UAT Plan

for

Tilt Rush

Index

[1.](#_heading=h.gjdgxs) Scope 3

[1.1.](#_heading=h.30j0zll) Objectives and business requirements 3

[1.2.](#_heading=h.1fob9te) Scope 3

[2.](#_heading=h.3znysh7) Testing team 4

[3.](#_heading=h.2et92p0) Environmental requirements 5

[3.1.](#_heading=h.tyjcwt) Hardware requirements 5

[3.2.](#_heading=h.3dy6vkm) Software requirements 5

[4.](#_heading=h.1t3h5sf) Test Scripts 1

# Scope

## Objectives and business requirements

In this section, outline the business requirements. In other words:

* What are our goals? What are we hoping to accomplish with this project/feature?
* How will we measure success?

This sprint is intended to add the 3 different types of obstacles to the game. This includes a diamond to kill the player, a mine to move them quickly, and a wall to act as a blocker. These will be randomly generated into the level. As the player gets a higher score, the obstacles will become more numerous, and the diamond can eventually move around on a track.

## Scope

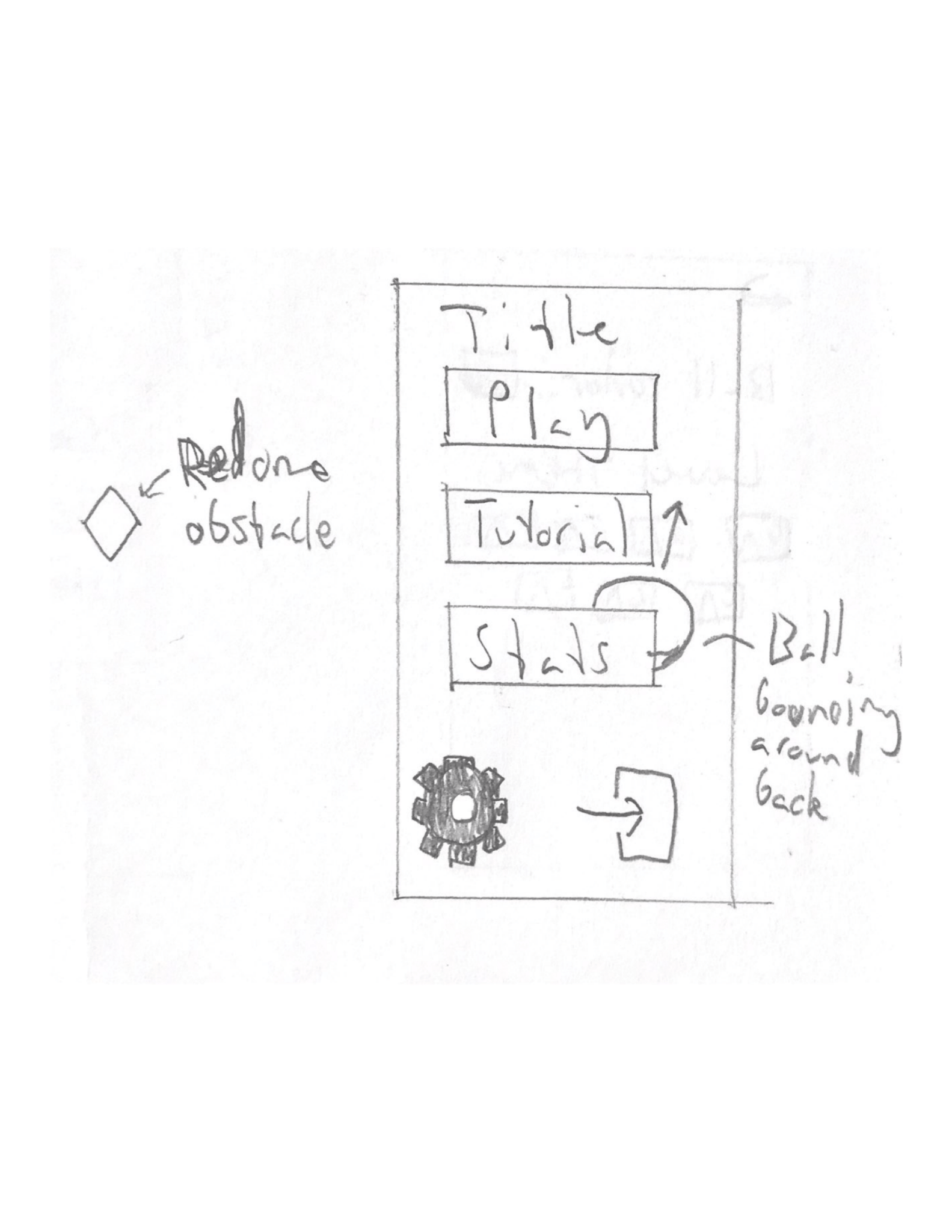
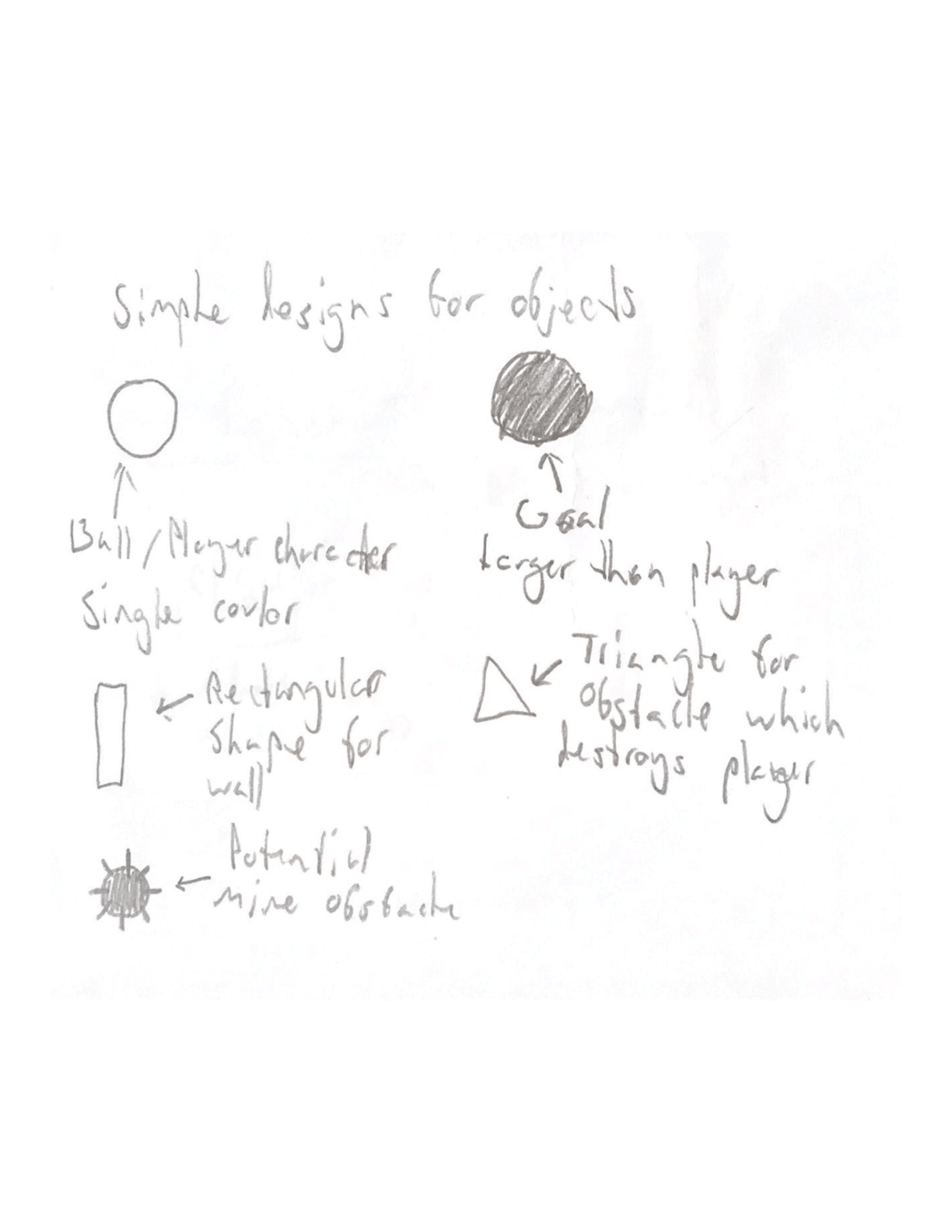
In this section, outline the scope. This means:

