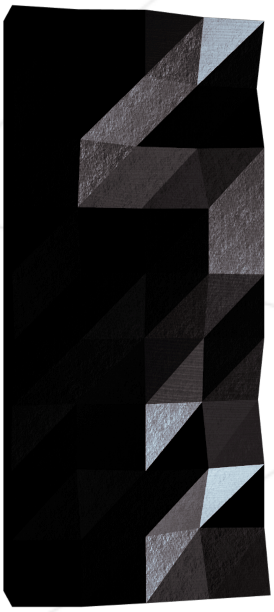


Unity Piscine - Module02

2D environment, tiles and sprites

Summary: In this document, you will find the Module02 subject of the Unity Piscine.

Version: 1.00

Contents

Chapter I

Instructions

- If you have problems installing the tools needed for your project on the 42 computers, you can use a virtual machine. In this case, you will have to :
  - install the virtual machine software on your computer.
  - install the operating system of your choice.
  - install the tools needed for your project.
  - Make sure you have the space on your session to install all of this.
  - You must have everything installed before the evaluation.
- Only this page will serve as reference. Do not trust rumors.
- The exercises have been ordered from easiest to most difficult. Under any circumstance you can submit or take into account an exercise if a previous one has failed.
- Be careful with the access rights of your files.
- You should follow the submit procedure for all you exercises.
- Your exercises will be corrected by your piscine peers.
- You cannot leave any extra file on your repository except the ones explicitly specify on you subject.
- Got a question? Ask your peer on the right. Otherwise, try your peer on the left.
- Everything you need can be found on the `man` or out there on Google.
- Read carefully the exercises: they may contain some features you should implement that are explicitly mentioned on the subject.
- Use your brain!!!

Chapter II

Day-specific rules

This Module02 is important for the next module. Here you create the basic elements that will be useful in Module03.

Chapter III

Foreword

Today we will create the basic elements of your future game.
To do his, go to Unity Asset Store and import the Free Pixel Art Overworld Tileset. In this Module we advice you to use a `GameManager` and start using the `Tags`. They will be very useful for this and future modules.

Chapter IV

Exercise 00: I see the world in 2D

	Exercise :
Turn-in directory : `unityModule02`	
Required elements : `"Map1" scene and anything useful`	
Forbidden functions : `None`	

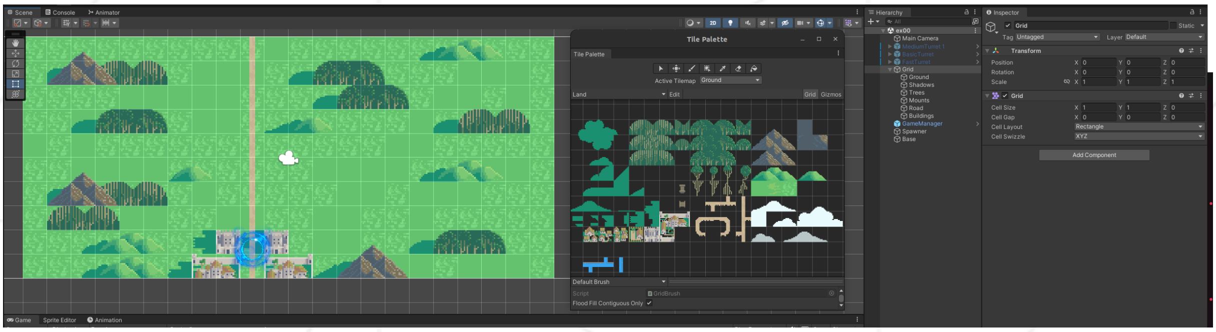
To start Create a new Module02 2D project.

So! With your new tiles you have imported, you will start by creating a `Tile palette` that will be very useful. Your palette should be saved in a new folder called TilesPalettes in your Assets folder. You must choose a consistent name.

