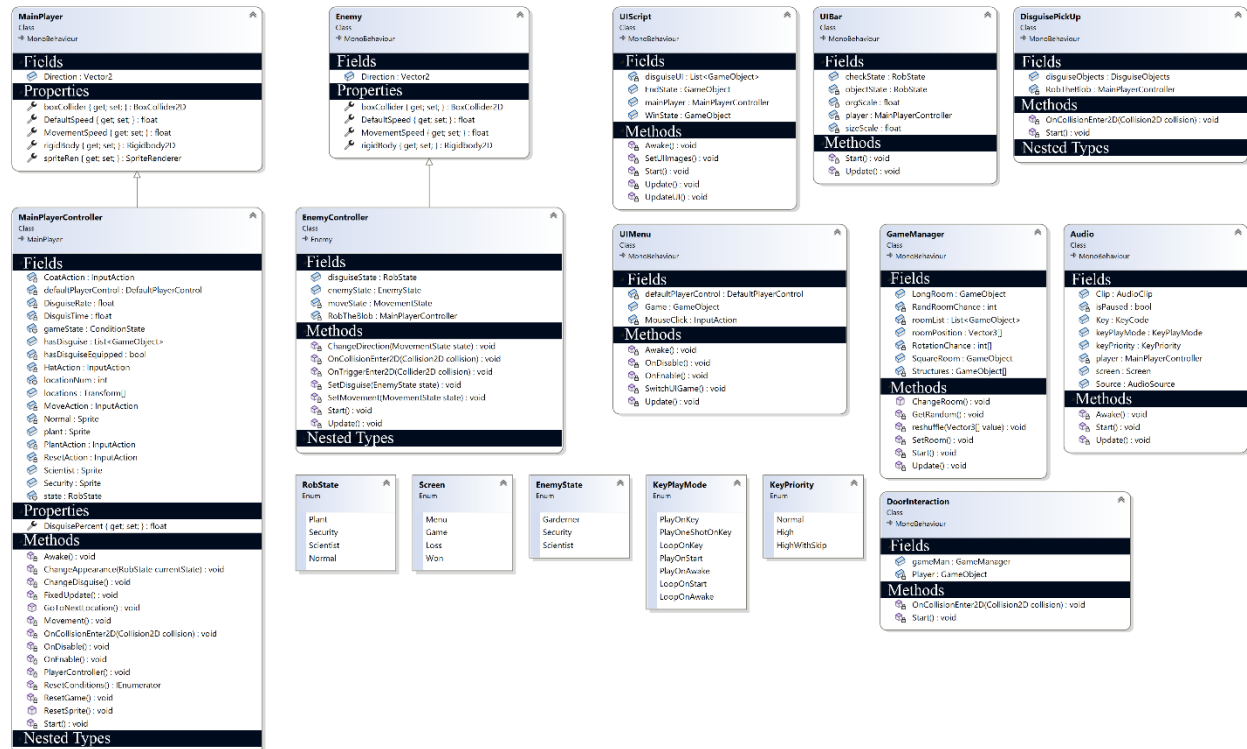


Evy Avila

Postmortem

Class UML diagram:



The black bars had blocked the text for each diagram. Thus, I had to manually insert the text to help visually see the sections.

The UML consist of the main classes used for the game. The main hierarchy used with essential items are the Enemy and Main Character classes. I have three sets of UI scripts as I was challenging myself to use gameobjects instead of the Unity Canvas UI. They each have one main function, or Separations of Concern, for handling the menu, gameplay UI, and bar line. I have a GameManager class that handles the randomization and placements of the rooms placed throughout the level. Two classes called DoorInteraction and DisguisePickUp mainly check if the player has interacted with the objects. Finally, I have an audio class that inserts the audio and

plays. Most of the code is not my own, however, I did edit some of the functions to take into account stopping the loop.

What patterns did you implement in your game:

I attempted to implement the strategy pattern as to help organize the dependencies, however, I'm not completely sure if I did it correctly. In general, I didn't use as much patterns in general.

What challenges did you have making your game components? How did you overcome them?

One of the main challenges I had was coming up with a way of randomizing or auto generating the level so that it was unique each time the player played. The way I attempted it was by inserting coordinates into an array, and for each element, insert the room at random and at a random rotation. Another challenge I had was having the bar line not shrinking correctly. I attempted to manually insert each value for how long the bar line goes down. However, it wasn't consisted with the other bars. After discussing with Jeff, I made some adjustments to allow the time to go down correctly. As for the bar line itself for UI, I wasn't sure how to get the line to shrink without affecting the entire transform. Even so, I was able to quickly create an empty GameObject parent that shrinks itself without the bar line being affected.

