

## **Patterns**

The two main patterns I used were object pool and strategy. Object pool is absolutely necessary for this project because it helps to organize and keep track of all my game objects. It helps to keep the memory usage consistent because I'm not constantly deleting and creating objects unnecessarily. The second pattern being strategy. This lets me dynamically decide the kind of gun the player has or the type of enemy my enemy pool will spawn. Strategy really helps because it means that instead of making multiple pools for every kind of enemy or pick up, I could just make one pool and change an object of the pool to whatever I need.

## **Challenges**

I had challenges with the firing of weapons at first. Originally, the gun would fire towards the mouse but then when it came to making the wave gun I had troubles making the arc shape. To fix this problem I made the gun targeting based on the rotation of the user. The player already rotates to face the mouse so this way all I had to do was add more rotation to the fire method to create the sweeping arc.

## **Reusability**

I think a few aspects of the code are reusable. The object pool is pretty reusable and a lot of the gun logic is non monogame specific so that would be reusable as well. A large portion of the working code is specific to monogame though which is unfortunate.

## **Maintainability**

I think my code is pretty maintainable. Adding new guns and enemies is pretty simple. You just need to make a new class and inherit from the desired parent class and make a little new logic. My pick up system could be a lot more maintainable I just didn't have enough time to refactor it.

## **Finished?**

I would say that in the bases sense, I did complete my project. I have some technical debt still lying around and my pickup system never got refactored but the game functions and I feel like I could feasibly put in the effort to continue making more content for it. Some of my classes got extremely bloated and should probably have been implemented over multiple children classes that add functionality piece by piece. My spawner is in my enemy manager which is like a half a class nested inside the enemy manager so I should probably have split those up. There were a couple features I couldn't get too. I wanted to add nice visual stuff like screen shake or flashing when the player took damage but I never got around to it.