Mow that you have your palette, you can start creating your map.
Your map must contain :

- The ground.
- The road.
- Trees, mountains or whatever you want to decorate.

it should look something tike this :



Active Tilemap: Ground

Default Brush

Script: GridBrush

Flood Fill Contiguous Only ☑

ex00
- Main Camera
- MediumTurret 1
- BasicTurret
- FastTurret
- Grid
  - Ground
  - Shadows
  - Trees
  - Mounts
  - Road
  - Buildings
- GameManager
- Spawner
- Base

☑ Grid □ Static

Tag: Untagged Layer: Default

Transform

	X	Y	Z
Position	0	0	0
Rotation	0	0	0
Scale	1	1	1

☑ Grid

	X	Y	Z
Cell Size	1	1	0
Cell Gap	0	0	0

Cell Layout: Rectangle

Cell Swizzle: XYZ

For the first map your road must be very simple. Is on this road that the enemies will move.

Chapter V

Exercise 01: White Walkers

	Exercise :
Turn-in directory : `unityModule02`	
Required elements : `"Map1" scene and anything useful`	
Forbidden functions : `None`	

- `Spawner: in video game a spawner is an object in a simulated game world that spawns further objects.`
- `HP : HP is the total of health point of an entity.`

Now we need enemies !
Because the animation is not our subject today, you must be create a very simple enemies with a basic sprite.
An white capsule will do very well. You must have only one script for all yours enemies.

You will also have to create a spawner. The enemy will have to spawn at the top of the map and will have to move to the bottom.
Make juste a simple spawner for now, make a fixed delay between spawns.

You must place your base represented by a circle at the bottom of the map and when enemy arrive on it, the base loses HP. An enemy make one damage.

For now your base will have 5 HP.
Display on the console the remaining HP of your base every time an enemy gets in.
When your base goes down to 0 HP, you should display Game Over in the console and enemies should not spawn anymore and all those on the map must be destroyed.

In any case, your enemy will always have to be destroyed if he enter the base, leave the map, is killed or if is game over.

Chapter VI

Exercise 02: Arms!

	Exercise :
Turn-in directory : `unityModule02`	
Required elements : `"Map1" scene, EnemyController script, TurretController script and anything useful`	
Forbidden functions : `None`	

Organize your defenses !
In the assets provided for this module, you will find a proposal for turrets in Prefabs folder, but, you can decide to create it by yourself.

For the bullet's, you can create them yourself.
A simple circle sprite or whatever you want will do.

You must have 3 different turrets for the moment:

- A turret has a low rate of fire with basics damages of 0,3.
- A turret has a medium rate of fire with basics damages of 0,2.
- A turret has a very fast rate of fire with basics damages of 0,1.

Your turrets must have a detection zone and when an enemy enters in this zone, the turret will have to target him and shoot him. She always target the closest enemy.

Turrets should not have too much range, and should not be placed far from the road to be able to detect the enemies. If they are placed far from the road they should not be able detect the enemies.

You must have only on scripts for all the turrets.

When the bullet has hit an enemy, it is destroyed and the enemy takes damages.

Your enemy will have 3HP and a bullet does one damage. Don't forgot to take into account the basic damage of your turret.

If your enemy is defeated, it will be destroyed.

`As you will have to reuse your work in the next module, don't forget to export your assets so that you can import them into module03.`

Chapter VII

Submission and peer-evaluation

Turn in your assignment in your `Git` repository as usual. Only the work inside your repository will be evaluated during the defense. Don't hesitate to double check the names of your folders and files to ensure they are correct.

You should not put all the files of a project on git, otherwise the disk space occupied by the repository will be unnecessarily increased. Here is how to configure Unity and GIT for an optimal use.

- Make sure that Unity saves as many files as possible in text form instead of binary.In Unity, go to Edit >Project Settings > Editor. Under texttttAsset Serialization, you have to choose the Force Text Mode.
- check that the .gitignore file automatically generated by unity is present.

The evaluation process will happen on the computer of the evaluated group.