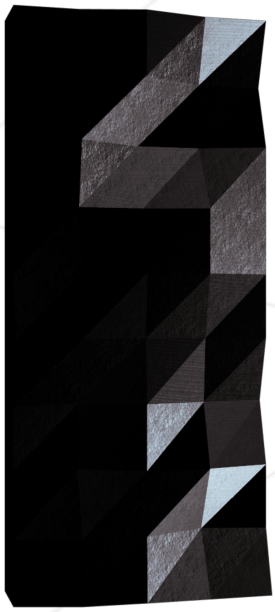


Unity Piscine - Module06

Navmesh, Light, Sound and Camera

Summary: This document contains the Module06 project brief for the Unity Piscine.

Version: 2

Contents

Chapter I

Instructions

- If you have trouble installing the required tools for your project on the 42 computers, you may use a virtual machine. In this case, you must:
  - Install the virtual machine software on your computer.
  - Install the operating system of your choice.
  - Install the necessary tools for your project.
  - Ensure that you have enough space on your session to install all of it;
  - Have everything installed before the evaluation.
- Only this page will serve as reference. Do not rely on rumors.
- Carefully read the entire document before starting.
- Your exercises will be evaluated by your fellow piscine participants.
- This document is your reference. Do not blindly trust demos or example pictures, which may include unnecessary additions.
- Got a question? Ask the peer to your right. If not, try the one on your left.
- By Odin, by Thor! Use your brain!!!

Intra shows the date and time when your repositories close. This also marks the beginning of the peer-evaluation period for that piscine day. The peer-evaluation lasts exactly 24 hours. After that, any missing evaluations will be scored as 0.

Chapter II

Foreword

John Lemon, like all small felines, doesn't like going to bed at night.
But one evening, during one of his nightly escapades, despite his excellent night vision, he failed to see a root sticking out of the ground and got his paw caught in it.
He stumbled and put out both paws to catch himself! But at that moment, the ground gave way beneath his feet, and poor John found himself in an endless fall.

After what felt like an eternity, he finally felt solid ground beneath him.
He found himself in total darkness. As he looked around, he realized he was in a cramped space. Panicking, he flailed in all directions until the wall in front of him gave way.
He fell to the ground and realized that the place where he had been trapped just moments before was actually a cupboard.

Dim lighting, eerie noises—John had not fallen into Wonderland.

In this hostile environment, stealth is the key!

Chapter III

Exercise 00: John Lemon in Nightmareland

	Exercise :
Exercise 00: John Lemon in Nightmareland	
Turn-in directory : `unityModule06`	
Required elements : `The "Stage1" scenes, scripts, and any relevant assets`	
Forbidden functions : `None`	

Poor John, he'll need to be stealthy!

To begin his adventure, start by creating his character.
You can find him in Models/Character.

John must:

- Have idle and walking animations.
- Be able to move.

Create the stage:

- Design the room where John arrives from the wardrobe, along with other rooms and corridors the player must navigate stealthily.
  Don't create too many rooms; manage your time wisely.
- Place doors on certain rooms. Some should open when the player approaches.
- Customize rooms with decorations to represent bedrooms, bathrooms, dining rooms, etc.
- Include a room that can only be accessed with 3 keys. This room should contain a wardrobe similar to the one where John arrived. The player must use it to finish the stage.

- Place one key in each of three different rooms. The player must find these keys to unlock the door leading to the exit wardrobe.

Your stage might look like this:

