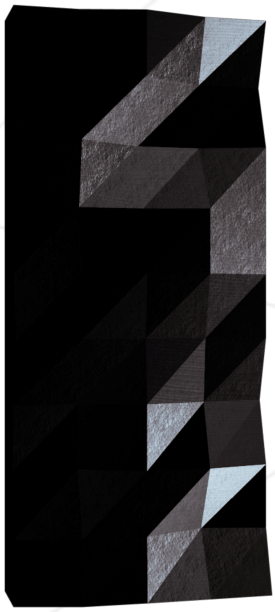


Unity Piscine - Module00

Learn Unity basics by building a 3D game with movement, jumping, level design, and simple game-over logic.

Summary: Here is the subject Module00 for 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

Piscine Unity Starter Kit

Welcome to the first module of the Unity Piscine. To get off to a good start, here is a list of useful tips and links that will help you throughout the Piscine.

- Official Unity documentation. You will also find a blue book icon in the Inspector near each component that links directly to its specific documentation.
- Official C# documentation
- The subject prevails. Don't fully trust the demos as they may contain additional elements not required.
- If you create reusable generic utility scripts unrelated to gameplay, they will save you time in future modules.
- nce the module is completed, don't keep your project in your home directory. Unity tends to generate a huge number of files that will quickly eat up your 5 GB quota (and slow down your session).
- These modules are long, so don't waste time on unnecessary details. You'll have time to enhance your game once the mandatory parts are complete.
- If you or one of your neighbors has the answer to a **TECHNICAL** problem, you can share it on the forum using the Unity Piscine tag.

Chapter III

Introduction

"The Floor is Lava" is a game in which players pretend the floor is made of lava (or another dangerous substance like acid or quicksand), and must avoid touching the ground or risk being "killed".

Players stay off the floor by standing on furniture or structures. They usually have to keep moving and can't stay on one object for long due to imaginary limitations (e.g., it's sinking or melting).

The game can be played solo or with others. It might include goals or races, and can be set up indoors or outdoors, with obstacles like padded chairs to increase difficulty. It resembles an obstacle course.

Anyone can start the game by shouting "The floor is lava!" Anyone still on the floor within seconds is out and must sit out temporarily.

Some versions include items or locations that restore health or body parts, with tasks ranging from silly to simple, like finding someone. make tre In *"Hot Lava Monster"* (also called *"Skies in the Ringuss"*), often played on playgrounds, the "monster" can walk on the lava and tries to tag others, who must stay on the play structure. The monster may have limitations like avoiding certain colors or platforms.

The game is similar to *"Puss in the Corner"*, where children must move from corner to corner without being tagged by the "Puss" in the center.

Chapter IV

Exercise 00: Floor Is Lava

	Exercise :
Exercice 00: Floor Is Lava	
Turn-in directory : `unityModule00`	
Required elements : `FloorIsLavaScene scene, Floor, Pathway GameObjects and any relevant elements`	
Forbidden functions : `None`	

The purpose of this module is to get you familiar with the different tools provided by Unity.

To start, create a new 3D project and name it *Module00*. A sample scene has already been created—you can work directly in it. Rename this sample scene to *FloorIsLavaScene*. You'll only need one scene for all the exercises.

To create this game, you'll need:

- The floor!
- A pathway.
- A player.
- Some decorations.
- A camera.

Let's start with the floor:

