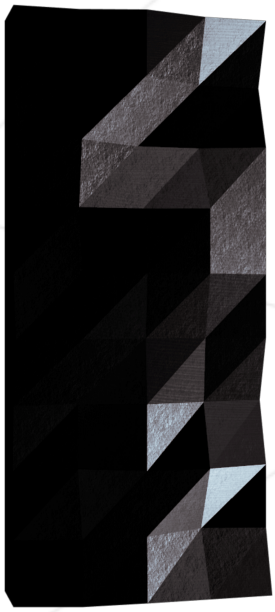

Piscine Unity - Day 08

Animations

Staff staff@staff.42.fr

Summary: This document contains the subject for Day 08 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 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 II

Foreword

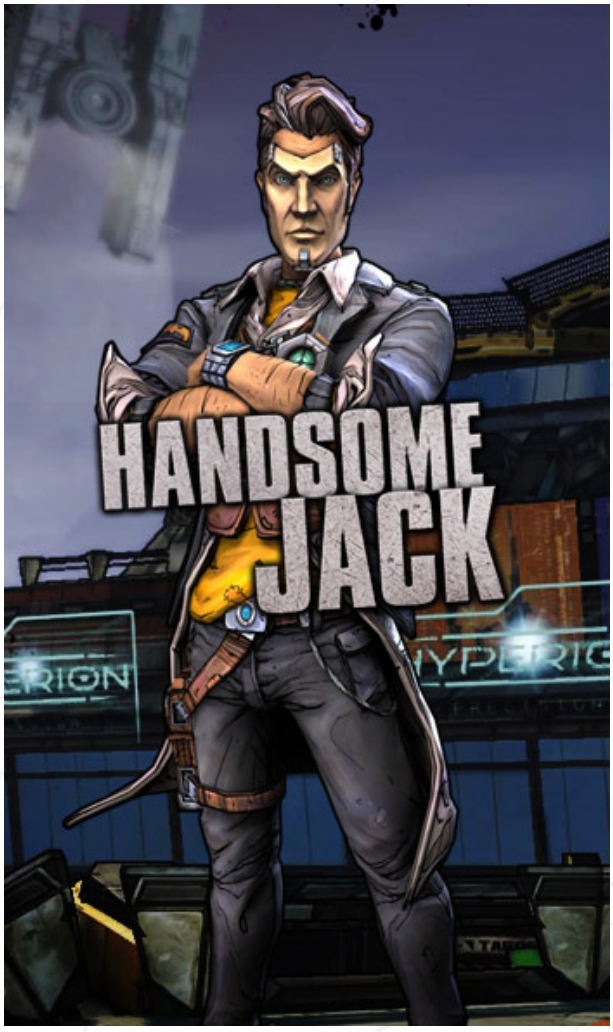

Figure II.1: "My day? Not bad, I bough myself a diamond pony because I'm full of dough. Well you know, It's cool, OK, bye."

Chapter III

Exercice 00 : Scenery implementation

	Exercise 00
Exercice 00 : Scenery implementation	
Files to turn in : `The "ex00" scene and everything that seems relevant`	
Forbidden functions : `None`	
Remarks : `n/a`	

WARNING! WARNING! Now that I have you ATTENTION! here are couple of precisions on today's exercises. Today we will do a technical demo don't lose time on level design, scenery or do it at the end of the day. Today's assets are really cool, it's true but don't worry you will have other occasions to use them!

For the whole project work on a human scale. Use a cube of 1 unit on the side and put it height at 2 to have an idea of the size of a character and use it as a template to create your screnery.

For this first exercise you will have to create a simple, globally flat lanscape, with couple of bumps and slopes more or less steep that will be used later.

Finally you will have to create a building like a shed for example. Its size must be reasonable, not like a small house, and not like a cathedral either! A shed will do the job. It must obviously have a door, opened.

If you work fast you should have finished this exercises in 30 min.
If you spend more than 1h it’s probably a bad start for today !

Chapter IV

Exercice 01 : Maya

	Exercise 01
Exercice 01 : Maya	
Files to turn in : `The "ex00" scene and everything that seems relevant`	
Forbidden functions : `None`	
Remarks : `n/a`	

`The provided assets today aren't equally balanced in their scale, and aren't all ready to be used in their state. It's up to you to update them to give them a unity by using everything you learned until now.`

Now that the basic scenery is placed, you will have to put Maya, our main character, into it, who will be controlled by the player.

