Unity Piscine - Module05

Singleton, PlayerPrefs and Coroutines

Summary: This document outlines the subject for Module05 of the Unity Piscine.

Version: 2

Contents

Chapter I

Instructions

- If you have trouble installing the required tools for your project on the 42 computers, you may use a virtual machine. In this case, you must:
  - Install the virtual machine software on your computer.
  - Install the operating system of your choice.
  - Install the necessary tools for your project.
  - Ensure that you have enough space on your session to install all of it;
  - Have everything installed before the evaluation.
- Only this page will serve as reference. Do not rely on rumors.
- Carefully read the entire document before starting.
- Your exercises will be evaluated by your fellow piscine participants.
- This document is your reference. Do not blindly trust demos or example pictures, which may include unnecessary additions.
- Got a question? Ask the peer to your right. If not, try the one on your left.
- By Odin, by Thor! Use your brain!!!

`Intra shows the date and time when your repositories close. This also marks the beginning of the peer-evaluation period for that piscine day. The peer-evaluation lasts exactly 24 hours. After that, any missing evaluations will be scored as 0.`

Chapter II

Specific Rules

- Today, you will build upon your work from Module04. Import it into your new project to begin.

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 all relevant assets`	
Forbidden functions : `None`	

From today until the end of your piscine, you'll need to store information throughout the game. It's strongly recommended not to duplicate data unnecessarily between scenes.

Take some time to learn about the **Singleton** design pattern and the `DontDestroyOnLoad` method via the Unity documentation
You are now required to implement these.

Name your singleton script `GameManager`. It must include the `DontDestroyOnLoad` method.

Using your previous module's work, create the Stage1, Stage2, and Stage3 scenes with platforms and enemies.
Set these up quickly; don't waste time on polish.

You must also create the following **prefabs** and add them to each scene:

- A visible Start Point: This marks where the caterpillar starts and respawns upon death.
- An End Point: When reached, the caterpillar progresses to the next stage.
- A Collectible Leaf Item: Leaves that the caterpillar collects throughout the level.

To pass the End Point, the caterpillar must collect at least 5 leaves.

You should place more than 5 leaves per stage to allow flexibility.

Each leaf grants 5 points, so a minimum of 25 points is required to proceed.

If the caterpillar reaches the End Point without enough points, display an in-game message (not in the console) informing the player.

Consider cases where the caterpillar might become stuck (e.g., falling into a pit). Include a button to restart the stage or ensure your level design allows for backtracking.

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 is saved using Unity's `PlayerPrefs`, so it can be reloaded after quitting and restarting the game.

For this exercise, store the caterpillar's HP, leaf points, and unlocked stages.

Create a `MainMenu` scene, which will be the game's starting point.
It should include:

- Game title
- Resume button
- New Game button
- Diary button (for now, only create the button; the Diary scene will be made later)

- Choosing "Resume" should load the caterpillar's last position, saved HP, leaf points, and the last unlocked stage. Leaves already collected should no longer appear.

- Choosing "New Game" resets all progress. You start from Stage 1 with 3 HP and 0 leaf points.

Include a button in each stage to return to the main menu. Ensure progress is saved when this button is used.

While PlayerPrefs can be useful in certain production contexts-such as storing non-critical data like player settings (e.g., keybindings or audio/video preferences), the way we use it in this exercise is intended purely for educational purposes. For example, storing critical data like player progress using PlayerPrefs is not recommended, as the data is stored in easily editable files.

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`	

For better in-game clarity, add a graphical interface visible in all stages!

It should display:

- The caterpillar's HP, updated in real-time.
- The number of leaf points, updated as leaves are collected.

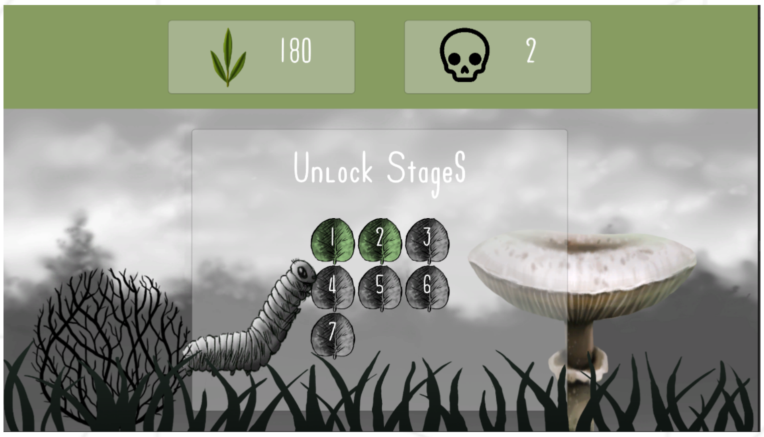
These values must reset to zero whenever the caterpillar changes stages or the level is reset.

The interface must persist across all scenes without being duplicated in each.

Additionally, create a `Diary` scene to track and display the caterpillar's progress.
This scene must include:

- Total leaf points collected since the game began
- Number of times the caterpillar has died
- List of locked and unlocked stages

Your Diary scene could look something like this:



Chapter VII

Submission and Peer Evaluation

Submit 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 make sure everything is correct.

You should not upload the entire Unity project to Git, as this can unnecessarily increase the size of your repository.

- Make sure Unity saves as many files as possible in text format rather than binary. In Unity, go to Edit > Project Settings > Editor. Under Asset Serialization, set it to Force Text.
- Ensure that the .gitignore file automatically generated by Unity is present.

The evaluation will take place on the computer of the learner or group being evaluated.