# Game Engine Design & implementation Course Project

Video Report Link: https://youtu.be/ZYSkvroyG-c

## Team and Responsibilities

Manvir Punglia (100828507):

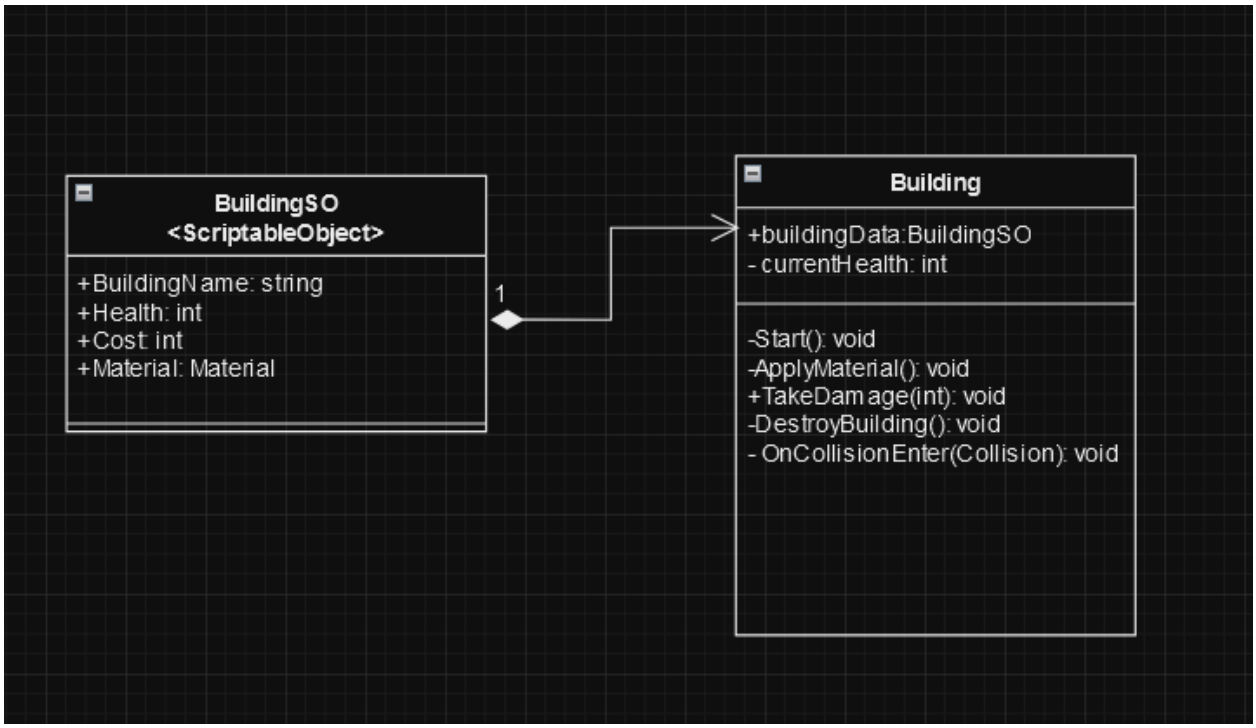
- Building Flyweight
- ConfigParser DLL
- The UMLs for DLL and Building Flyweight

Saad Khan (100829159):

