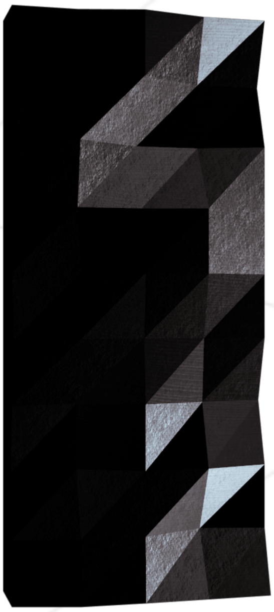

Piscine Unity - Day 07

Navmesh, 3D Physics, 3D GUI

Staff staff@staff.42.fr

Summary: This document contains the subject for Day 07 for the „Piscine Unity" from 42

Contents

Chapter I

Foreword

The time has come to go on an expediction. Aboard your tank, you will be able to move around enemy territory and sneak through to surprise them. Today we got inspiration from `Worlds of Tanks` or other games of the same kind. As usual I will strong advise you to go take a look at these jewels to get an idea about the style we're expecting today.

Finally and because I'm nice I am providing you with quite appropriate musics that fits appropriately, that gives you energy and that will stick in your mind at least the whole day.

Good luck !

Chapter II

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 explicitely 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 III

Specific instructions of the day

- Freedom

Chapter IV

Exercise 00 : Draw me a sandbox

	Exercise 00
Exercise 00 : Draw me a sandbox	
Turn-in directory : *ex*00/	
Files to turn in : `*.cs`	
Allowed functions : `Everything except if said otherwise in today's specific rules`	
Remarks : `n/a`	

So now that you are starting to be an expert in terraforming art you will have to create a magnificant landscape. Be imaginative but remember that the theme must remain compliant with today's game, meaning tanks.

Here are a few indication about your game landscape:

- It must be 256 by 256.
- Try to divide it in game zones with places where it's eventually possible to hide or different heights.
- Use several textures, treat yourself.
- For the environment type, chose what you want you don't have to use the provided example.

`Think about your landscape in advance to create you NavMesh. Avoid too full or too grainy landscapes.`

Chapter V

Exercise 01 : A tank to rule them all

	Exercise 01
Exercise 01 : A tank to rule them all	
Turn-in directory : *ex*01/	
Files to turn in : `*.cs`	
Allowed functions : `Everything except if said otherwise in today's specific rules`	
Remarks : `n/a`	

Now that your sandbox is set we will be able to have fun in it and add toys. Start by creating a tank, we must be able to move the body of the tank using W to go forward and S to go backward, we can direct the body of the tank left with A and right with D. And finally direct the cannon with the mouse. The cannon must only turn on the y axis and must always aim in the mouse pointer's direction (rely on the most representative example).

It must be possible to use a boost to make the tank go faster by pressing the Left Shift key. The boost is limited and and must be recharged when used too much. What would be a tank without missiles and machine guns? Implement the possibility to shot the machine gun with the right click and missiles with the left click or the mouse. Missiles are limited and cannot be fired anymore after a certain number. The machine gun as well as the missiles must produce particules at the location of impact.

⚠ `For the machine gun as well as for the missiles you will have to use a Physical Raycast that will start from the cannon and that will have a limited range.`

Don't forget to put sounds in your game (machine gun, missiles, explosions, background musics, ...) to make it more realistic.

Chapter VI

Exercise 02 : Wargames

	Exercise 02
Exercise 02 : Wargames	
Turn-in directory : *ex*02/	
Files to turn in : `*.cs`	
Allowed functions : `Everything except if said otherwise in today's specific rules`	
Remarks : `n/a`	

Now that we have something to play let's implement a little bit of challenge. Nothing better than a few enemies. Create an IA capable of detect the closest enemy and attack when in range. By "in range" we mean when the tank is at the right height and at the right range to be shot. You can also add some HP to tanks. And when a tank doesn't have any life left, it explodes and disappear. If your tank explodes the game restarts.

Two or three technical considerations:

- You have to use a NavMesg to move tanks controlled by the AI.
- There must be at least 2 enemy tanks on the map and they can attack each other.
- The AI doesn't have to always be accurate and can miss its target.
- You don't need to manage scenery obstructing the AI's aim. An IA that shoots a montain because you are behind will not get penalised.
- If properly designed an AI should not make any collide with another tank.

Chapter VII

Exercise 03 : Pimp My Tank!

Exercise 03
Exercise 03 : Pimp My Tank!
Turn-in directory : *ex*03/
Files to turn in : `*.cs`
Allowed functions : `Everything except if said otherwise in today's specific rules`
Remarks : `n/a`

Let's make the game more classy by adding a GUI. But not any kind of GUI, no, a really cool and sexy 3D GUI. We want couple of essential information around the tank, the remaining life, the number of missiles left. If you want to add additional ones feel free to do so. Implement a distinctive sign to understand that it's a 3D GUI and that will give a usage to the third dimension..

Add a small crosshair sprite that would change color is the player shot successfuly hits.