Secret Feature Detailed Plan  
(Attack Option B)

**for**

The Matrix

Team: TeamMatrix

Project: The Matrix

Team Members:   
*Afoke Abogidi*

*David Kelly*

*Kulpreet Singh*

# **Table of Contents**

[**Table of Contents**](#_heading=h.gjdgxs) **2**

[**Introduction**](#_heading=h.3znysh7) **3**

[Existing Infrastructure](#_heading=h.vlpb7zl7fzab) 3

[Secret Feature Plan](#_heading=h.lfg2n7adzjyy) 3

[Classes are involved](#_heading=h.au0sbvnnncm) 4

[Design patterns](#_heading=h.4yq5o4ff6dv6) 4

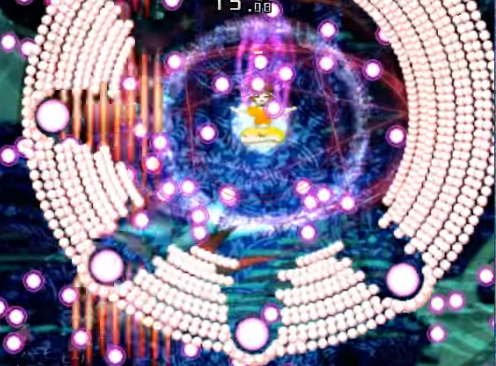
# Introduction

This secret feature option B is Team Matrix’s implementation choice. This option features collision detection between enemy projectiles and their corresponding responses. This document outlines the implementation plan of this feature and provides details about the design impact on the existing infrastructure of the system, what changes are anticipated and how these changes would be made.

## Existing Infrastructure

Our current collision detection subsystem detects and handless collisions between the enemy entities and the player and their corresponding projectiles. This is achieved by implementing a hit box zone for each entity and detecting if and when an opposing projectile hits this zone. On detecting a collision, an action is triggered using the Observer pattern to induce a change in the health of the entity collided with. The constant check for a collision happens in the game framework’s Update method and collisions between projectiles of the same parent are ignored.

## Secret Feature Plan



The new feature requires the projectiles of the boss to be spawned from a spawner whose initial position is close to the boss’s position with a direction of movement away from the boss and off the screen at a slow speed. This movement pattern would exist in the movement class and visually take the shape of an orbit expanding outward as the projectiles move outwards. A large number of projectiles would be created and spawned at once to form each of the orbital shapes moving outwards.

The feature also calls for a bigger sized projectile spawned a few seconds after the smaller projectiles. These larger projectiles would adopt the same movement strategy from the movement class but move at a much faster speed.

The highlight of the feature would include the collision detection between these two aforementioned projectile shapes. On detection, the smaller projectiles with an indirect collision (side hit) would slightly change direction, creating room for the bigger projectiles, while the smaller projectiles directly collided into, would adopt the speed of the colliding projectile.

Also, on reaching the boundaries of the screen, the smaller projectiles change color, movement and time out after a few seconds giving rise to the look of splattered projectiles.

## Classes Involved

The architecture of the system would remain intact. No new sprites or classes are required but several classes need modification including;

**Bomb Class:** An additional property for size would be added to the Projectile subclasses. Also, the onCollide Method would be implemented to handle the after collision logic described above.

**ProjectileFactory:** A new projectile factory method is needed to call the current create Projectile method with the additional parameter of a size to resize the created Projectile. This would allow for creating a regular sized as well as a bigger sized projectile when needed and would be an extrinsic state property of the projectiles.

**SpriteManager:** This class will call for the creation of a large number of projectiles in the loop and also call for the creation of the larger sized projectile. A new method will need to be written to create the new projectiles. The method would use PI, Cos and Sin math functions to help in creating the circle of projectiles. The calculated coordinates will be used to set Vector2 structures to be used to set the Position and Direction of the projectiles.

## Design patterns

**FlyWeight Design pattern** would be used to create the large number of Projectiles for the attack pattern and also for the larger Projectiles.

**Strategy pattern:** Although already in use, would be modified to include the new movement pattern to form the orbital shape while moving outward.