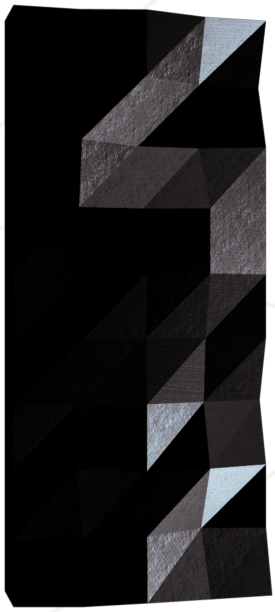


Unity Piscine - Module05

Singleton, PlayerPrefs and coroutines

Summary: In this document, you will find the Module05 subject for 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

- Today you will use your work from the previous Module04. So, you need import it into your new project.

Chapter III

Foreword

Ash nazg durbatulûk, ash nazg gimbatul, ash nazg thrakatulûk agh burzum-ishi krimpatul.

« One Manager to rule them all, one Manager to find them, one Manager to bring them all and into the darkness bind them.. »

Chapter IV

Exercise 00: A good leaf

	Exercise :
Exercise 00: A good leaf	
Turn-in directory : `unityModule05`	
Required elements : `The Stage1, Stage2, Stage3 " scenes, GameManager.cs script and anything relevant`	
Forbidden functions : `None`	

Today and until the end of your piscine, you will need to save some informations throughout your game, and is strongly recommended not to duplicate them unnecessarily from one scene to another.

Take some time to go find out about the `Singleton` design pattern, and the `DontDestroyOnLoad` method, on the Unity Scripting API.
You will now be required to use it. Named your singleton script, GameManager. It must be contain the DontDestroyOnLoad method.

With the work you have done in the previous module, you will create Stage1, Stage2 and Stage3 scenes which contain platforms and enemies. Create your scenes quickly, don't waste time on them.

You will need to create a new prefabs that you will add to each of your scenes :

- An visible Start point : Placed at the beginning of the stage, this is where the caterpillar start the stage and where it reappear when he dies.
- An end of stage point : When the caterpillar arrives at this point, it is sent to the next stage.
- An collectable leaf item : An leaf item that your caterpillar will have to collect throughout the stage.

For pass the end stage point, the caterpillar will have to collect 5 leaves.

You will have to place more leaves than necessary in you stage,in order to let finish the stage without having collected them all.

The more Leaves the caterpillar collects, the more points it will get. 1 leaf will bring 5 points, so you will need at least 25 points to finish the stage.

if your caterpillar reaches the end point and doesn't have the required number of leaf, it won't be able to go the next stage and you will have to display, in game (not in the console), a message telling the player that he doesn't have enough points to go through.

It is possible that your caterpillar is blocked.
For example, if it falls in a deep place without being able to go back to get the leaves if needs to pass the end point.
So don't forget to add a button that will allow your caterpillar to return to the beginning of the stage, or, make sure that your scene design allows you to go back.

Chapter V

Exercise 01: PlayerPrefs

	Exercise :
Exercise 01: PlayerPrefs	
Turn-in directory : `unityModule05`	
Required elements : `The "MainMenu, Stage1, Stage2, Stage3" scenes and anything relevant`	
Forbidden functions : `None`	

You must create a user profile that will be saved in the player preferences so it can reloads if you quit and restart the game.

This is good, because Unity has a special class for this, `PlayerPrefs`.

So for this exercise you must saved the HP of your caterpillar, its leaf points and the stages it has unlocked.

So now, create an MainMenu scene, which be your starting scene when you start the game.

It Should contain:

- Title of your game.
- Resume button.
- New Game button.
- Diary button (For now, just create the button, you will do the scene later).

When you choose Resume, your caterpillar must have the same number of HP, same number of leaf points and be at the start of te last stage you unlocked the last time you played. Therefore the leaves already collected in the level should not appear anymore.

When you choose New Game, all progress is erased. You start the game again from the 1st stage, you have 3HP and 0 leaf points.

You will need to add a button in all your stages that allows the player to return to the main menu at any time.
Don't forget to save the progress of your caterpillar when you use this button.

You will use the playerprefs only for educational purpose. You should NEVER use it to store infos the player is not supposed to modify (their progress for instance). The playerprefs are stored in a file, hence they're editable. They will be mostly used to keep the player settings regarding the game options in memory: the keybindings, the audio and video prefs, etc...

Chapter VI

Exercise 02: User Interface

	Exercise :
Exercise 02: User Interface	
Turn-in directory : `unityModule05`	
Required elements : `The "MainMenu, Stage1, Stage2, Stage3, Diary" scenes and anything relevant`	
Forbidden functions : `None`	

Now for little more clarity in game, you will add a graphical interface that will be visible in all stages.

It will be display :

- Number of HP of the caterpillar. You must see in real time the HP decrease or increase according to the HP that the caterpillar loses or gains.
- The number of leaf point. You must see in real time the number of point increase when the caterpillar collect a leaf.

These two elements will have to be reset to zero each time the caterpillar changes stage, or if the level is reset (if you have added the possibility to do so).

This interface must be visible in all scenes but not duplicated in each scenes.

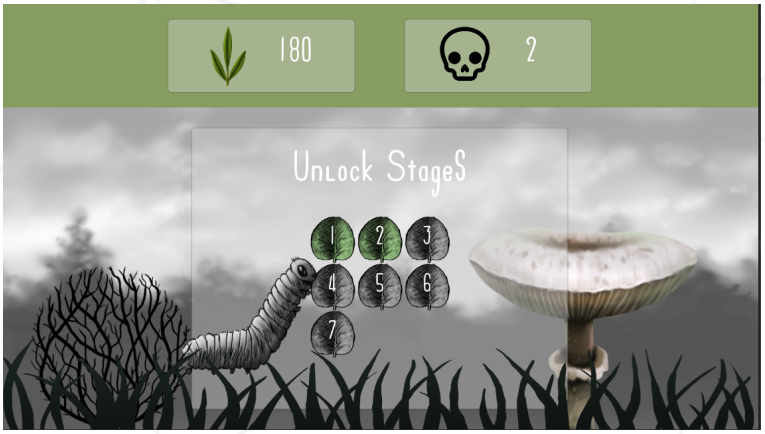
You must also create a Diary scene that allows to view all informations of your caterpillar progression.

This scene must contain :

- The number of leaf points your caterpillar has obtained since the start of the game.
- The number of time your caterpillar has died since the start of the game.

- The lock and unlock stages of your caterpillar.

your scene should look something like this :



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 textttAsset 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.