth the time to rebuild it instead of repeatedly fixing the issues it causes.

4. Version Control and data safety:

Even today, with the availability of free version control systems (VCS), there are still teams out there working with shared folders or drop boxes to collaborate. Developing without version control is a bit like driving without a seat belt. Wearing a seat belt might not always be comfortable and will restrict you a bit, but once you crash full speed into that tree, it will save your life.



5. Polish:

Particles, VFX and SFX help boost the game quality and improve the gameplay .

V. Operations:

1. Google/iOS developer account:

Google developer account requires 25\$ for a lifetime subscription to google play store while apple require 99\$ for 1 year.

The screenshot shows the 'Créer un compte de développeur' (Create developer account) page in the Google Play Console. At the top, there's a header with the Google Play logo and the email address 'ajroudaymen@gmail.com'. Below the header, the title 'Créer un compte de développeur' is displayed. A note states: 'Ce nouveau compte de développeur sera associé au compte Google sélectionné. Si vous essayez de rejoindre un compte de développeur existant, veuillez contacter votre administrateur pour faire une demande d'invitation.' A note below it says: 'Si vous représentez une organisation, nous vous conseillons de ne pas utiliser un compte personnel pour configurer des comptes de développeur. Vous pouvez configurer des comptes Google avec une adresse e-mail existante. [En savoir plus](#)'.

A callout box contains the text: 'Pour créer un compte, vous devez vous acquitter des frais d'enregistrement de 25 \$ (payables une seule fois). Pour finaliser l'enregistrement de votre compte, vous devrez peut-être valider votre identité à l'aide d'une pièce d'identité en cours de validité. Si votre identité ne peut pas être validée, les frais d'enregistrement ne vous seront pas remboursés.'

Below the note, there's a field labeled 'Nom de développeur public *' with a placeholder 'Les utilisateurs de Google Play verront ce nom de développeur' and a character limit of '0/50'.

The screenshot shows the 'Apple Developer Program Enrollment' page. At the top, there's a navigation bar with links for 'Discover', 'Design', 'Develop', 'Distribute', 'Support', and 'Account'. The user is signed in as 'aymen ajroud'. The main section is titled 'Confirm your personal information.' It asks for legal name, phone number, and address. The 'First Name' field contains 'Given name', and the 'Last Name' field contains 'Family name'. The 'Phone Number' field has a dropdown menu for country code ('+') and a text input field for the number. Below the phone number fields are 'Intl. Code', 'Phone Number', and 'Extension' labels. The 'Address' field is a large text input area.

2. Unity monetizing:

Monetization of your Unity project through ads is a great way to generate revenue without charging your customers directly. A successful free ad-supported game will bring in many times the asking price of an ad-free game that must be purchased outright. Though Unity Ads is one of the Unity Services, it is not in the default Unity package and must be downloaded from the Asset Store.

When you first enable Unity Services for a project, you are asked to assign the project to an organization. By default, your account has one organization: your username. Organization names must be unique. You can add (but not currently delete) more names, or rename an account at will, as long as the proposed new name is available. You might wish to have one organization for your solo projects, and one for your team.

Ads are shown in designated placement zones in your project. Rather than a literal scene or GameObject, a placement zone is just a mode defining how your players can interact with ads. Options include muting the audio and if or when they can skip the ad. For example, you would not want players to have the ability to skip a lucrative rewarded video. All projects in the Dashboard begin with two placement zones, video and rewarded Video, for regular and incentivized video ads.

3. Reviews and updates:

The Google Play Developer Reply to Reviews API allows you to view user feedback for your app and reply to this feedback. You can use this API to interact with users directly within your existing customer support toolkit, such as a CRM system.

The Reply to Reviews API allows you to access feedback only for production versions of your app. If you want to see feedback on alpha or beta versions of your app, use the Google Play Console instead. Also, note that the API shows only the reviews that include comments. If a user rates your app but does not provide a comment, their feedback is not accessible from the API.

4. Updates and continued maintenance:

Based on game reviews, further concept and design review and updates for the game are made and must go through the production cycle again.

VI. Conclusion:

Looking back, I am very happy that I chose this path. The project structure taught me a lot and my failures allowed me to improve as a developer.

No doubt I consider this game a success and I am more than proud about what I created, but as always, I believe I could have done better.

I now know how to shift attention during development and manage time in a more productive way.

I am looking forward to building more games and learning a lot more.