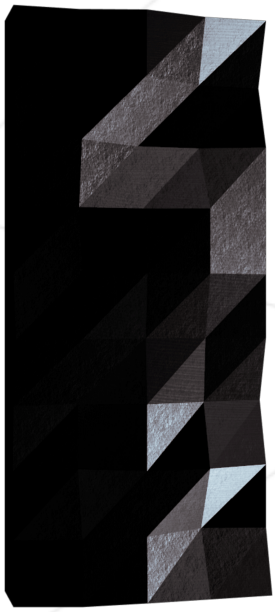

Piscine Unity - Day 09

First Person Shooter

Staff staff@staff.42.fr

Summary: This document contains the subject for Day 09 for the „Piscine Unity" from 42

Contents

I	Foreword	2
II	General Instructions	3
III	Specific instructions of the day	5
IV	Exercise 00 : Armed and dangerous	6
V	Exercise 01 : This ain't no place for no hero	7
VI	Exercise 02 : Endless Arena	9
VII	Exercise 03 : Who's the boss	10

Chapter I

Foreword

Yes it's already the end of the bootcamp. I hope you learned a lot of things and that you were able to do cool and nice games. No videos today, but a last super cool game before reaching the climax of the rush! Good luck everybody!

Chapter II

General Instructions

- The Unity bootcamp has to be made entirely, exclusively and mandatorily in `C#`. No `Javascript/Unityscript`, `Boo` or any other horrors.

- The use of functions or namespace not explicitely authorised in the exercise header or ini the rules of the day will be considered cheating.

- For a optimal usage of Unity, you have to work on `~/goinfre`, which is on the local drive of your computer. Remember to make appropriate backup on your own, the local goinfre can be purged.

- Unlike any other bootcamps, each day doesn't require a folder `ex00/`, `ex01/`, ..., `exn/`. Instead you'll have to submit your project folder which will be name like the day: `d00/`, `d01/`, ... However, a project folder, by default, contains a useless folder: the `"projet/Temp/"` sub folder. Make sure to **NEVER** try to push this folder on your repository.

- In case you're wondering about it, there is no imposed norme at `42` for `C#` during this bootcamp. You can use whatever style you like without restrictio. But remember that code that can't be read or understood during peer-evaluation is code that can't be graded.

- You must sort your project's assets in appropriate folders. For every folder correspond one and only one type of asset. For exemple: `"Scripts/"`, `"Scenes/"`, `"Sprites/"`, `"Prefabs/"`, `"Sounds/"`, `"Models/"`, ...

- Make sure to test carefully prototypes provided every day. They'll help you a lot in the understanding of the subject as well as what's requested of you.

- The use of the Unity Asset Store is forbidden. You are encouraged to use the daily provided assets (when necessary) or to look for additional ones on the Internet if you don't like them, exception made of scripts obviously because you have to create everything you submit (excluding scripts provided by the staff). The Asset Store is forbidden because everything you'll do is available there in one form or another.

However the use of Unity Standard Assets is authorised and event advised for some exercises.

- From d03 for peer-evaluation you'll be required to build the games to test them. `The corrector` will have to build the game, you must therefore always push projects/sources. You project must always be properly configured for the build. `No` last minute tweaks will be tolerated.

- Warning: You'll not be corrected by a program, except if stipulated in the subject. This imply a certain degree of liberty in the way you can do exercises. However keep in mind the instructions of each exercise, don't be LAZY, you would miss a lot of very interesting things.

- It isn't a problem to had additional or useless files in your repository. You can choose to separate your code in different files instead of one, except if the exercise's header stipulate a list of files to submit. One file must define one and only one behaviour, so no namespace. Those instructions don't apply to the `"projet/Temp/"` sub-folder which isn't allowed to exist in your repositories.

- Read carefully the whole subject before beginning, really, do it.

- This document could potentially change up to 4 hour before submission.

- Even if the subject of an exercise is short, it's better to take a little bit of time to understand what's requested to do what's best.

- Sometimes you'll be asked to give specific attention on the artistic side of your project. In this case, it'll be mentioned explicitely in the subject. Don't hesitate to try a lot of different things to get a good idea of the possibilities offered by Unity.

- By Odin, by Thor ! Use your brain !!!