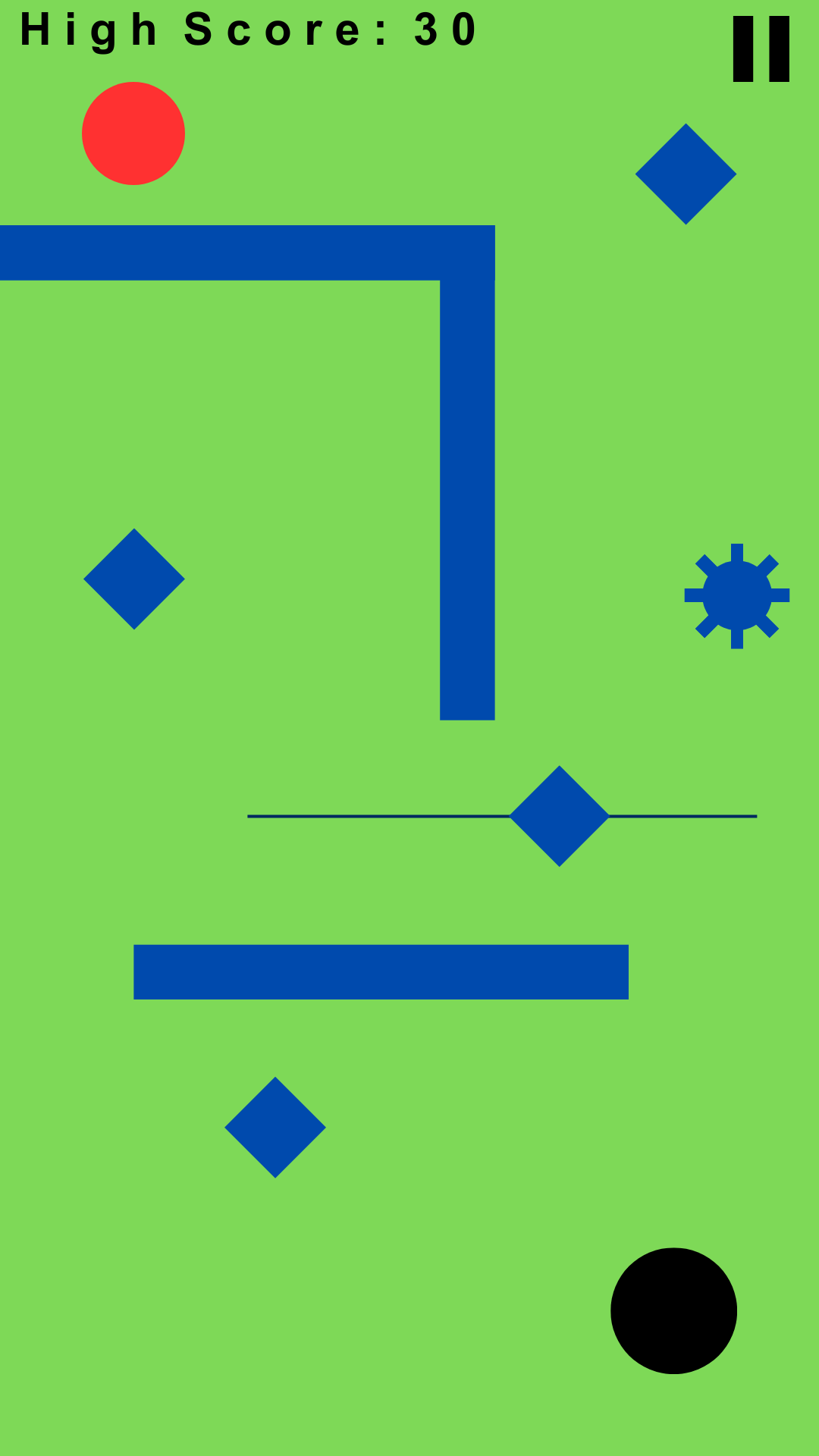
* What is the pain point we’re trying to fix?
* What are we testing exactly, and what are we *not* testing?