The model is provided in the assets as well as a serie of animations. You need to create her an animator in 4 states:

- An idle state that will be the default state when the player doesn't do anything.
- A running state that will be triggered when clicking somewhere on the landscaepe.
- An attack state that will be triggered when clicking on an enemy.
- A dead state that will be triggered when Maya's HP reach 0

We will come back on the enemies in the next exercise for now prepare just an "attack" state that will be triggered when Maya attacks.

You can also put the camera always centered on Maya and diving. The idea is to create an "Hack and Slash" camera, look for ingame Diablo or Torchlight images on google

to get an idea.

You must create a NavMesh and put a NavMeshAgent on Maya to manage her movements. Your NavMesh will have to be properly configured so that the character moves in a consistent way. You must have impassable slopes on your landscape to force your character to turn around. If it's not the case don't hesitate to update your lanscape.

Once your NavMesh created, you will have to manage your character's movements and her animation transition from idle to run when the player clicks somewhere on the landscape. Obviously the camera must follow the movement.

Chapter V

Exercice 02 : Enemies and non artificial intelligence

	Exercise 02
Exercice 02 : Enemies and non artificial intelligence	
Files to turn in : `The "ex00" scene and everything that seems relevant`	
Forbidden functions : `None`	
Remarks : `n/a`	

Let's now move on to enemies and combat! To simplify things we will be original and go for: zombies.

Create an EnemySpawners. It's an empty GameObjects with a script that takes as a parameter a list of enemy prefabs and that make randomly one spawn from the list. You must create 2 enemy prefabs that will use the male and female models, provided in the assets.

These enemies must:

- Have a NavMeshAgent to manage their movements in the scenery.
- Have an animator with 4 states: idle, run, attack, dead.
- Have a detection zone to know when the player is close to them and run towards him to attack.
- Have a scrip to manage all of this with a simple HP management. Every enemy has 3 HP.

You have to put these EnemySpawners on the map and ensure that when an enemy is

killed another one spawns a little bit of time after from the corresponding spawner. The "a little bit of time after" is up to you, the aim is to avoid having a too small time to avoid the player staying close to the spawner to farm enemies but not too long to have a well populated map.

You have to manage combat. For now Maya is immortal but you must ensure that enemies lose a life when she attacks and therefore die after 3 attacks. Make sure that the damage is applied when the weapon touches and not when the animation starts.

To attack the player must click on an enemy. If Maya is in attack range she automatically attacks in hand-to-hand combat. If she is to far away from the enemy she must first run towards him and then attack. If the player keeps the mouse button pressed after clicking the enemy, Maya must keep on attacking until the enemy is dead even if the mouse cursor goes out of the enemy hitbox.

When an enemy has no more HP, run his death animation. You will have to create a coroutine so that when an enemy is on the ground since 2 seconds, he starts sinking into the ground progressively to destroy him completely when the player doesn't see him anymore, instead of making him disappear instantly. Don't use any other way to destroy your enemies, to make them disapear will create some problems and the resolution is the aim of the exercise.

Chapter VI

Exercice 03 : Stats, because we need some

	Exercise 03
Exercice 03 : Stats, because we need some	
Files to turn in : `The "ex00" scene and everything that seems relevant`	
Forbidden functions : `None`	
Remarks : `n/a`	

Now that you have a gameplay let's be serious and create an advanced stats and combat system.

`From this point onward we will see a in a very detailled way the gameplay mecanics of a hack and slash. Some mecanics being quite quite difficult to explain in a succint way, if you have any doubt try the provided demo to get an idea of the expected behaviour. If you already played DiabloII/Torchlight you should be in familiar ground. If you don't know hack and slash or only the recent one such as DiabloIII or PoE, try the demo, really.`

Every character in the game (Maya and the enemies must have the following stats:

- 3 main stats: Strengh(STR), Agility(AGI), Constitution(CON)
- An Armor stat that will subjectively attributed for now and that will be modified later with equipment.
- An HP stat that must be equal to 5 * CON
- A minDamage stat that must be equal to STR / 2 and a maxDamage stat that

must be minDamage + 4. Like the armor these basic stats will be updated with equipment later.