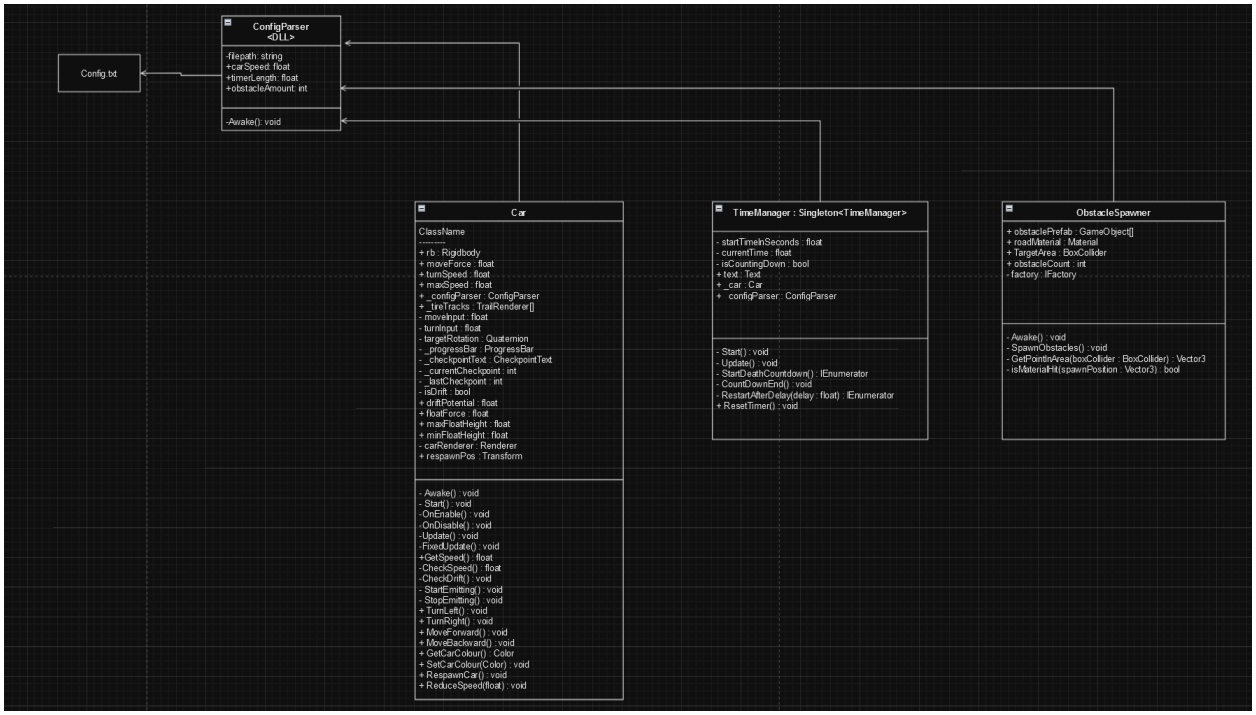
- Obstacle Flyweight
- Dirty Flag implementation
- VFX
- Car modification

