



Piscine Unity - Module01

3D physics, Tags, Layers, and Scenes

Summary: This document contains the subject for Module01 of the Unity Piscine.

Version: 2

Contents

I	Instructions	2
II	Foreword	3
III	Exercise 00 Thomas and His Friends	4
IV	Exercise 01: Exit This Way!	6
V	Exercise 02: Stage2!	8
VI	Exercise 03: Interactivity	9
VII	Exercise 04: Buttons!	10
VIII	Exercise 05: A Deadly Game!	11
IX	Submission and Peer Evaluation	12

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


Foreword

Today's stars:

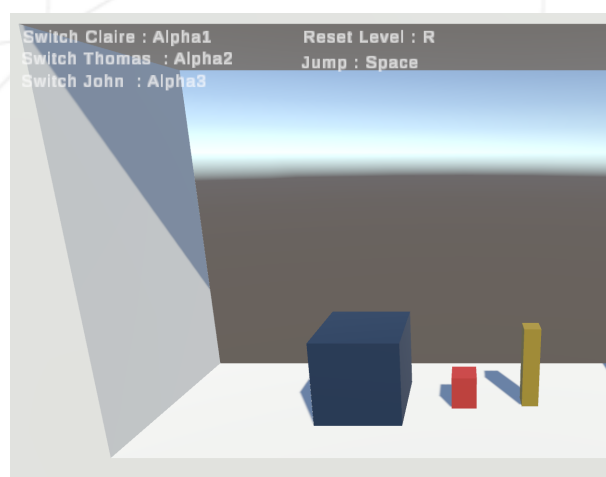
-  Thomas - Lonely and a bit naive, Thomas enjoys listing his observations about the world, and he's absolutely fantastic at falling.
-  Chris - Pessimistic, irritable, and suspicious, Chris might not be the best jumper, but he was doing just fine on his own.
-  John - Proud of his agility and athleticism, John rather enjoys an audience, so he decides to look after Thomas and Chris.
-  Claire - Claire lacks confidence. She moves slowly and thinks she's rubbish at jumping. But then she discovers she might actually be a superhero.
-  Laura - Laura isn't great at jumping either, though she has a unique ability, one she's too ashamed to tell anyone about. Recently, an ominous pixel cloud has been following her, and the others are getting worried.
-  James - James has always been different; not least because of his unique disregard for the laws of Newtonian physics.
-  Sarah - On a quest for the fountain of knowledge and the truth about her world, Sarah sees herself as far more intelligent than the other, "lesser" quadrilaterals.

Chapter III

Exercise 00 Thomas and His Friends

	Exercise :
Exercise 00: Thomas and His Friends	
Turn-in directory : <code>unityModule01</code>	
Required elements : The "Stage1" scene, <code>PlayerController</code> scripts on each character, and anything else relevant	
Forbidden functions : None	

- Create a scene called **Stage1** with the following:
 - A ground.
 - A camera.
 - Three characters: Claire, John, and Thomas.



- At the beginning, no character is active. You can activate and switch between them using the `Alpha1`, `Alpha2`, and `Alpha3` keys. Characters must help each other to complete the levels!

- Move characters left and right using the 'A' and 'D' keys ('Q' and 'D', depending on your keyboard layout). Press the **spacebar** to jump.
- The camera should automatically center on the active character.
- You must be able to reset the scene by pressing a key, either R and/or Backspace.




Warning! You must create ONE single script that will be applied to all characters.



You're free to create other independent scripts - to manage the camera, for example.

Chapter IV

Exercise 01: Exit This Way!

	Exercise :
Exercise 01: Exit This Way!	
Turn-in directory : <code>unityModule01</code>	
Required elements : The "Stage1" scene, and anything else relevant	
Forbidden functions : None	

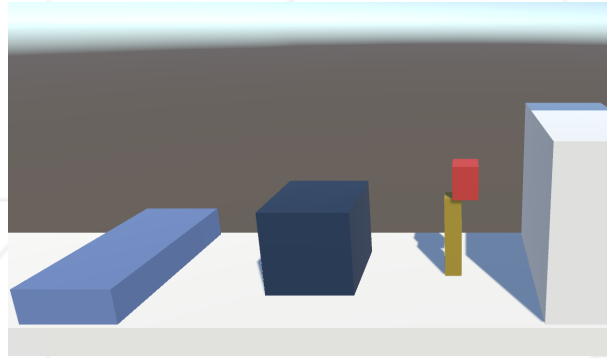
Now, the characters must each have different traits:

- Claire, the blue square, moves slower and jumps lower than the others.
- John, the yellow stick, moves faster and jumps higher than the others.
- Thomas is somewhere in between, with average speed and jump height.

