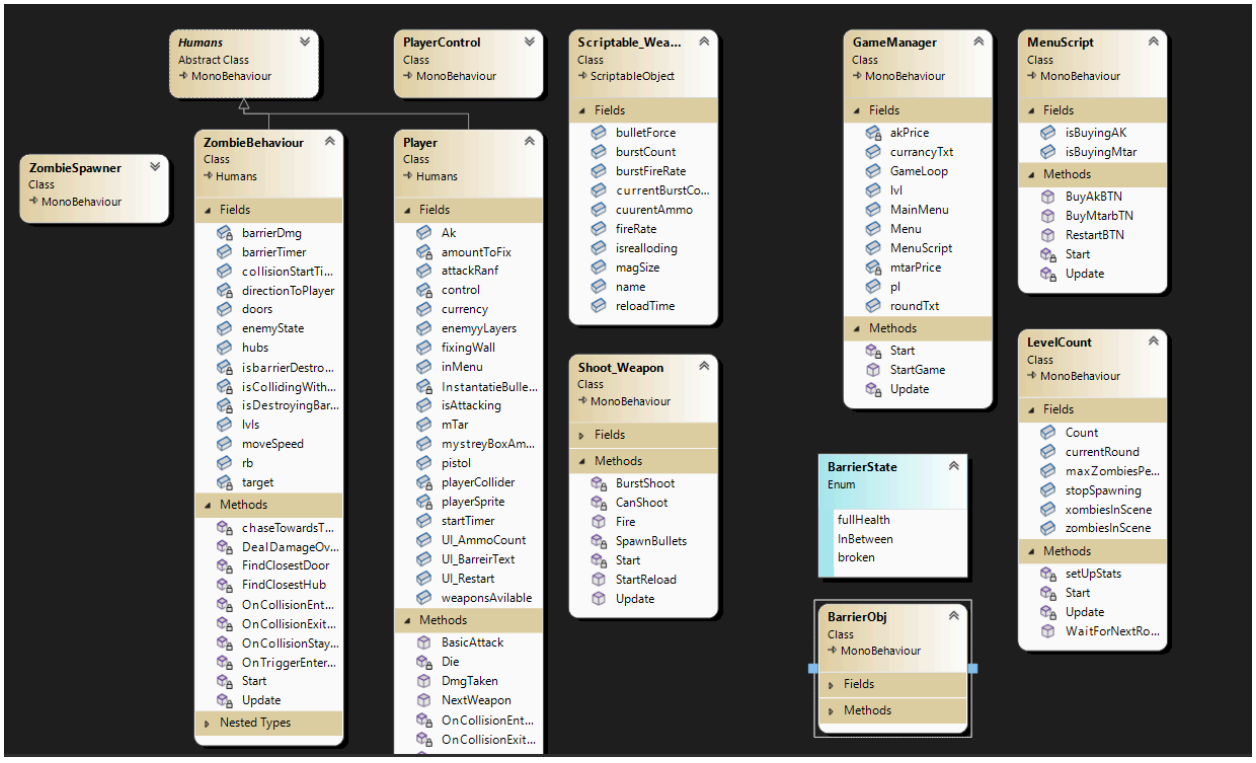
Jazib Shakeel
Postmortem - ZAF

- Class UML diagram with descriptions of class responsibilities.
- What patterns did you implement in your game. How did the help manage the structure of your game classes and their dependencies.
- What challenges did you have making your game components? What techniques did you use to overcome these challenges.
- Rate how reusable the game codebase is? What percentage is specific to the game? What percentage is specific to the platform (monogame/unity)?
- How maintainable is your code? Is it easy to add more enemies, pickups, weapons, levels etc?
- Did you finish your game? Is there any outstanding technical debt?

UML:



Structure: I tried to follow the separation of concern strategy, naming things based only on what they are intended to do. That helped me construct classes and methods that performed a single task at a time allowing for easy debugging and transfer of data.

Player Control for example only takes in input from the player keyboard and sends that data to Player class where now we know when to activate reloading, shooting, or interacting. Other scripts do the same thing such as levelCount and GameManger. Player class being the main class only has dependency to PlayerControl and Shoot_weapon script only because player script keeps track of changing weapons and shooting based on the current weapon.
Another strategy that is implemented is the flyweight strategy. There was no need for it due to only 3 weapons existing but it allows for more efficient code and change behavior of weapons through one instance of a weapon.

Challenges: The biggest challenge i would say would be Player Weapon Scripts and Zombie Behaviour scripts. It took me a few times to delete and restart my code to get multiple weapons or at least data of multiple weapons and be able to use those weapons without having to create new instances or have multiple active gameobjects running in the back. AKA to follow proper strategy patterns.
When it came to zombie behavior the difficult part was AI. as of right now i still believe it needs to be remapped but I couldn't afford to waste that much time rewriting zombie behavior code so as of right now what i had the zombie do to follow player properly and not get stuck on objects following a direction is to give the zombies state. If zombie is not stuck continue chasing towards direction of player but if zombie comes in contact with obstacle not a player for certain amount of time relocate to a nearby designated hub free of obstacles and potentially closer to player.

Reusability: due to all scripts using Start(),Update() or even GetComponent the code base is unfortunately not reusable outside unity. Inside of unity between other games classes such as abstract class Human(better name would Entity) which solely keep track of player dead or alive state while setting variables such as MaxHealth, CurrentHealth, AttackDmg, along with overridable methods. In fact this script was from another personal project.
Another component that is reusable inside unity is the scriptable weapons. The functionality is basic but player can easily assign and change values to call any weapon type.

Maintainability: as i said earlier with scriptable objects those can be used to ad infinite amount of weapons as long as they are of Interface IShootable. levelCount is a simple increment on round and number of zombies able to spawn. In other words adding random number of weapons or zombies doesnt not effect the game besides difficulty and any method attached with monitoring these game objects such as switch weapon will work just as they do expect having to switch infinite amount of weapons. There is no longer any

repeating code so tracking and debugging should be easier with the help of separation of concerns strategy.

Finished?: I am glad to say i am done with the main components of the game. All components i thought of originally or later down the line have been added lacking in art department and with a shitty zombie Behavior script that just barley works. The biggest technical debt i have to live with is that the current scripts might be to difficult to improve if I wanted to go further with separation of concern and strategy patterns. Because of the logic already implemented it seems like a daunting tasks to try and interfaces for more customizable logic.