## UMLs

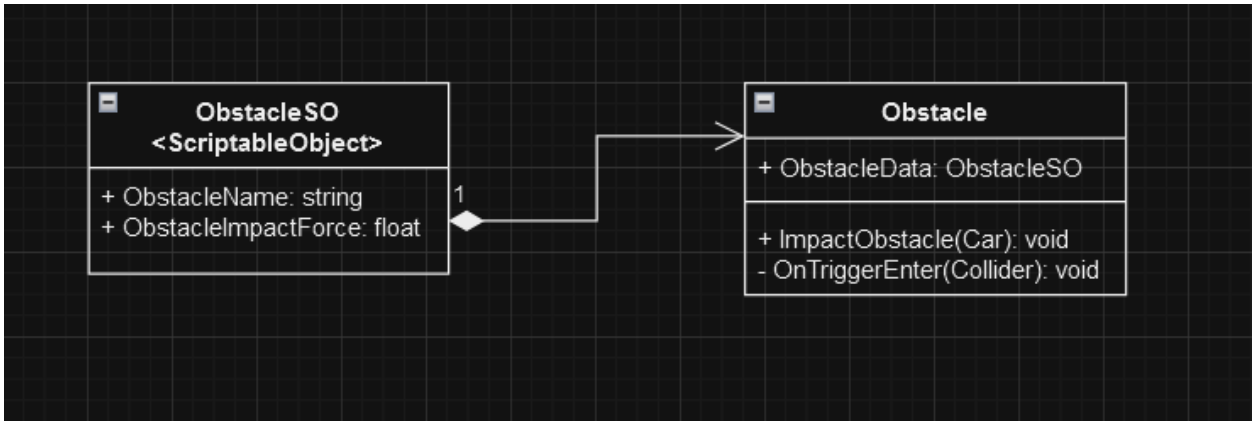
Building Flyweight



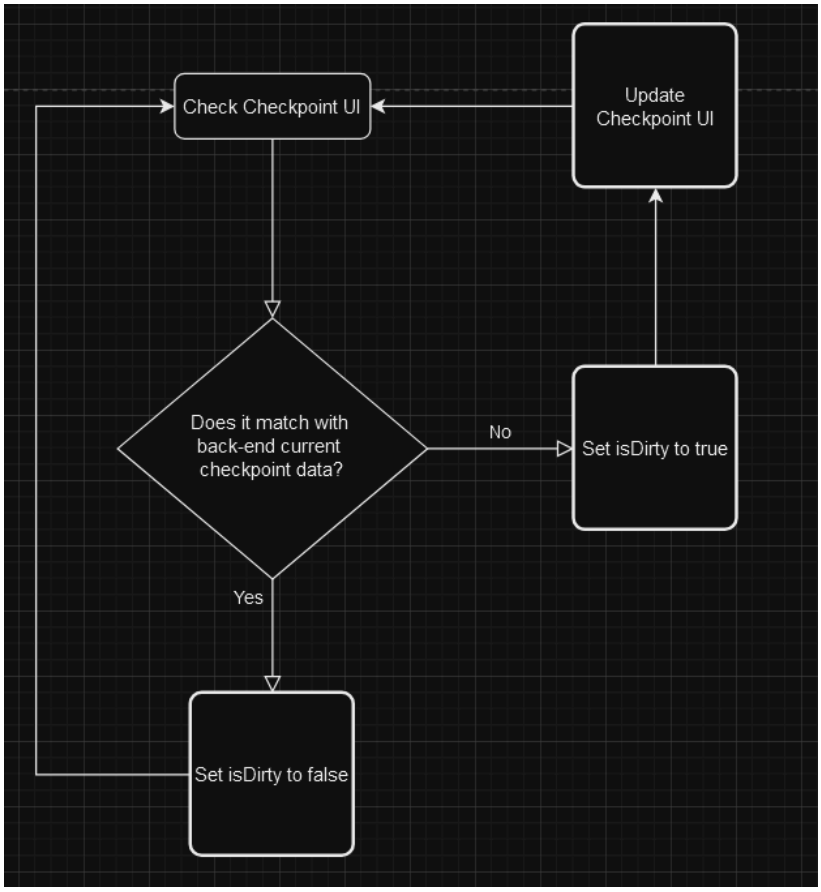
## ConfigParser DLL




## Obstacle Flyweight

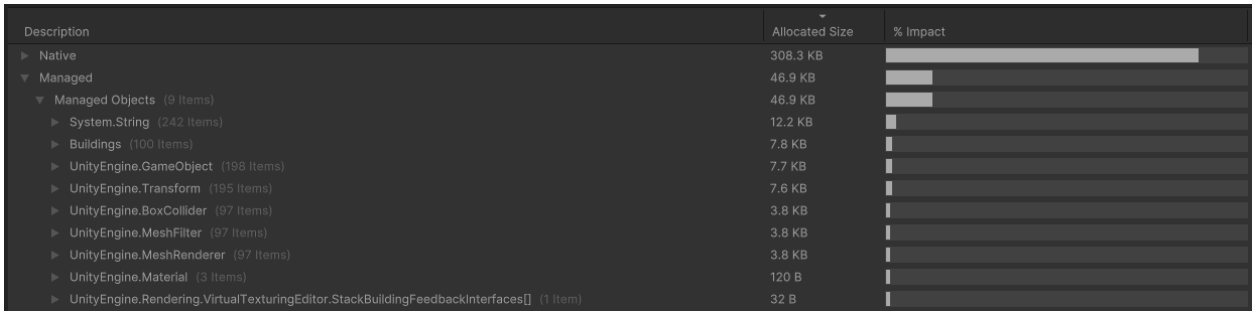


## Checkpoint Dirty Flag



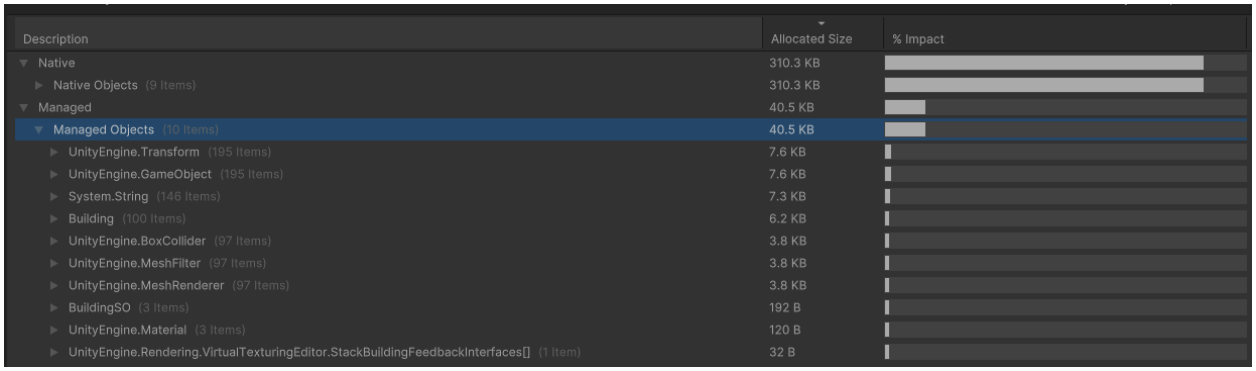

# Profiler:

## Building Flyweight (Unoptimized)

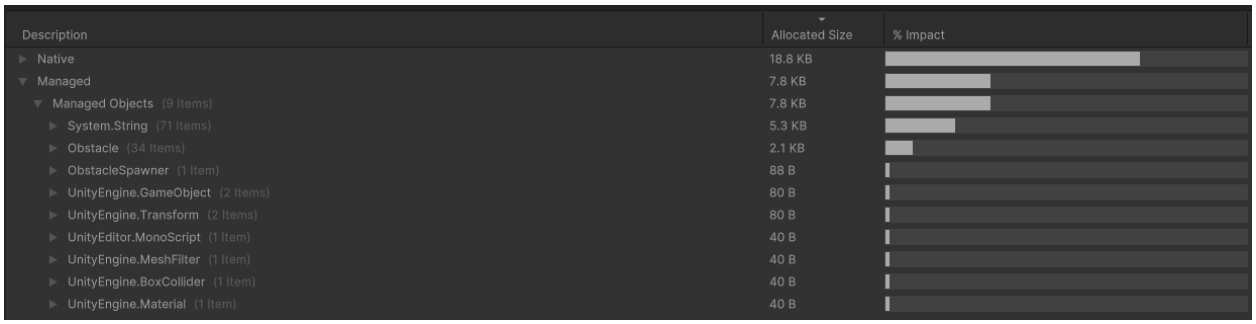


| Description | Allocated Size | % Impact |
|---|---|---|
| Native | 308.3 KB | |
| Managed | 46.9 KB | |
| Managed Objects (9 Items) | 46.9 KB | |
| System.String (242 Items) | 12.2 KB | |
| Buildings (100 Items) | 7.8 KB | |
| UnityEngine.GameObject (198 Items) | 7.7 KB | |
| UnityEngine.Transform (195 Items) | 7.6 KB | |
| UnityEngine.BoxCollider (97 Items) | 3.8 KB | |
| UnityEngine.MeshFilter (97 Items) | 3.8 KB | |
| UnityEngine.MeshRenderer (97 Items) | 3.8 KB | |
| UnityEngine.Material (3 Items) | 120 B | |
| UnityEngine.Rendering.VirtualTexturingEditor.StackBuildingFeedbackInterfaces[] (1 Item) | 32 B | |