Chapter IV

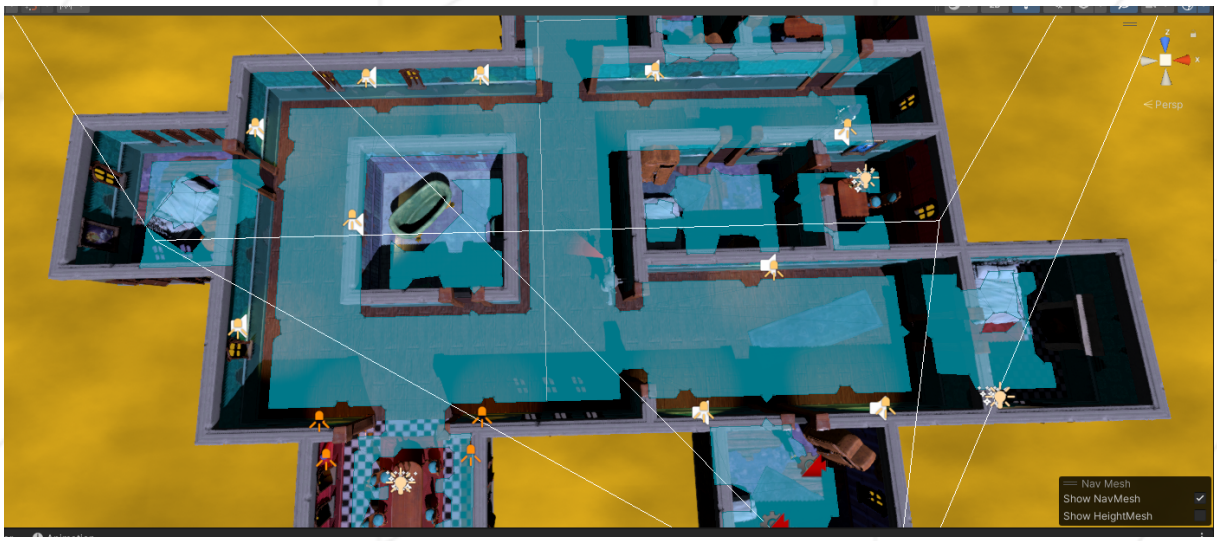
Exercise 01: Finding Your Way

	Exercise :
Exercise 01: Finding Your Way	
Turn-in directory : `unityModule06`	
Required elements : `The "Stage1" scenes, scripts, and any relevant assets`	
Forbidden functions : `None`	

Now that your playground is ready, it's time to add enemies: gargoyles and ghosts.
You can find the ghost and gargoyle models in the Models folder.
Make sure that the ghosts can move in a believable way.

First, create a **`NavMesh`** for the environment where the enemies will move.

Your stage's NavMesh might look like this:



Once the `NavMesh` is created, configure the enemies to use it.

Ghost behavior:

- Should have a walking animation.
- Has a detection zone that senses the player when entered.
- Must move between two points using the NavMesh.
  (It might be a good idea to look into the**NavMesh Agent** and its `SetDestination` method).

Gargoyle behavior:

- Must have an idle animation.
- The player cannot pass through the gargoyle.
- Has a detection zone represented by the red torchlight. It detects the player if they enter it.

When a ghost detects the player:

- The ghost chases the player briefly, then returns to its original position.
- If the ghost catches the player, the player faints and the stage restarts.
- You must have at least 4 ghosts in your scene.

When a gargoyle detects the player:

- All ghosts are alerted and move toward the player.
- You must have at least 2 gargoyles in your scene.

Chapter V

Exercise 02: Question of Viewpoint

	Exercise :
Exercise 02: Question of Viewpoint	
Turn-in directory : `unityModule06`	
Required elements : `The "Stage1" scenes, scripts, and any relevant assets`	
Forbidden functions : `None`	

Camera issues can ruin a game experience; Yes you know that!
How many times have you raged at a game with a chaotic camera system?

Now it's your turn to make a good one.

You may import the **Cinemachine**.
You're free to use it or not.

Your camera must support two viewpoints:

- Third-Person Shooter (TPS).
- First-Person Shooter (FPS).

Players can switch views by pressing a key (e.g., the C key).
TPS camera:

- Positioned above and behind the player.
- Follows the player.
- May or may not rotate with the player, based on your preference.
- Player moves using the WASD keys (or, 'ZQSD' on AZERTY keyboards).

FPS camera:

- Positioned at the player's eye level.
- The player can look around using the mouse.
- When pressing the 'W' key, the player moves in the direction they are looking.

Like Cinemachine, Unity also provides a powerful Input System that can help manage your character's movements. Using it is optional.

