

# **Pac Man Man**

Retro Game with a Twist

# Concept

## PACMAN MEETS GHOSTBUSTERS

Imagine, being dropped into the wild of pac-mans world, eating away on these delicious pellets of savory sweetness, when suddenly you are confronted with the ghosts of the maze. Time and time again you're hunted, swallowed, sent to the depths below. But not this time! This time you fight... Gather weapons and turn the sweet savory pellets into weapons of ghost destruction!

## THE TWIST

Instead of just walking around, avoiding ghosts and eating pellets, it is time to fight. Weapons are added to the game, each depending on different types of pellets you need to collect. To defeat the ghosts you need to use the right weapon in a rock paper scissor kind of way.

Weapons are obtained by random drops and by destroying statues and destroyable walls, and their ammunition is filled by picking up pellets as is done in the classic PacMan game.

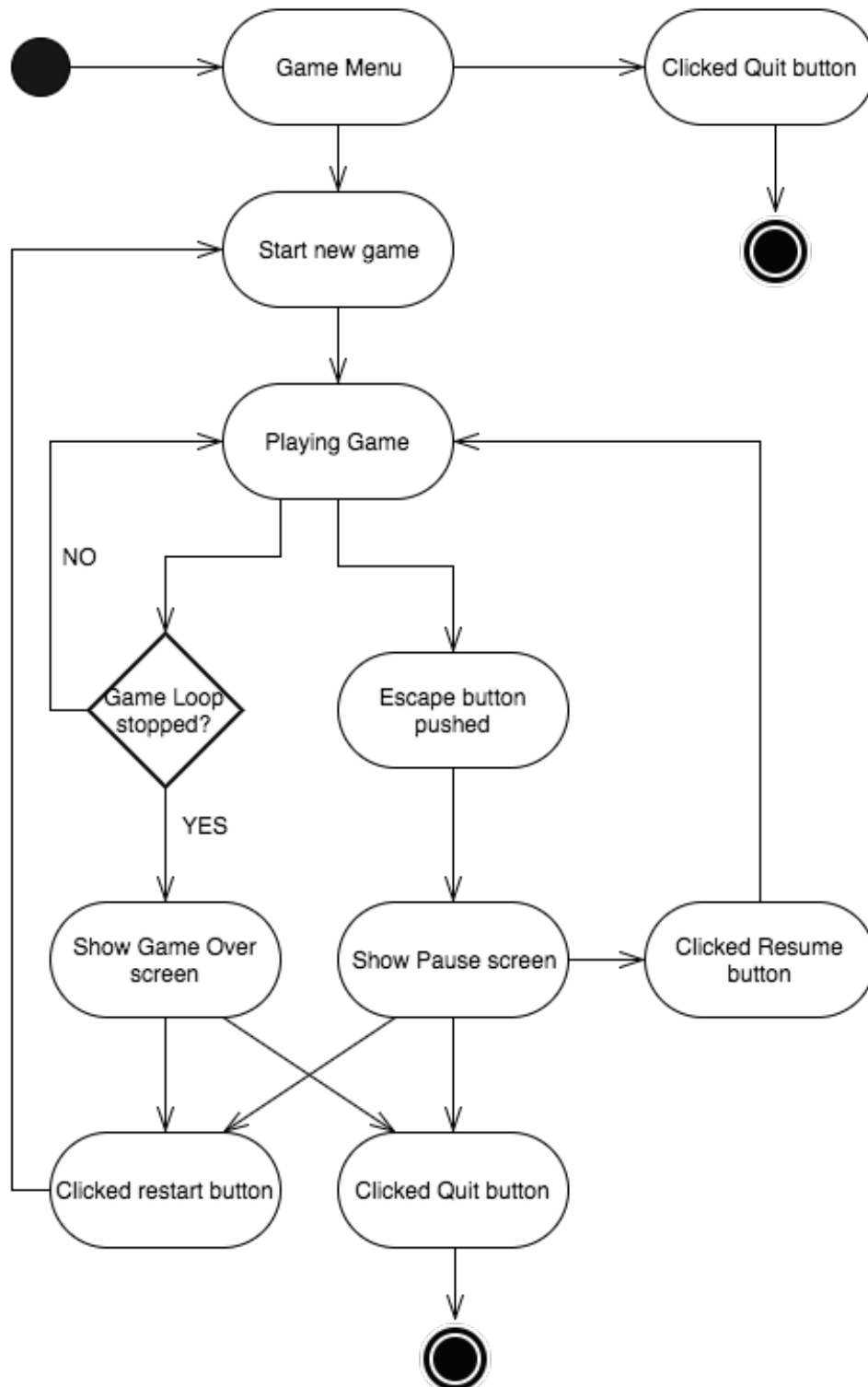
To beat the high score the player needs to kill as many ghosts as possible with as little pellets as possible. The amount of pellets in possession after all the ghosts are killed will be added to the total score.