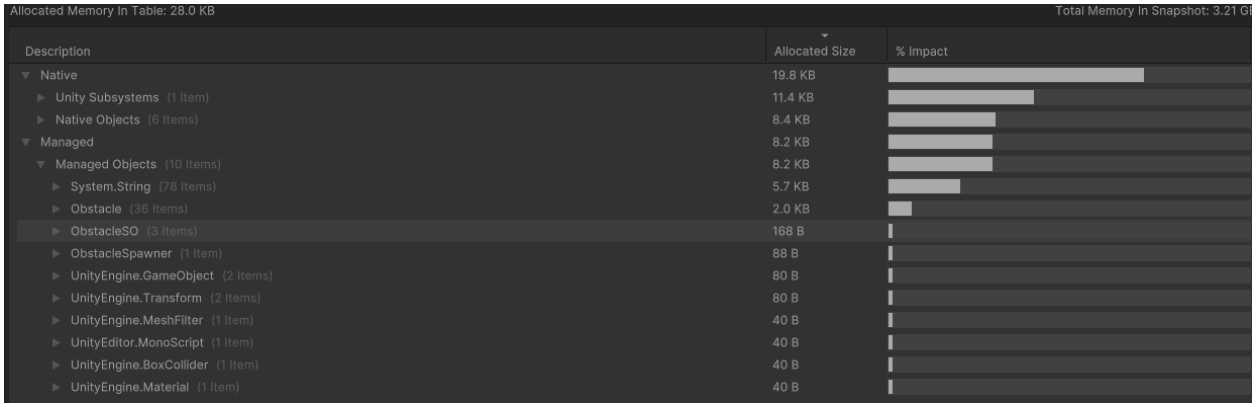
Building Flyweight (Optimized)




Obstacle Flyweight (Unoptimized)




Obstacle Flyweight (Optimized)




# Use of third-party resources:

Use of generative AI | Chat GPT:
Chat GPT was used in the creation of the DLL. It was given a text file like the one in the streamable assets folder called config and told to create code which would allow the information to be read and the tooltips to be ignored. I further edited the script by making it a DLL, Making the read information be stored

in accessible variables for other scripts to use and I added functionality to the information like setting the screen to the resolution entered in the config file.

Low Poly Cars | Toy Cars:
https://assetstore.unity.com/packages/3d/vehicles/land/low-poly-cars-toy-cars-258470
This asset was used to streamline the creation of the car model. To focus on optimizing and completing our project, it was in our best interest to use a car model rather than create one ourselves. However, we did modify the car model slightly to fit in the context of our game. We added a bomb game object and attached it to the back of the car. For background context, the player character (car) will explode if you don't reach the next checkpoint. We created more visual feedback by showing the bomb that exploded.

EasyRoads3D: https://www.easyroads3d.com/
It was used as a tool to create the road structure.

Checkpoint System: https://www.youtube.com/watch?v=IOYNg6v9sfc&t=531s
The reason behind using this system is that it creates a robust experience that fulfills the needs of what we're trying to achieve. The contributions I made were implementing a function that resets the checkpoints, this function will be invoked every time the player explodes and respawns. I also created a flag that will check whether the checkpoint objects are null, this way, the game won't throw an error and will notify me if the checkpoint objects don't fill the list automatically for any reason.

CineMachine: https://unity.com/unity/features/editor/art-and-design/cinemachine
It was used to provide a smooth camera when turning and moving.

**Unity. (2021).** ***Level Up Your Code With Game Programming Patterns*** **(LTS EDITION).**

**https://unity.com/resources/level-up-your-code-with-game-programming-patterns**

Used to help understand design patterns, UMLs and their implementation into unity