Chapter VI

Exercise 03: Ending the Game

	Exercise :
Exercise 03: Ending the Game	
Turn-in directory : `unityModule06`	
Required elements : `The "Stage1" scenes, scripts, and any relevant assets`	
Forbidden functions : `None`	

You can create a simple game ending.
Two images are available in the Textures/UI folder.

Create two fade-in/fade-out animations:

- One for when the player is caught and returned to the beginning, using the appropriate image.
- One for when the player wins the game, also with the appropriate image.

Chapter VII

Exercise 04: Light and Sound

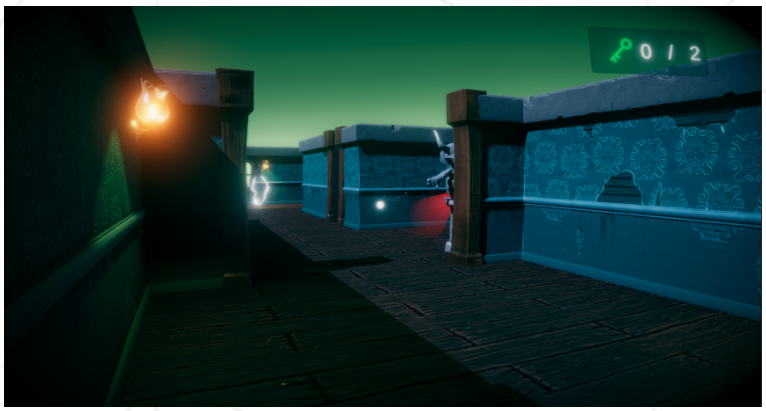
	Exercise :
Exercise 04: Light and Sound	
Turn-in directory : `unityModule06`	
Required elements : `The "Stage1" scene, GlobalPostProcess GameObject, and any relevant assets`	
Forbidden functions : `None`	

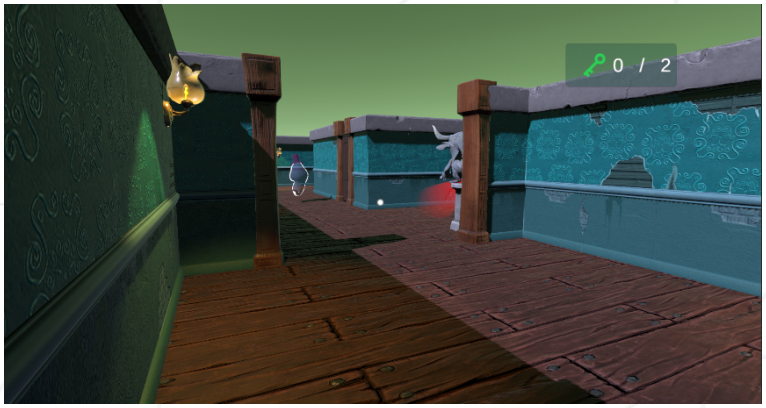
Lighting and sound are crucial for building atmosphere.
Unfortunately for John, he has landed in a dark, gloomy place.

You must add a **GlobalPostProcess** `GameObject` to your scene that will include post-processing effects.
The `Post-processing` components will be very helpful.

Adjusting the properties of your Directional Light and tweaking the Lighting Settings will also help.

Here's the difference with and without post-processing:





For audio, you may have noticed some sounds are already integrated, but that's not enough.

- Add ambient sound.
- Add a sound effect when the player faints.
- Add a sound effect when the player wins.
- Add footstep sounds for your character.
- Add ghost sounds, audible only when nearby.

Chapter VIII

Submission and Peer Evaluation

Submit your assignment in your `Git` repository as usual. Only the work inside your repository will be evaluated during the defense. Don't hesitate to double-check the names of your folders and files to make sure everything is correct.

You should not upload the entire Unity project to Git, as this can unnecessarily increase the size of your repository.

- Make sure Unity saves as many files as possible in text format rather than binary. In Unity, go to Edit > Project Settings > Editor. Under Asset Serialization, set it to Force Text.
- Ensure that the .gitignore file automatically generated by Unity is present.

The evaluation will take place on the computer of the learner or group being evaluated.