## DAMAGE TABLE

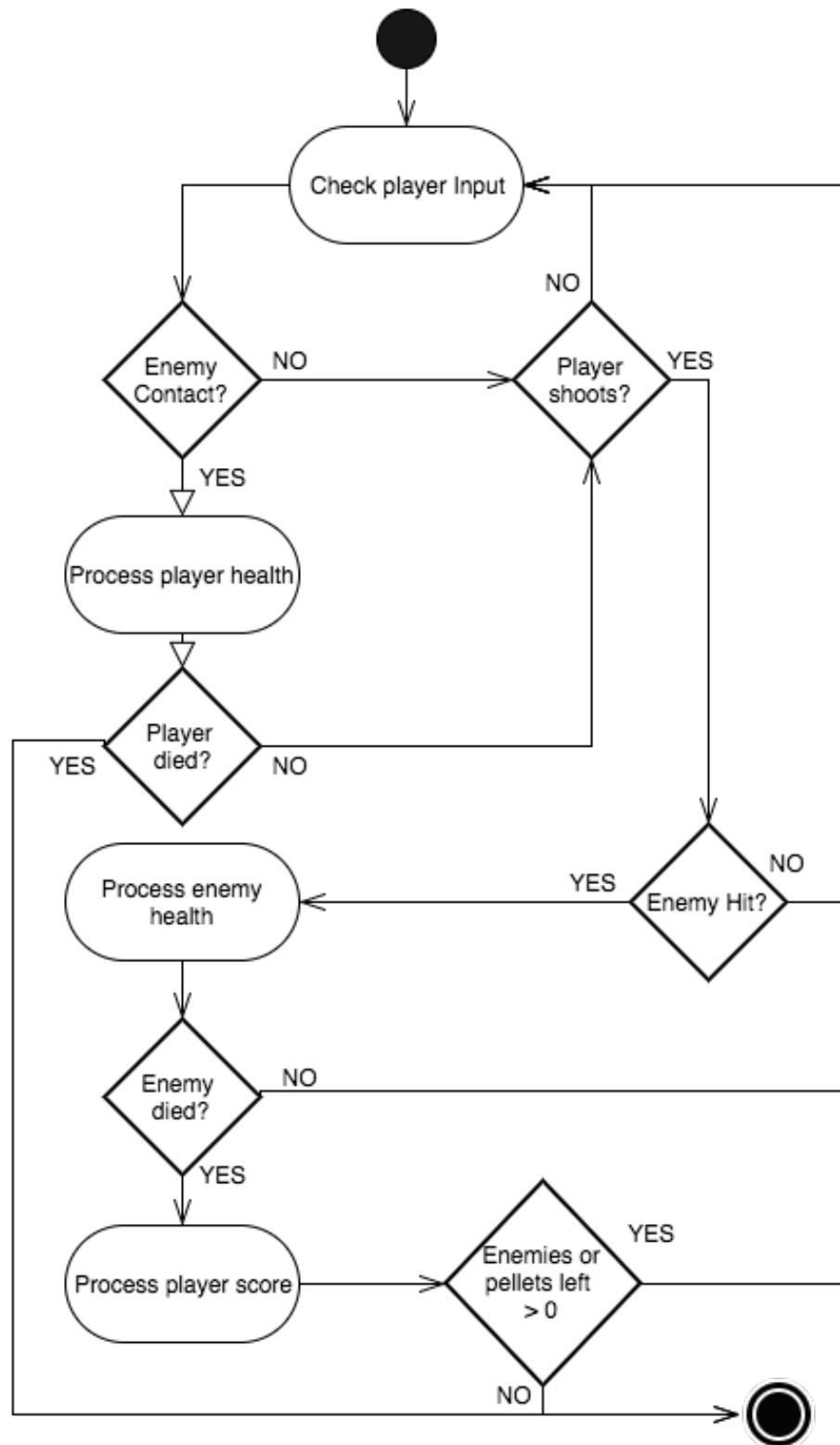
	Loner Ghost	Herd Ghost	Stalker Ghost
The Shocker	High	Low	None
The Blaster	None	High	Low
The Piercer	Low	None	High