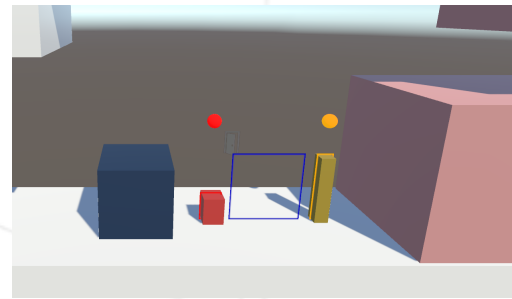
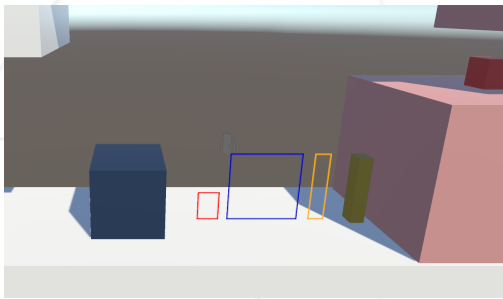


You must always use the same script for all your characters.

- Characters should not be able to jump multiple times without first landing on a surface, whether it's the ground or another character. No infinite jumps or wall jumps allowed!
- They must navigate through a first stage that forces them to cooperate in order to reach the exit.




- Each character's exit is indicated by an outline that matches their shape or color. Once all characters are correctly aligned with their respective exits, the stage is complete. You must display a message stating this — for now, a simple message in the console will do.



The stage should be designed to require cooperation - no character should be able to reach their exit on their own.

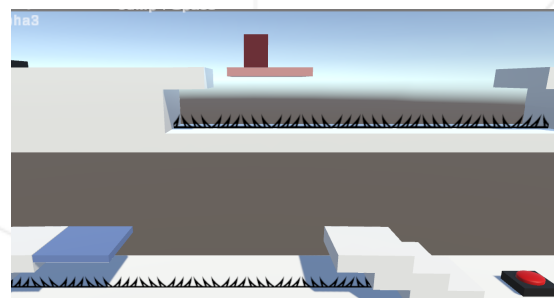
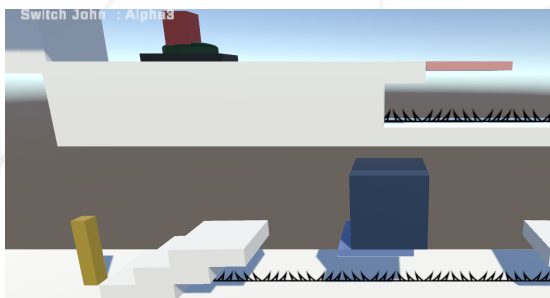
Chapter V

Exercise 02: Stage2!

	Exercise :
Exercise 02: Stage2!	
Turn-in directory : <code>unityModule01</code>	
Required elements : The "Stage2" scene and anything else relevant	
Forbidden functions : None	

You must create a second stage using **Physics Layers** as part of the level design:


- Platforms should be either white or match a character's color.
- Characters can only use platforms that are white or match their own color; they will fall through the others.



- You must link the stages. When the characters finish a stage, they should move on to the next one.
- You must link the stages. When the characters finish a stage, they should move on to the next one. If there are no more stages, they should return to the first stage.

Chapter VI

Exercise 03: Interactivity

	Exercise :
Exercise 03: Interactivity	
Turn-in directory : <code>unityModule01</code>	
Required elements : <code>"Stage3" scene and anything relevant</code>	
Forbidden functions : <code>None</code>	

Things are about to get interesting! Create a Stage3 featuring teleporters and moving platforms.

Feel free to design fast, technical pathways or sadistic elevators — just like in a good old retro game, where it takes 10 seconds to catch the right elevator timing.

About teleporters:

When a character enters a teleporter, they should instantly appear at a linked destination. You're free to design how the teleporters are visually represented, but make sure the player understands where they lead, either by color, shape, or effect. Each teleporter should have a clear entrance and exit.


The goal is to build a level that's enjoyable to play through — not just a flashy display of your skills.



Don't forget to progressively add your levels to the build.

Chapter VII

Exercise 04: Buttons!

	Exercise :
Exercise 04: Buttons!	
Turn-in directory : <code>unityModule01</code>	
Required elements : The "Stage4" scene and anything else relevant	
Forbidden functions : None	

Time to push some buttons!

Create a Stage4 with switches that open doors — or better yet, color-coded switches that open doors of the same color.


Even better! white switches that take on the color of the character who activates them, and open doors matching that color.

And finally! switches that change the color of platforms, affecting which characters can use which paths.

You're completely free to design this level however you like — as long as the buttons work as described above.

Chapter VIII

Exercise 05: A Deadly Game!

	Exercise :
Exercise 05: A Deadly Game!	
Turn-in directory : <code>unityModule01</code>	
Required elements : The "Stage5" scene and anything else relevant	
Forbidden functions : None	

For the final stage — Stage5 — it's time to raise the difficulty a bit, so the player doesn't feel like they're just out for a stroll:

- Create color-coded turrets that fire at regular intervals. Each turret's shots can only hit characters of the same color.
- Add traps — on the ground or in the air.
- Add holes. The camera should not follow a character once they fall into a hole.
- If a character is hit by a turret shot, triggers a trap, or falls into a hole, the game is over.



You can implement everything using what you've learned in the last two days. Don't use any other systems - especially not timed actions or coroutines to manage turret firing. We'll get to that later.

Chapter IX

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.