Piscine Unity - Day 01

2D Physic, Tags, Layers and Scenes

Staff <staff@staff.42.fr>

Summary: *This document contains the subject for Day 01 for the „Piscine Unity" from*
42

Contents

Chapter I

General Instructions

- The Unity bootcamp has to be made entirely, exclusively and mandatorily in `C#`. No `Javascript/Unityscript`, `Boo` or any other horrors.

- The use of functions or namespace not explicitely authorised in the exercise header or ini the rules of the day will be considered cheating.

- For a optimal usage of Unity, you have to work on `~/goinfre`, which is on the local drive of your computer. Remember to make appropriate backup on your own, the local goinfre can be purged.

- Unlike any other bootcamps, each day doesn't require a folder `ex00/`, `ex01/`, ..., `exn/`. Instead you'll have to submit your project folder which will be name like the day: `d00/`, `d01/`, ... However, a project folder, by default, contains a useless folder: the `"projet/Temp/"` sub folder. Make sure to **NEVER** try to push this folder on your repository.

- In case you're wondering about it, there is no imposed norme at `42` for `C#` during this bootcamp. You can use whatever style you like without restrictio. But remember that code that can't be read or understood during peer-evaluation is code that can't be graded.

- You must sort your project's assets in appropriate folders. For every folder correspond one and only one type of asset. For exemple: `"Scripts/"`, `"Scenes/"`, `"Sprites/"`, `"Prefabs/"`, `"Sounds/"`, `"Models/"`, ...

- Make sure to test carefully prototypes provided every day. They'll help you a lot in the understanding of the subject as well as what's requested of you.

- The use of the Unity Asset Store is forbidden. You are encouraged to use the daily provided assets (when necessary) or to look for additional ones on the Internet if you don't like them, exception made of scripts obviously because you have to create everything you submit (excluding scripts provided by the staff). The Asset Store is forbidden because everything you'll do is available there in one form or another.

However the use of Unity Standard Assets is authorised and event advised for some exercises.

- From d03 for peer-evaluation you'll be required to build the games to test them. `The corrector` will have to build the game, you must therefore always push projects/sources. You project must always be properly configured for the build. `No` last minute tweaks will be tolerated.

- Warning: You'll not be corrected by a program, except if stipulated in the subject. This imply a certain degree of liberty in the way you can do exercises. However keep in mind the instructions of each exercise, don't be LAZY, you would miss a lot of very interesting things.

- It isn't a problem to had additional or useless files in your repository. You can choose to separate your code in different files instead of one, except if the exercise's header stipulate a list of files to submit. One file must define one and only one behaviour, so no namespace. Those instructions don't apply to the `"projet/Temp/"` sub-folder which isn't allowed to exist in your repositories.

- Read carefully the whole subject before beginning, really, do it.

- This document could potentially change up to 4 hour before submission.

- Even if the subject of an exercise is short, it's better to take a little bit of time to understand what's requested to do what's best.

- Sometimes you'll be asked to give specific attention on the artistic side of your project. In this case, it'll be mentioned explicitly in the subject. Don't hesitate to try a lot of different things to get a good idea of the possibilities offered by Unity.

- By Odin, by Thor ! Use your brain !!!

Chapter II

Today's starring:

- Thomas - Lonely and slightly naive, Thomas likes to list his observations about the world - and is 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 - Rather proud of his agility and sportiness, John quite likes an audience so decides to look after Thomas and Chris.

- Claire - Claire lacks confidence, she moves slowly and considers herself rubbish at jumping. But then she discovers that she might be a superhero.

- Laura - Laura isn't great at jumping, although she does have her own unique ability - which, sadly, she's too ashamed to tell anyone about. An ominous pixel cloud has been following her around lately, and this worries the others.

- James - James had always been different. Not least because of his unique disregard for Newtonian laws.

- Sarah - On a quest to find the fountain of knowledge and learn the truth about her world, Sarah sees herself as rather more intelligent than the other "lesser" quadrilaterals.