This sprint is to add gameplay to the game, and is the last one for everything which is necessary for the game to properly function. This sprint is purely focused on the canvas object inside of the gameplay, and will add a number of new features.

## System Diagrams

In this section, paste any drawings or diagrams that help the UAT team understand the program being tested. With each drawing include a brief explanation of how the drawing represents the application or system being tested.





# Testing team

In this section, list out members of your QA team and what their roles will be during UAT.

Example:

| **Name** | **Responsibilities** |
| --- | --- |
| Lachlan Ng | To test if the various obstacles work and generate as intended, and to give feedback on the current level generation. |
| Matheus Pincus | To test if the various obstacles work and generate as intended, and to give feedback on the current level generation. |

# Environmental requirements

## Hardware requirements

What hardware has the solution been designed for and should be tested on.

If that is the case, outline the minimal and recommended requirements so the QA team can verify that the software runs on the testers’ machines.

Phone with online capabilities.

## Software requirements

If any extra software or dependencies must be downloaded and installed, list them here.

* Google Chrome (or similar browser)

## Network requirements

Some software (design, video editing…) can be demanding on hardware specifications.

If that is the case, outline the minimal and recommended requirements so the QA team can verify that the software runs on the testers’ machines.

* Working internet connection

# Test Scripts

This section is more important than it seems—it is crucial that both the QA team and the testers know what features must be tested, especially if you’re testing a lot at once.

| **Test** | **Describe the feature being tested** | **Describe the user input or test data** | **Describe the pass criteria** |  |
| --- | --- | --- | --- | --- |
| 6.1 | Diamond obstacle | 1. If on iOS go to settings and press “iOS Support” 2. Get to a score of 1 3. Continue increasing your score up to above 30 | 1. The diamond should not generate until a score of 1, at which point it should always generate 2. The diamond should get smaller with the player, stopping past 30 | Tester name: Lachlan Ng   |  | PASS | | --- | --- | |  | FAIL |   Observations:  Tester name: Matheus Pincus   |  | PASS | | --- | --- | |  | FAIL |   Observations: |
| 6.2 | Wall obstacle | 1. Get to a score of 4 2. Continue increasing your score up to above 30 | 1. The wall should not generate until a score of 4, at which point it should always generate 2. The wall should get smaller with the player, stopping past 30 | Tester name: Lachlan Ng   |  | PASS | | --- | --- | |  | FAIL |   Observations:  Tester name: Matheus Pincus   |  | PASS | | --- | --- | |  | FAIL |   Observations: |
| 6.3 | Mine obstacle | 1. Get to a score of 10 2. Continue increasing your score up to above 30 | 1. The mine should not generate until a score of 10, at which point it should mostly always generate 2. The mine should get smaller with the player, stopping past 30 | Tester name: Lachlan Ng   |  | PASS | | --- | --- | |  | FAIL |   Observations:  Tester name: Matheus Pincus   |  | PASS | | --- | --- | |  | FAIL |   Observations: |
| 6.4 | Moving diamond obstacle | 1. Get to a score of 20 2. Continue increasing your score up to above 30 | 1. The diamond should not generate until a score of 20, at which point it should mostly always generate 2. The diamond should get smaller with the player, stopping past 30 | Tester name: Lachlan Ng   |  | PASS | | --- | --- | |  | FAIL |   Observations:  Tester name: Matheus Pincus   |  | PASS | | --- | --- | |  | FAIL |   Observations: |
| 6.5 | Level generation | 1. Get to a score of 1 2. Continue increasing your score until you feel you’ve seen the extent of the generation 3. Repeat the process multiple times | 1. The generation never gives a purely impossible level 2. If it does please take a screenshot and place it here 3. Under observations, write your thoughts on the current state of the level generation | Tester name: Lachlan Ng   |  | PASS | | --- | --- | |  | FAIL |   Observations:  Tester name: Matheus Pincus   |  | PASS | | --- | --- | |  | FAIL |   Observations: |

Write step-by-step, detailed but concise instructions on how to test the feature.