# Activity Diagrams

## SCENE FLOW

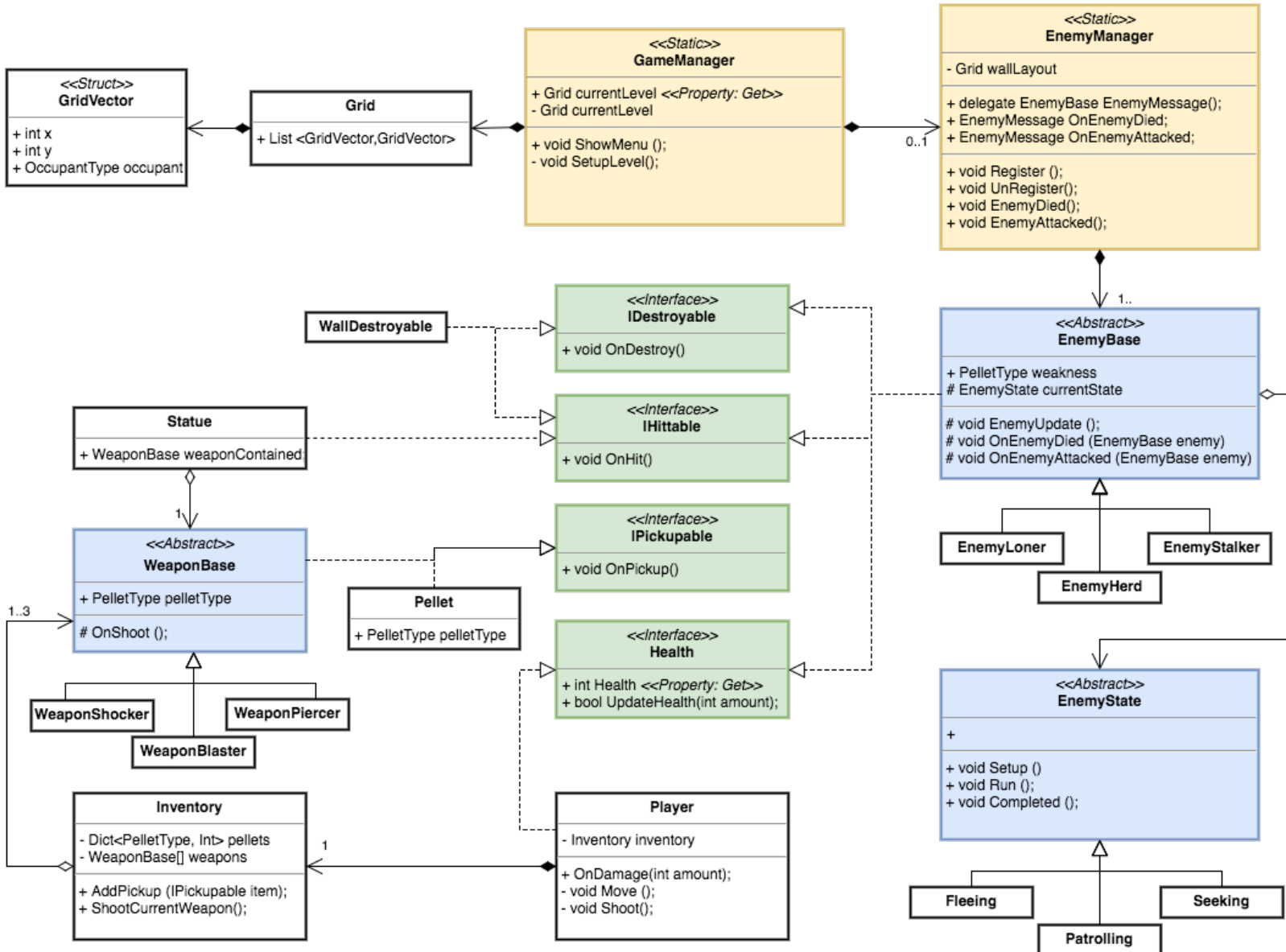


## PLAYING GAME

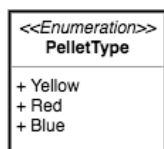


# Class Diagram

## DIAGRAM



## FURTHER EXPLANATION



The diagram seen above mostly speaks for itself, but for the sake of clarity I'll dive a bit deeper into it here. An important element to note is that the Pellet class, as well as the EnemyBase, Weaponbase and Inventory classes, all use the PelletType Enumeration, shown to the left. This is the base for all comparisons made concerning damage and ammunition.

## **Design Patterns and Unity**

I'll be using the following design patterns and standards to create the game:

- Singleton for the Game Manager and Enemy Manager
- Observer for the communication between Enemies and the Enemy Manager
- State for the behavior of Enemies
- Strategy for the various path finding systems of Enemies
- Decorator for the weapons, allowing for upgrades
- Factory for the creation of Enemies
- Objectpool for Enemy instances to limit instantiation