CRITICAL HIT STUDIOS

**Design Specifications**

for

Plants Vs Zombies Game

Presented By:

Aman Garg

Avanti Dorle

Apurva Surve

Mayank Nagpurkar

Purva Ekatpure

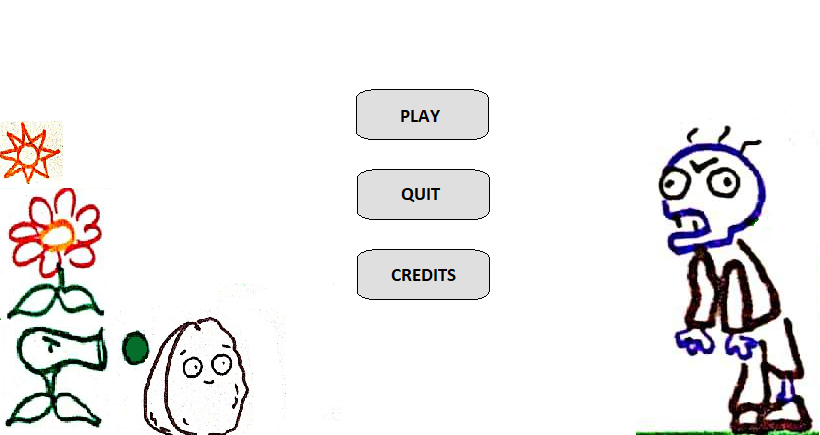
Shivkumar Patil

**Contents**

1. Problem Statement
2. Wireframe
   1. Start Screen
   2. Playing Screen
3. Use Case Diagram
4. Architecture Diagram
5. Class Diagram
6. Communication Diagram
7. Activity Diagram
8. Specifications
9. **Problem Statement**

Creating a clone of Plant vs Zombies game, which is a desktop application. The aim of the game is to protect user’s house from zombies, using magical plants.

1. **Wireframe**
   1. **Start Screen**

****

The start screen of the game will contain three buttons Play, Quit, Credits button as shown above.

* 1. **Playing Screen**

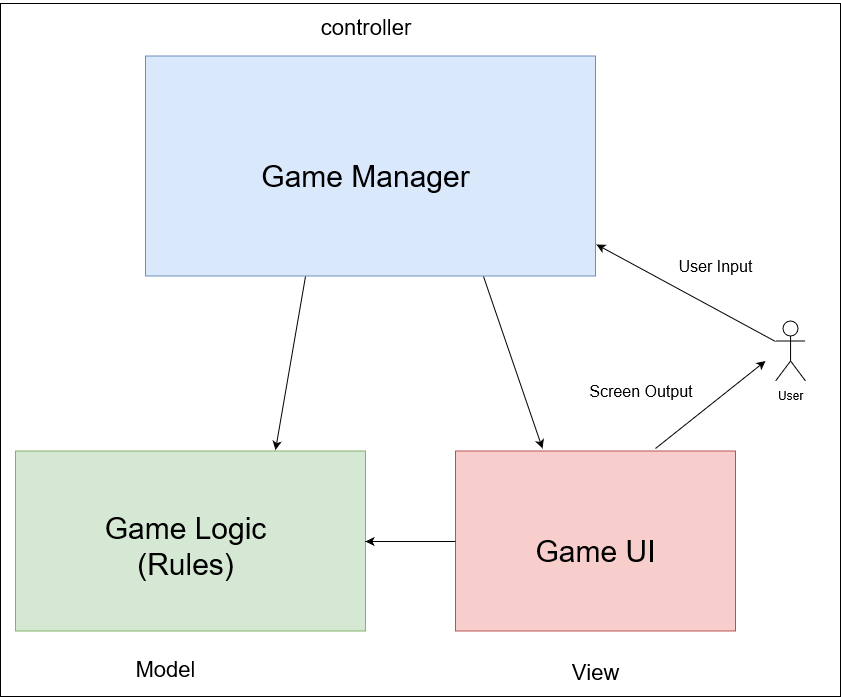
**A close up of a map

Description generated with very high confidence**

On clicking ‘Play’ button on the start screen, playing screen will open, as shown above. It contains a 5 x 9 grid, on which we can place different plants. A panel on which plant seed to select plants and the available sun count is also present as shown.

1. **Use Case DiagramA picture containing text, map

   Description generated with very high confidence**
2. **Architecture Diagram**

****

This game will follow MVC architecture. The three main modules of this game are:

1. Game Logic:

Game Logic which is model contains all the rules of the game world, and it also contains the state data of every game object.