Chapter III

Specific instructions of the day

- To manage character's movements don't lose to much time and import the First Person Controller from Unity's Characters package. You can find it in the Standard Assets.

Chapter IV

Exercise 00 : Armed and dangerous

	Exercise 00
Exercise 00 : Armed and dangerous	
Files to turn in : `Everything that seems relevant`	
Forbidden functions : `None`	
Remarks : `n/a`	

The base in a respectable FPS is weapons. You will have to start by implementing these. You will have to take the following rules into consideration:

- It is possible to distinguish 2 types of weapons. The first one shots with precision and will always hit only one enemy. The second one does area damage and can hit several enemies in a specific range.
- Weapons must all have a different attack speed and damage.
- When shooting it's possible to see a trace of the shot as well as particules at the location hit by the bullet. It must be possible here again to distinguish range/particules.
- It's possible to switch from one equipped weapon to another by pressing 1 and 2. It's obviously possible to shoot by pressing the left click on the mouse.
- Weapons make different sounds when shooting and are a minumum animated (see the demo of the day).

`If you are not able to able to finish the second exercise, put together fixed targets to test the implementation of the weapons. It must be possible to "kill" them.`

Chapter V

Exercise 01 : This ain't no place for no hero

	Exercise 01
Exercise 01 : This ain't no place for no hero	
Files to turn in : `Everything that seems relevant`	
Forbidden functions : `None`	
Remarks : `n/a`	

Awesome we have weapons, now we need to something to shoot at else we're quickly gonna get bored. Add some enemies equipped a minimum of intelligence so that the battles are a little bit interesting.

Here are the minimum required regarding the AI, you are free to add additional rules to make your AI better:

- An ennemy starts from a corner of the map and goes towards the center.
- The AI has quite a big detection zone. If the player enters it, he will be followed. If the enemy loses the player, he will go to the last known position.
- If the enemy is hit he goes to the position of the player at the moment of the shooting. When the life of the enemy reaches 0 he dies and disappear.
- Once the enemy is close enough from the player he attacks and makes him lose X HP per shot.
- The enemy must always use the shortest way to reach a destination.
- The enemy is completely animated it being when he runs, walks, when he attack or when he takes damages/dies.

Put some enemies on a map to be able to play a little bit in a nice scenery.

Chapter VI

Exercise 02 : Endless Arena

	Exercise 02
Exercise 02 : Endless Arena	
Files to turn in : `Everything that seems relevant`	
Forbidden functions : `None`	
Remarks : `n/a`	

Now is the time to implement the gameplay of your game. Let's put together a simple system that proved fun in a lot of cases: The Arena.

For it to be efficient you will have to put together several essential things. First you have to put spawners at each corner of the map. They will make enemies appear every randomly an non simultaneous choosen time. There must never be more than 20 enemies on the map.

You will also have to design a wave system. For each wave the enemy will become stronger. Add a progression that will define the global progression of the difficulty of your game. Try to tune this last one so that your game is still playable after 5 waves. Plan a break between between two waves during which no enemy will spawn. Waves have a definite time but even if the waves is finished, enemies stay alive.

Let's also put a GUI together that will display the life of the playe, the remaining time on the current wave/break and that will notify the beginning of a new wave/break with a message. When dead the GUI displays a message and will show the number of waves passed.

Chapter VII

Exercise 03 : Who's the boss

	Exercise 03
Exercise 03 : Who's the boss	
Files to turn in : `Everything that seems relevant`	
Forbidden functions : `None`	
Remarks : `n/a`	

The cherry on the top for your game will be the boss. Add an additional enemy that will physically be different that the others (can be recognized from far) equipped with a more advanced AI.

The boss always knows where the player is and is always running after him whatever is position is. He has a lot more HP and inflict a lot more damages. To spice things up the boss will be able to thrown energy balls in the direction of the player (no guiding though, check the demo). To represent these balls use a particule, if the ball hits the player he cannot move for 2s.

With great power comes great responsabilitis. Create a special wave for the boss triggered every 3 waves. This wave unlike the others will be finished only when the boss is dead. Usual enemies don't spawn during this wave.

As usual if you reach this exercise, first congratulations, but mostly don't hesitate to have fun and add a lot more cool details to make your game unique and personnal!