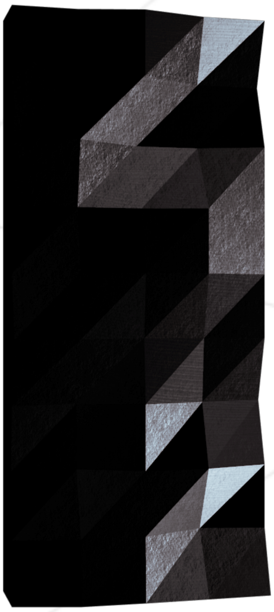

Piscine Unity - Day 02

Audio, Animation and Inter-script Communication

Staff staff@staff.42.fr

Summary: This document contains the subject for Day 02 for the „Piscine Unity" from 42

Contents

Chapter I

Foreword

Today you will teleport to the magical and violant Warcraft universe. Humans, Orcs but mostly a lot of blood. You will learn basic RTS mechanics.

Planned for you today, battles, loyal units more than happy to serve the master that will be your mouse. Plundered villages turned into ruins and ashes. To accompany you in your quest nothing is better than Youtube the famous minstrel whom will make battle war drums roar during your research:

- A soundtrack that deserve your attention
- And because all of this is really a lot of violence, here is something to calm down.

Good luck to you all and may this day be glorious.

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

Exercise 00 : Point and click

	Exercise 00
Exercise 00 : Point and click	
Files to turn in : `You project's folder.`	
Forbidden functions : `None`	
Remarks : `n/a`	

In this first exercise you will put a character on a map provided in the assets. It must be possible to move him by clicking somewhere on the map that will act as his location. The character cannot teleport, it must move to the destination. The sprite must be oriented in the direction of the destination as well.

Make your character a little more alive by adding a sound when he starts to move. Also add a walking animation which will activate only when the character will effectively move.

Chapter IV

Exercise 01 : The more the merrier

	Exercise 01
Exercise 01 : The more the merrier	
Files to turn in : `You project’s folder.`	
Forbidden functions : `None`	
Remarks : `n/a`	

One character is nice, more is better. You will therefore put together several character (with the same attributs) and ensure that it is possible to select them all before giving an order.

It must be possible to left click on a character to select him. It will add him to the current selection. If one or more other characters are already in the current selection, it will be emptied. However it must be possible to add a character without emptying the current selection by left clicking with the `control` key pressed.

If a left click isn’t made on a character then the whole selection goes to the location clicked like the previous exercise. If there is a right click, selection is emptied.

Chapter V

Exercise 02 : Buildings and villages

	Exercise 02
Exercise 02 : Buildings and villages	
Files to turn in : `You project's folder.`	
Forbidden functions : `None`	
Remarks : `n/a`	

The goal of this exercise is to set up two ennemy villages: one will be Orc and the other Human. Both villages are identical but the sprites representing them will obviously be different.

A village is composed of a townhall and of 4 buildings of your choice. The HP number of each buildings is up to you however the townhall must have significantly more than the 4 other.

The townhall of each village will spawn a new human or orc warrior, depending on its allegiance, every 10 secondes in front of its gate. It is not a problem if the characters are superposed.

The players's warriors must be movable according to the last exercise. The ennemy's warrior are static for now since we'll go over battles in the next exercise.

Chapter VI

Exercise 03 : Some action

	Exercise 03
Exercise 03 : Some action	
Files to turn in : `You project's folder.`	
Forbidden functions : `None`	
Remarks : `n/a`	

Everything is now ready we are lacking a little bit of action. Your characters will now be able to hit everything that moves, even things that don't move.

Add the possibility to attack a target. It must be possible to click on a target to make the active selection attack. If the characters are too far away to attack, they must move to an acceptable distance to do so. If the target runs away they will have to follow it except if the player give another order.
Your characters may attack ennemy units and buildings. Don't forget to add a combat sound and animation. When a unit or a building doesn't have any HP left, it disappear with an appropriate sound. The game ends when one of the 2 townhall is destroyed.

Warning: each time a building that is not the townhall is destroyed, the spawn time of units of the townhall is increased by 2,5 secondes. It means that if only the townhall of a player remains, units will spawn every 20 secondes instead.

Also add information in the console to know who is attacked and how much HP are left.

```
Orc Unit [50/100]HP has been attacked.
```

Ensure that a message is displayed when the game ends as well:

```
The Human Team wins.
```

For this exercise, the ennemy village is still inactive don't lose time to script a small AI. It'll done in the next exercise.

Chapter VII

Exercise 04 : Player vs AI

	Exercise 04
Exercise 04 : Player vs AI	
Files to turn in : `You project’s folder.`	
Forbidden functions : `None`	
Remarks : `n/a`	

Now that your game is ready, the time has come to script the opponent. Your ennemy will want to destroy everything in his way, because he personnaly hates you. His first priority will be to tear down everything especially your townhall. However if his troups meet yours on the way he will not be able to stop himself from changing his objectif a minute to smash your troups.

Additionally, if the townhall of the AI is attacked by the human player, the AI will call back his troups to defend it.

The previous instructions constitue the minimum acceptable for your AI. When these basic behavior are correctly implemented you are free to improve your AI to make it a little less stupid and more interesting to fight.