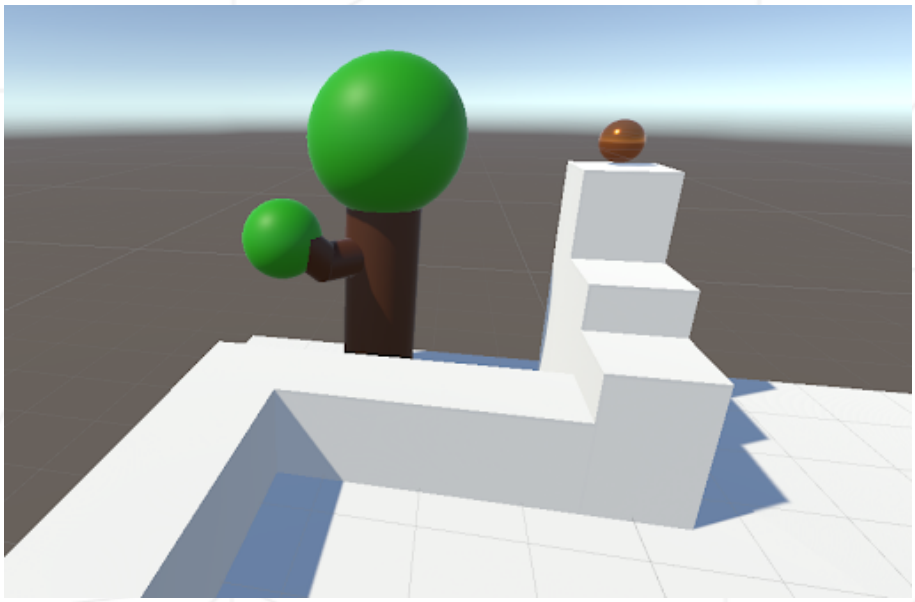
- Create a Floor GameObject using a primitive square.
- Choose its size (it doesn't need to be huge).

Now the pathway:

- Create an empty GameObject named *Pathway*, in which you'll place all the different GameObjects (stairs, bridges, or whatever you like) that will make up your pathway.
- When building it, keep in mind the goal of the game: move along the pathway without touching the floor!

Decoration time:

- Your scene must include at least one composite structure (with colors, because it looks nicer!).
- Here's an example of a compound structure, a tree (but feel free to come up with your own creations):



The Player:

- Your player will look like a ball, so choose an appropriate 3D object.
- Make it look nice by adding a material, color, reflections, whatever you like.
- Place it at the start of the pathway.
- You'll add movement in the next exercise.

And finally, the camera:

- The main camera is automatically created when you start the project. For this module, you don't need to modify it. Just position it high enough so the player can see the entire pathway.

Chapter V

Exercise 01: Path Of Exile

	Exercise :
Exercise 01: Path Of Exile	
Turn-in directory : `unityModule00`	
Required elements : `A FloorIsLava scene, PlayerController script and any relevant elements`	
Forbidden functions : `None`	

Now that you've got your structure and your player, it's time to make them move along the path.

To do this, you'll need to create a script named `PlayerController.cs`, and attach it to your player.

The player should be able to move using the `'WASD'` ('ZSQD', depending on your keyboard) keys or the `arrow` keys, both options should work.

Let's spice things up a bit!

Your player should also be able to jump.
So, place some obstacles along your pathway that the player will need to jump over, and of course, don't forget to implement jumping in your `PlayerController` script.

Chapter VI

Exercise 02: My Even More Beautiful World

	Exercise :
Exercise 02: My Even More Beautiful World	
Turn-in directory : `unityModule00`	
Required elements : `A FloorIsLava scene and any relevant elements`	
Forbidden functions : `None`	

`You can find the texture pack needed for this exercise on the Unity Asset Store: Official Unity Asset Store.`

Right now, your scene is looking a bit dull. Let's fix that by adding some texture and life to it using assets from the Unity Asset Store.

Make your floor look like lava, your pathway like stone, or maybe add some leafy vegetation as obstacles!

In short: let your imagination run wild, but don't spend too much time on it either.

Chapter VII

Exercise 03: The Floor Is 'Always' Lava

	Exercise :
Exercise 03: The Floor Is 'Always' Lava	
Turn-in directory : `unityModule00`	
Required elements : `A FloorIsLava scene file, the assets and scripts specific to the exercise`	
Forbidden functions : `None`	

Now we get to the heart of the matter, and the end of it too.
The floor is still lava!

So you'll need to make sure that if your player falls off the pathway, it's game over.

For now, you don't need a dramatic title screen or fancy effects. Just a simple `Debug.Log` that displays `"Game Over"` in the console will do. Also, the Player GameObject must be destroyed when the game is over.

Chapter VIII

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.