



Piscine Unity - Rush00

Hotline Miami

Staff staff@staff.42.fr

*Summary: This document contains the subject for Rush 00 for the „Piscine Unity”
from [42](#)*

Contents

I	Foreword	2
II	About teamwork	3
III	Specific instructions	4
IV	General Instructions	5
V	Mandatory part	7
V.1	Gameplay	7
V.2	Weapons	7
V.3	Enemies	8
V.4	Maps	8
V.5	Sounds	8
V.6	Menus	9
VI	Bonus part	10

Chapter I

Foreword



Figure I.1: This image represents a Unity collaboration. Unity is like a giant Jenga where you would steal pieces from your colleague to build your own part of the tower.



Figure I.2: This image represents Unity during a merge.

But rest assured that you will find in the next chapter some tips and advises to tackle properly teamwork.

Chapter II

About teamwork

The time has come to make your first Unity group project. As powerful as Unity is, it's not always a happy time when it comes to collaboration. To avoid losing precious hours/days of work and a lot of nerve crisis here are some advice to work properly as a group.

- Since we are a nice bunch you will find in the project's assets a .gitignore file which will allow you not to push file that should remain local.
- You should **NEVER** work at the same time on the same scene or the same prefab. Unity still has a lot of issues with the merge of these assets.
- Be cautious of the code of your partner when you update an object containing his script. A missing tag is enough to break down a feature.
- Communicate as much as much with your partner on modification you make on the project.
- Divide the work properly and avoid working on the same feature or the same game-play element at the same time to avoid conflicts.
- Commit as often as possible. When facing a conflict you will be able to roll back to a past functional version of the game.
- Put some "milestones" together, meaning a functional version at key stage of the development of the game and synchronise your repos before starting a new step. If you do this you should always have a stable version available if one of you breaks something and in worse case scenario you will be able to push your last valide "milestones" and avoid having a broken project.

Good luck! You will need it.

Chapter III

Specific instructions

- Read the whole subject. Some of the bonuses can be done easily if you do your script right from the start.
- For this rush the p2p correction will not start at the first error but to make sure you're not confused and stay organised, the grading will be sorted by priority order.
- For this rush, you are allowed to use C as well as the minilibx **only**. If you do, your code will have to be normed. To help you a little bit, you will exceptionally be allowed 6 functions per file.
- The last instruction is obviously a joke.

Chapter IV

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 V

Mandatory part

V.1 Gameplay

Hotline Miami is a game with a very nervous gameplay. Your main objective is to reproduce this spirit. But let's start with the basics. YOU have to put together a character movable in every directions using WASD (W -> Up, S -> Down, A -> Left, D -> Right). The character always look at the direction of the mouse. The camera must follow his position but not his rotation.

It's possible to get weapons on the ground by pressing E and drop the one currently equipped by using the **right click**. It is then obviously possible to shoot by using the **left click**. Every character only has one life point. If they're shoot they instantly die. The goal is to either reach the end of the level or kill everybody. If killed it's game over and the player has to restart le level.

V.2 Weapons

It must be possible to distinguish 2 types of weapons::

- Firearms that sends bullets in the mouse direction. They have limited ammunition (no charger) and can be depleted.
- Cutting weapons, are hand to hand weapons. They don't fire any bullets and aren't limited in use.

When the player drops a weapons, it's thrown in the mouse direction, in a more or less stylish way. We are expecting at least **4 different weapons**.

V.3 Enemies

Enemies and their AI is one of the core of this project. It will be them that will make this game fun or not. You will have to implement the following behavior:

- Enemies can see the player from far in front of them and a little bit from his back. When they spot the player they have to run shooting at him. They stop following after a certain time or when the player or the enemy is dead.
- Enemies can't go through walls, they know the doors location and can run after the player through several rooms with no problem.
- If the player shoots close to an enemy, it will hear it and start running after him even if not in the same room.
- Enemy's weapons don't have any ammunitions, they can fire as much as they want.
- Enemies cannot kill each other.
- Some enemies must patrol on a predetermined path.

V.4 Maps

You must at least make one level playable. Every level must be divided in several rooms. Rooms must be connected by doors. Enemies and player cannot go through walls or doors. Doors have to be freely openable and closable inside the level. Try to make the game fun to go through.

V.5 Sounds

What would be **Hotline Miami** without sound, you will have to attach quite a bit of attention to it. You will find in the asset package of the project the OST of the game. You will therefore have to put music in your whole game. Try to use various ones to avoid growing tired of it.



You have to use the original OST of the game. No your awesome techno sample will not be better than it. Who would want to put something else anyway?

You will also have to put sound effects for the following elements:

- When the player or an ennemy is killed.
- When a weapon is picked up.
- Every weapon must have a different sound more or less consistant.
- When winning or losing.

V.6 Menus

To make all of this awesome you will have to design menus. We are at least expecting the following of you:

- A very stylish main menu with an animated logo using the hotline miami font. It must sting the eyse a little bit but be nice at the same time. It must be possible to launch or exit the game from that menu.
- A pop-up menu that will be displayed over the scene when the player wins or loses the game. The text must obviously be different depending on that last parameter.
- An in-game GUI informing the player of the remaining ammunicions.



All of this is a lot of very precise work. Note that chapters are ordered in priority order so you can know where you are at and know what will be important during the p2p correction.

Chapter VI

Bonus part

If you were bale to go through the last part, well congratulation. Now you can add some more interesting gameplay elements. You are globally able to implement anything you think is cool and can make you game better or improve the ambiance or the fun.

If you are out of ideas, find some in the list below:

- The possibility to stun ennemies when throwing weapons at them or when slamming doors at them.
- More maps, more weapons, always more.
- Various different ennemies that spawn with a different head and/or weapon.
- Sub categories of firearms (long range, machinguns, pistols, ...).
- The katana kill ennemies when thrown at them.

Possibilities are endless anyway, have fun and create your dream **Hotline Miami**.