TEST PLAN – ALL IN

| In your /doc folder, upload a test plan that outlines how to user-test your game, title the document: **test-plan.docx**. For example, what keys to press or what mouse actions are necessary. This short document will be used by the TAs to test and mark your game. |
| --- |

# **Basic overview of the game**

- WASD controls (W for up, A for left, S for down, D for right)

- Auto shooting

- The player (called protagonist) always faces the mouse regardless of movement direction

- Collect coins from killing enemies

- Use the coins to purchase upgrades from the main menu shop upon death

- Press space to dash, only if player is not taking damage/recently took damage

# **Basic User controls and visuals**

- Load the game

- Shop screen: Players can navigate to this screen to spend their gold on permanent upgrades which are persistent across player deaths and game loads.

- Use WASD to move: (W for up, A for left, S for down, D for right)

- Move the mouse around the protagonist and see the protagonist face the mouse in 4 directions (up, down, left, right)

- Press space to do a short dash

- Notice that the protagonist automatically shoots four types of projectiles at the mouse (however not all types are unlocked on the first wave)

- Roulette ball (yellowish small ball)

- Bounces on enemies and walls

- Card

- Pierces multiple enemies

- Dart (most infrequent)

- Has higher damage

- Diamond Star (mesh-based collision projectile)

- Also pierces but with high damage

- The protagonist’s projectiles do damage and eventually kill enemies.

# **Basic environment**

- Enemies, which are cards, spawn in the ‘room’. There are currently three types of enemies: King of Clubs, Queen of Hearts, Boids.

- King Clubs:

- slowly move toward the player and damage the player on collision

- Queen of Hearts:

- Locates low health King Clubs based on an AI decision tree

- Follows and heals nearby and low health King Clubs by shooting a heart projectile that tracks the target King Club

- “Bird” boids:

- Boids are club birds that work together in a swarm.

- When alone or in few numbers, they will seek out each other to form a larger group

- Once grouped together, they will begin to fly towards the player quickly and drag the player along their flight path while preventing the player from dashing

- Boss bird:

- It’s a boss club, which is an enlarged boid. However it is not a boid. It jumps towards the player like a frog would.

- Every jump, it spawns a bird boid out its behind, like a duck would.

- Enemies only spawn in the room in waves.

- Once all enemies are defeated, a door spawns in the top left to progress to the next room (wave)

- The protagonist is enclosed in a room (surrounded on all four sides by walls)

- Travel in a direction till you meet a wall.

- The protagonist is not able to pass through the wall

- No enemies spawn outside the wall.

# **Thorough test for each criterion**

Playability & Stability

- Buttery smooth. You see it.

No crashes, glitches, or unpredictable behaviour.

- All crashes are planned, and we have none planned.

# **Creative elements: 5 basic features**

* Use flow fields for pathfinding
  + During each frame, a distance map is computed from the player’s position. All other entities will look up their current position on the distance map and add up the sum of f(x)=1/x of its surrounding distances to create the vector pointing towards its next direction. We opted for this approach over A\* due to the large amount of entities that we have on later rounds. This approach also allows for smoother/natural movement.
* Add audio feedback for at least three interactions in the game as well as background music with tones reflecting the journey of the game.
  + There are 5 phases of the game, each with distinct background music, enemy variety, and opening door sounds. The 5 phases are waves 0-2, 3-6, 7-11, 12-19, 20 onwards.
  + Add sound effects for the following interactions (implemented in M3)
    - Player taking damage “knife stab” sound effect
    - Player collecting coin “ding” sound effect
    - Player opening door mysterious sound effect
  + Other sound effects not implemented in M3:
    - Roulette ball hitting enemy sound effect
* Simple creative use of the fragment shader. For example changing the colour of a sprite over time. The colour should change based on a uniform input (e.g., change the uniform based on time, user input or, when a collision is detected).
  + Player will glow red upon collision with an enemy and taking damage
  + Utilizes the fragment shader (textured.fs.glsl) to mix in a red glow
  + Utilizes uniforms
* Procedural generation of rooms
  + After the first room (which is the same every game), when the player moves to the next wave via the door, a new room is randomly generated. Currently only one room style has been implemented (donut). The outer ring and inner ring dimensions are randomly selected within a set range of possible values, and the walls/floors are set accordingly.
* Shop
  + Created a shop that lets players spend their collected gold on permanent upgrades that help them advance through the waves of the game
  + Upgrades scale as the game continues and they become more expensive as you buy more