



## Piscine Unity - Day 06

Staging, Shaders, lights, 3D sounds

Staff [staff@staff.42.fr](mailto:staff@staff.42.fr)

*Summary: This document contains the subject for Day 06 for the „Piscine Unity” from*  
*42*

# Contents

<b>I</b>	<b>Foreword</b>	<b>2</b>
<b>II</b>	<b>General Instructions</b>	<b>3</b>
<b>III</b>	<b>Specific instructions of the day</b>	<b>5</b>
<b>IV</b>	<b>Exercise 00 : The crime scene</b>	<b>6</b>
<b>V</b>	<b>Exercise 01 : And there was Light.</b>	<b>7</b>
<b>VI</b>	<b>Exercise 02 : Big Brother is watching you</b>	<b>8</b>
<b>VII</b>	<b>Exercise 03 : The objective</b>	<b>9</b>
<b>VIII</b>	<b>Exercise 04 : The final touch</b>	<b>10</b>

# Chapter I

## Foreword

It is now time to reveal the artistic talents that lurks inside you. Today's exercises will be centered around the staging and the creation of an atmosphere. And nothing better for this than an infiltration game. Remember splinter cell and Metal Gear Solid, these games really absorbing games that did have a very attention about the general atmosphere and the stagings.

It is now your turn to create this atmosphere, to make sure that your game has a specific atmosphere, a universe. That's what will make it unique and close to your personality. In a world today unleash yourself.

Lastly a small advise for those who take time to read the foreword, it's free. You probably know that Unity provides Standard Assets in it's engine. If I were you I would go take a look at the Effects package today ...

Good luck for this 7th day!

# 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 explicitly authorised in the exercise header or in the rules of the day will be considered cheating.
- For an 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 named 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 norm at 42 for **C#** during this bootcamp. You can use whatever style you like without restriction. 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 example: `"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 even 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. Your 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 implies 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 have 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 stipulates 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 hours 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 III

## Specific instructions of the day

- Today the staging is the guest of honor. It will therefore be your artistic touch that will make a difference. Favor quality over quantity.

# Chapter IV

## Exercise 00 : The crime scene

	Exercice : 00
Exercise 00 : The crime scene	
Files to turn in : *.cs	
Allowed functions : Everything except if said otherwise in today's specific rules	
Remarks : n/a	

To make an infiltration game you must have a proper level design. Put together an inside level. You have to have the following elements at minimum:

- A surveillance camera.
- A ventilation shaft (or anything else that will allow the player to avoid cameras later).
- A locked door with a card reader on the side.
- A keycard relatively far from the door.
- Papers hidden in one of the rooms.


We are talking about decoration elements here, their use will be implemented at a later stage in next exercises. We will add a player that can be moved around using the WASD keys of the keyboard though. It must be a first person view oriented with the mouse.



Don't hesitate to get inspiration from the example of the day. But give it your own touch. Go crazy!

# Chapter V

## Exercise 01 : And there was Light.

	Exercise : 01
Exercise 01 : And there was Light.	
Files to turn in : *.cs	
Allowed functions : Everything except if said otherwise in today's specific rules	
Remarks : n/a	

Infiltration in the dark, why not, but you need to add a little bit of challenge. Create a light atmosphere in your level using everything at your disposal. The more you will temper your lights the more interesting your atmosphere will be..


It is also a good time to start coding as well. Implement a stealth bar in a GUI. If the player runs or enters an area with lights the bar increases. If the bar goes over 75% the player is detected and must return in the shadows. When the player is detected an alarm activates, it can either be visible or audible or both. If the bar reaches 100% it's game over and the game restarts.

When no visible and walking the bar decrease. One more time think about your level design carefully to make your game interesting.



# Chapter VI

## Exercise 02 : Big Brother is watching you


	Exercice : 02
Exercise 02 : Big Brother is watching you	
Files to turn in : *.cs	
Allowed functions : Everything except if said otherwise in today's specific rules	
Remarks : n/a	

Now that we have the basic gameplay set, let's add a little bit of difficulty. Implement a light for the camera to know the range of their vision. If the player enters that area his detection bar will increase drastically, make sure that walking in a camera's range is a dangerous action.

Implement a way to avoid cameras as well. If you have no idea use the ventilation shafts to create smoke particles. If the player pass through the smoke, the camera will detect him a lot less quickly. Here again, try to make sure your design contributes to the atmosphere of your game.

# Chapter VII

## Exercise 03 : The objective

	Exercise : 03
Exercise 03 : The objective	
Files to turn in : *.cs	
Allowed functions : Everything except if said otherwise in today's specific rules	
Remarks : n/a	


To spice up the game a little bit, it's now time to add a final objective. Being a spy the player will have to gather those hidden documents referred to in the first exercise. Hide them in the locked room and make sure it is possible to get the keycard to open the door. It must be the only way to open the door.

When the hidden documents are found, the game ends and restart. Use this exercise to add a little bit of music atmosphere to your scene. Implement "normal" music as well as an "alert" music when the player is detected. Finally add a lot of noises and sounds, everything that can increase the immersion is good (ambiance sounds, fan sounds, key-card sounds, ...).

If you have any doubt about what is expected of you today, check out the demo of the day.

# Chapter VIII

## Exercise 04 : The final touch

	Exercise : 04
Exercise 04 : The final touch	
Files to turn in : *.cs	
Allowed functions : Everything except if said otherwise in today's specific rules	
Remarks : n/a	

To top this off, implement tooltips on the when an action is possible. The text must appear and disappear with a fading effect. Add a text presenting the objective at the beginning of the game that will change if the player wins or loses.

Again, tweak your game design and level design as much as possible to make this atmosphere as addictive and artistic as possible. Not a lot of code is required today, the grading will take it into consideration.