- A Level stat that correspond their level.
- An XP stat. For Maya it will be the accumulated xp by killing enemies, for the enemies it's their individual value in xp.
- A money stat. For Maya it's the credits she accumulated, for the enemies it's their average value in credits.

You must choose the values you want for you characters/enemies but for the level 1 a good stat will be between 10 and 20, per stat.

You must add as well a variable on Maya that will describe the number of xp points required for her to level up.

Now that characters have real stats we will update combats to make them less basic. You must implement the following formulas for your combats (don't spend time trying to tweak them, it's out of subject)

Hit chance: hit = 75 + AGI - Target.AGI

Basic damage: Random entre minDamage et maxDamage

Final damage calculation: baseDamage * (1 - Target.Armor/200). For obvious reasons, the armor cannot go over 150-170. No logarithmic decrease of damages for now, we will see that later.

Maya is still invisible no need to run the method that applies damages when an enemy attacks her.

Chapter VII

Exercice 04 : Death and interface

	Exercise 04
Exercice 04 : Death and interface	
Files to turn in : `The "ex00" scene and everything that seems relevant`	
Forbidden functions : `None`	
Remarks : `n/a`	

You now have to create the whole interface so that the player has a better feedback of the combats:

- An HP bar for Maya, with the exact number of HP displayed on the side.
- An xp bar, that will be filled progressively as Maya wins experience. You must also display the current number of xp Maya has + "/" + the number of xp required to level up.
- Maya's current level close to her HP bar.
- The selected enemy's HP bar (see below) at the top of the screenm as well as its name and level.

An ennemy can be "selected" in two cases:

- If the player pass the mouse on him. In this case the information are displayed at the top of the screen and must disappear when the cursor of the mouse isn't on the enemy anymore.
- If Maya is attacking. In this case the information are displayed at the top of the screen, but must remain displayed (even if the cursor isn't on the ennemy anymore)

until it dies, or until Maya stops attacking.

Now that we finally have a visual feedback of our character it's time to make her die. Yeah the God Mode is fun for 5 minutes... You now have to make Maya mortal and make her lose the appropriate number of HP when she is attacked by enemies. If Maya dies, you must display and interface/message saying so but mostly ensure that nothing crashes!

Don't make enemies that hits too hard to avoid Maya to die in 4 hits or less because for now she has no way of regaining her HP, it will be done in the next exercise. Ideally she should be able to kill an enemy in 2-4 hits and be able to sustain around twenty attacks to have something playable.

Chapter VIII

Exercice 05 : Level up !

	Exercise 05
Exercice 05 : Level up !	
Turn-in directory : $ex05/$	
Files to turn in : `The "ex00" scene and everything that seems relevant`	
Forbidden functions : `None`	
Remarks : `n/a`	

The time has come for the final touches. Let's start with the interface. You have to create a window that can be displayed by pressing "C" and that displays the following elements:

- Character name
- The 3 main stats
- The min/max attacks
- The armor
- The level
- The xp points
- The xp required to level up
- The credits

You now have to manage the player's xp point. Each time Maya kills an enemy she must win xp points and when her xp reaches the value required she must level up. In this case you must create a visual effect so that the player is clearly informed (don't lose hours on this though), you must put back Maya's HP to the max and give her 5 points

for the player to use. Her xp point must be reseted and the score required to level up again increased by 150%.

A clickable button must appear on the HUD close to Maya's HP if there is any habil-ity points remaining to be attributed. When clicked the character windows must appear, exactly as if the player pressed "C". A clickable button is now available close to every main stats. When clicked on a point is deducted from the available points and added to the corresponding stat.

You have to manage the enemies' level as well (in a simple way). For every level over 1, the enemy must have stats boosted by 15%. For example an enemy of level 3 will have 130% more stats than an enemy of level 1.

Even if this isn't an interesting usage from a game design perspective, you must ensure that enemies that spawn are of the same level than Maya, to have a quick progression in the game. Remember that it's a technical demo.

Finally put together a globe/life potion system together, that will randomly fall when an enemy is killed. They automatically heal Maya of 30% of her maximum life when she walks on them.

The different exercises of today (quite busy!) end here but if you still have time you are free to put together a loot and equipment system together. You can also implement the credits management that the enemies would drop when dying. There is already couple of stats ready in the enemies and Maya's scripts. Note that everything you add will be considered personnal bonus and will not be rewarded during p2p.