Piscine Unity - d02

Audio, Animation and communication between scripts

Staff staff@staff.42.fr

Summary: *Here you will find the subject for the Day02 of the Unity piscine of 42.*

Contents

Chapter I

Preamble

Today, you're gonna transpose the violent and faery world of Warcraft. No humans, orcs, but mostly, blood. You will learn the rudiments and basic mechanics of the RTS.

Here is a preview of the program: war. Dedicated units ready to serve their master, that is: your mouse. Looted villages burnt to the ground. And to support you along your quest, what about a minstrel named Youtube, that will have the war drums beating during your researches:

- A mighty fine soundtrack

- Ok, and since it is a little violent, here is a fair compensation.

Good luck and may your day be victorious.

Chapter II

Instructions

- 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.

- Think about discussing on the forum Piscine of your Intra!

- Use your brain!!!

Chapter III

Exercise 00: Point and click

	Exercise : 00	
	Exercise 00: Point and click	
Turn-in directory : *ex00/*		
Files to turn in : `The folder in your project.`		
Forbidden functions : `None`		

In this first exercise, you will set up a character in the map provided with the assets. You must be able to move it clicking anywhere on the map. The click will give the character its destination. It must not teleport but move towards the destination. The sprite must also be oriented towards its destination.

Bring life to your character adding a little sound when it starts moving. Also add a walking animation that will only be triggered when it walks.

Chapter IV

Exercise 01: The more the merrier

![icon]	Exercise : 01
	Exercise 01: The more the merrier
Turn-in directory : *ex01/*	
Files to turn in : `The folder in your project.`	
Forbidden functions : `None`	

One character is good. Several is better. That's why you will set up several characters (with the same features) and make sure you can select them before giving them an order.

You must be able to left click a character to select it. This adds it to the active selection. If one or several characters are already present in the active selection, it is emptied. You must however add a character without emptying the active selection clicking left with the `control` key pressed.

If you left click anywhere on the map but on a character, the whole selection will move towards this point, like in the previous exercise. If you right click, the selection is emptied.

Chapter V

Exercise 02: Buildings and villages

	Exercise : 02
	Exercise 02: Buildings and villages
Turn-in directory : *ex02/*	
Files to turn in : `Your project's folder`	
Forbidden functions : `None`	

The goal of this exercise will be to set up two enemy villages: one will be orc, the other human. Both villages are identical, but their sprites will obviously be different.

A village is made of a City Hall and 4 buildings of your choice. You will choose the resistance of each building but the City Hall must be way more resistant than the others.

Each village's City Hall will spawn a new human or orc warrior every 10 seconds. It doesn't matter if the characters pile up in front of the City Hall's door.

The player must be able to move their warriors like in the previous exercise. For now, enemy warriors stand still since fight will only start with the next exercise.

Chapter VI

Exercise 03: Action

	Exercise : 03
	Exercise 03: Action
Turn-in directory : *ex03/*	
Files to turn in : `Your project's folder`	
Forbidden functions : `None`	

The cards are dealt. We just miss a little action. Your characters are about to knock down any moving target... or even unmoving!

Add the possibility to attack a target. You must be able to click a target to get your active selection to attack it. If your characters are too far to attack, they will move to do so. If the target escape, they will have to follow it unless ordered otherwise.

Your characters can attack enemy units and buildings. Don't forget to add sound and animation for the fight. When a unit or a building runs out of life, it is destroyed and disappears from the map with a matching sound. The game stops when one of the City Halls is destroyed.

Warning: each time a building that is not a City Hall is destroyed, the unit spawn time increases by 2,5 seconds for each destroyed building. This means when the City Hall is the last building remaining, the spawn time will be 20 seconds.

Also add informations in the console to show to know who is attacked and the life they have left.

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

Make sure a message is displayed when the game is over:

```
    The Human Team wins.
```

For this exercise, the enemy camp is always still. Don't waste your time scripting an IA. Save it for the next exercise.

Chapter VII

Exercise 04: Player vs AI

	Exercise : 04
	Exercise 04: Player vs AI
Turn-in directory : *ex04/*	
Files to turn in : `Your project's folder`	
Forbidden functions : `None`	

Now your game is ready, it's time to script an enemy. Your rival will want to destroy everything with your name on it. Damn, he hates your guts. First, he will want to wipe you out, starting with your City Hall. However, if its units are moving close to yours, he won't be able to help himself and will go after them.

Besides, if the AI's City Hall is attacked by the player, the AI calls back its units to defend it.

Previous instructions are the least behavior expected of your AI. Once this basic behavior is implemented, you will have to enhance your AI to make it a little less dumb and turn it into a worthy opponent.