1. Game UI:

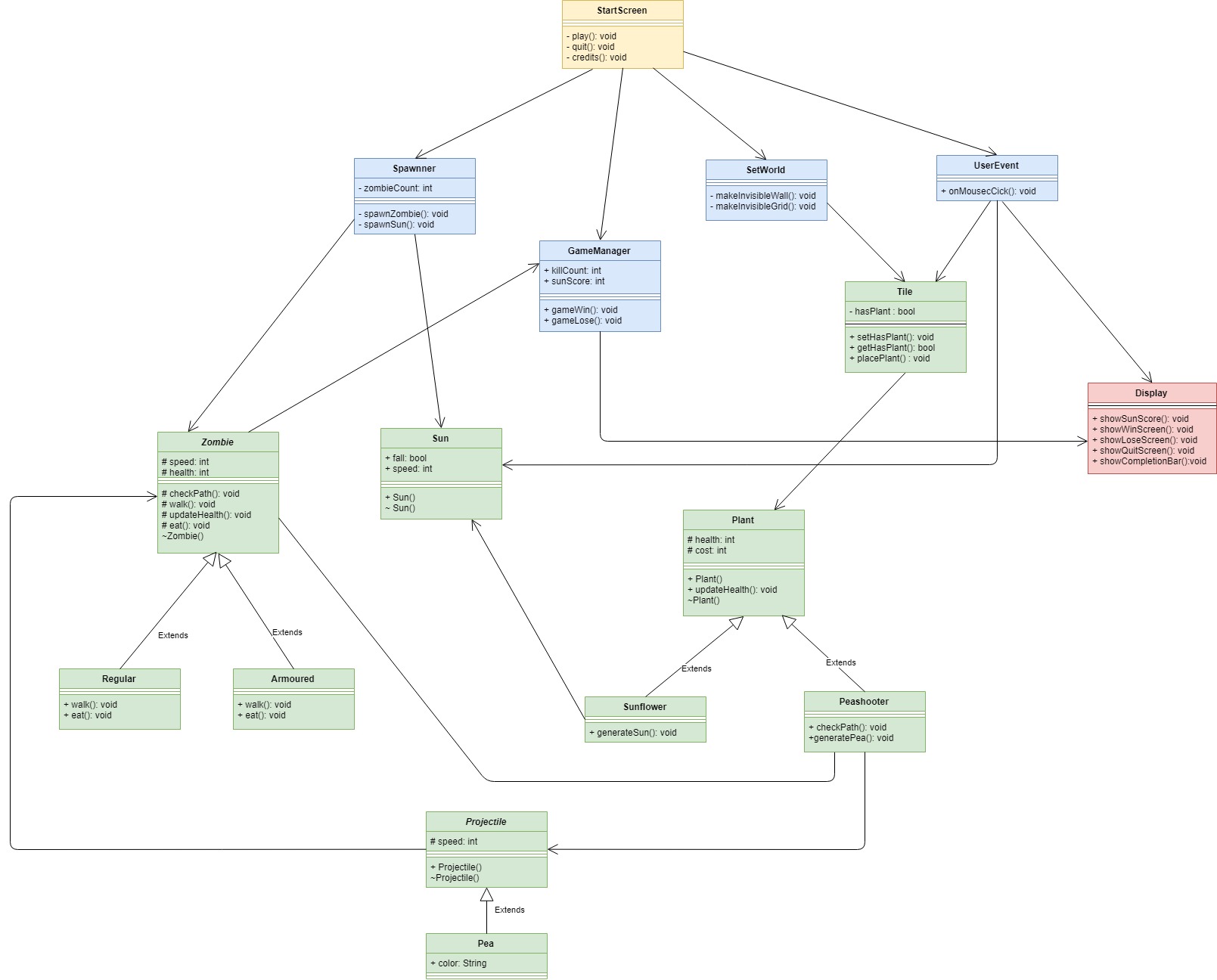
Rendering to the screen is handled by the view. It uses the model to where to draw everything.

1. Game Manager:

The Game Manager which is controller handles the user input, then it might query the view to see which on screen objects are being clicked by the mouse, and finally it manipulates the model.

Each module will contain following classes:

1. Game Logic (Model):
   1. Zombie
   2. Regular
   3. Armoured
   4. Sun
   5. Plant
   6. Peashooter
   7. Sunflower
   8. Tile
   9. Projectile
   10. Pea
2. Game UI (View):
   1. Start Screen
   2. Display
3. Game Manager (Controller):
   1. Spawnner
   2. GameManager
   3. SetWorld
   4. UserEvent
4. **Class Diagram**

****

1. **Communication Diagram**

**A close up of a map

Description generated with high confidence**

1. **Activity Diagram**

**A close up of a map

Description generated with high confidence**

1. **Specifications**

Framework – Unity

Programming Language – C Sharp