Rate how reusable the game codebase is? What percentage is specific to the game? What percentage is specific to the platform (monogame/unity)?

25% reusable codebase. 70% is specific to the game. 85% is specific to the platform.

How maintainable is your code? Is it easy to add more enemies, pickups, weapons, levels etc?

Some of the code is maintainable. For example, the GameManager can include more room types to be added for randomized placement. As for the rest, it is very heavy in the manual insert.

Did you finish your game? Is there any outstanding technical debt?

I did finish the game. I was able to insert all of my main functions, and have it more polished by updating the art and music. One of the main technical debts I had was not organizing some of the classes correctly as they should have. For example, the door and disguise pick up classes should have derived from a base class. Additionally, the enemy and player class should have one higher base class, since they contain similar components. As for the room coordinates, it would have been better to have a range of options to place the rooms instead of manually inserting them. It's hard to tell where the rooms will be locked without a reference. I have the menu section add a source component instead of using the Audio script as preferred. The reason for this was because the script required the player script, which is deactivated during the time. Since it was a last minute add, I didn't have time to fix it. If given more time, I could have find a way to avoid using the player as an indicator. Finally, the bar line for the plant UI goes down slower than the other bar lines. I'm not sure for the reason as to why it doesn't go down

Dev Logs:

Dev Log 1: I've created a POC in Unity for my game Sneak.

I first worked on creating an "auto generated" level. It mainly has set locations and randomly sets the game object and rotation. Since I was testing in a small location, I may expand the area for

the player to travel to the next level by teleporting. As a way to feel as if they moved to another floor.

Next, I created a `MainPlayerController` script that consist of simple movement and the ability to switch between disguises. I used the old `Input` system as temporary use to make sure the commands and changes were correct.

Finally, I created a door interaction for the player to pass by and enter the next level.

For the next update, I aim to:

- Insert enemies
- Add function to earn disguises
- Expand the level
- Create ending states (Win/Lose)

Dev Log 2: I've created a VS in Unity for my game Sneak. I was able to complete the main objectives from my checklist.

I first worked on adding the enemies into the game. At the current version, the enemies don't move, but are placed near the door where the players needs to enter. Their purpose is to catch the player who isn't in disguise. If the player is close to their range without a disguise or is wearing the wrong disguise will cause a game over.

Next, I added a function where the player can earn and use the disguises they find during the level. I also fixed the issue where the player can easily switch between disguises. Now, if the player is wearing a disguise, they cannot switch until the player returns back to normal.

Next, I expanded the level. It's in a similar format with the walls and set positions. The main addition is the player changing their position to the next floor level.

Finally, I created a simple end states for when the player wins or loses. The player wins by making to the last floor or level. The player loses if they are caught by the enemies. I added a condition where, if the player wins or loses, the controller method is not called.

For the next update, I aim to:

- Convert the controls to the new Input System
- Add an interface
- Expand the level further
- Update the art assets

~~~~~

Dev Log 3:

I've created a MVP in Unity for my game Sneak.

I first updated the Input System to use the new version. While it took some time for converting and changing some of the methods, I only had one block. I wanted to use one Action with three binding paths for the disguises. Where, it would hold three keys/buttons that would check if the specific button was pressed. However, while I was searching online and experimenting, nothing was working. Instead, as a technical debt, I created three separate Actions. This resulted me into having to check the value three times, which is not correct. I'll have to continue working on it.

Next, I added a simple interface. The main purpose is to show which disguises the player found and are able to use. I attempted to include text with the Text Mesh Pro, however, I didn't enjoy how much the text would move when adjusting the Game View. Even while it was anchored, I decided to remove it entirely and come up with a better solution at a later time.

Afterwards, I updated the Enemies by adding movement. I wanted to have the enemies feel a real threat as some will walk around or standing. Thus, it help create more of a fun challenge. I had an issue where the player would get caught at the start, which I change a few conditions for it to stop.

Finally, I expanding the levels by making them larger and longer. It gave room for both the enemies and the walls to be placed. Thus, looking more active.

The only point I didn't get to update was the art assets. While it was on my To Do List, it wasn't a priority to have completed.

For the Final update, I aim to:

- Add an additional obstacle
- Create a title screen
- Update the art Create and insert music
- Create ending transitions: A win and lose end screen

~~~~~

Dev Log 4:

I've created the final version in Unity for my game Sneak.

I was able to complete my main objectives from the proposal and personal additions to the game.

The only objective I did not complete was adding additional obstacle to the game. I also edited the issue with the bar line counting down correctly.

As for the update in general, I mainly focused on polishing the art, adding music, and adding screen images for the player to see.

~~~~~