Chapter III

Exercise 00 : Thomas and his friends

	Exercise 00
	Exercise 00 : Thomas and his friends
Files to turn in : `A scene file ex00, a "playerScript_ex00" script on each character, a script for the camera`	
Forbidden functions : `None`	
Remarks : `n/a`	

⚠ `WARNING! From today and until the end of the bootcamp, you'll have to submit all your exercises in Assets/ as explained in the general instructions and NOT in sub-folders like ex00, ex01 ...`

To begin, run the demo to ge an idea about what today is about. You can move characters using the arrows of the keyboard and space to jump. The `1`, `2` and `3` keys enables the player to change the active character. Mutual aid between characters is key to succeed levels!

Small warmup exercise. Create a scene with aground, a camera and 3 characters: Clair, John and Thomas.

We must be able to select a character by pressing `1`, `2` or `3` and move him using the arrows. The camera will center automatically on the active character.

Every character can jump, move to the lest and to the right. It must be possible to reset the scene using a key, for example `R` or `Backspace`.

Warning! You must imperatively create ONLY ONE script which will be applied on the 3 characters.

You are free to create additional independent scripts for the camera management for example.

Chapter IV

Exercise 01 : Here is the exit!

![icon]	Exercise 01
Exercise 01 : Here is the exit!	
Files to turn in : `A scene file ex01, a "playerScript_ex01" script on each character, everything that seems relevant`	
Forbidden functions : `None`	
Remarks : `n/a`	

> Unity doesn't like scripts duplicates much, you should create a `playerScript_ex01` and copy/paste your code from last exercise to upgrade it with the features of this exercise.

Now characters need to own their own caracteristics:

- 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 in between the other with an average jump heights and speed.

> ⚠ You still must imperatively use the same script for every character.

They can't jump several times without falling on a surface first - ground or other

character - meaning no infinite or wall jump!

They have to go through a first level that forces them to cooperate to get to the exit. The exit of each character is indicated by the outline representing it. When every character is aligned with its own exit the player wins the level. You must display a message indicating it - a simple debug message is enough for now.

> The level must completely force cooperation so it must be impossible
> for characters to reach the exit without supporting each other.

Chapter V

Exercise 01 : Level 2!

	Exercise 01
	Exercise 01 : Level 2!
Files to turn in : `A scene file ex02 and everything that seems relevant`	
Forbidden functions : `None`	
Remarks : `n/a`	

You will now create a second level using physical layers in the level design:

- Platforms are now either white either the same color than a character.

- Characters can only use the white platforms or the ones of the same color than them, they'll go through the other ones.

- You must link both levels together, so that the player goes to this level when he finished the last level with every characters.

Chapter VI

Exercise 01 : Interactivity

![]	Exercise 01
Exercise 01 : Interactivity	
Files to turn in : `A scene file ex03 and everything that seems relevant`	
Forbidden functions : `None`	
Remarks : `n/a`	

Time has come to make it interesting! Create a level with teleporters and moving platforms.

It's up to you to decide if you want to make crazy fast levels requiring technics or sadistic lifts like old NES games where you had to wait10 secondes to have the right plateform alignment. The aim here is really to try to create interesting levels and not just a technical demo.

> `Don't forget to add you level to the build progressively.`

Chapter VII

Exercise 02 : Buttons!

	Exercise 02
	Exercise 02 : Buttons!
Files to turn in : `A scene file ex04 and everything that seems relevant`	
Forbidden functions : `None`	
Remarks : `n/a`	

Now you will have to create a level with switches that open doors. Better than this, color switches that open doors of their color.

Even better, white switches that absorbs the color of the character that uses it which opens the door os the corresponding color...

And finally, switches that change the platform's color, and change the path that characters can take.

As you guessed, you are quite free with the level design, as long as there is buttons that operate as listed above.

Chapter VIII

Exercise 03 : A deadly game!

	Exercise 03
	Exercise 03 : A deadly game!
Files to turn in : `A scene file ex05 and everything that seems relevant`	
Forbidden functions : `None`	
Remarks : `n/a`	

For the last level let's raise the difficulty level a little, so that the player doesn't fell like going through a park:

- Create color towers that regularly fire. The shot touches only the character of the same color.

- Create traps, in the form of ground areas or aerial obstacles.

- Creates holes. The camera doesn't follow a character that falls into a hole.

- If a character is touched by a tower shot, activates a trap or falls into a hole, it's Game Over.

It is possible to code everything with yesterday and today's notions so don't use anything else like timed actions or coroutines to manage towers' shots. You will see all of this at a later stage.