Outliers Design Document

# Overview

Outliers is a simple mobile platformer designed to provide quick, engaging, puzzles based in part on “spot the difference” games from newspapers and other media in the past. With a focus on simplicity, the game will also take inspiration from mobile “social games” like candy crush, using high scores, speed, and difficulty as metrics to allow players to compete and promote replayability. Using simple visuals, clear UI, and score as metrics the game is intended to be a fun, simple, quick experience.

# Platform

Mobile is the platform that best augments the simple gameplay loop, replayability, and quick session format this game follows. As such we plan to deploy to mobile and use a simple button system for most of the controls in the game.

# Genre

The game falls under the genre of Puzzle Game

# Core Gameplay

## Mechanics

The core gameplay mechanics for this game are simple. The player will use 2 metrics and their finger to achieve the highest score. To do this the player has to tap the correct answer as fast as possible. Each question will have a timer. If a player answers a question incorrectly, or fails to answer correctly within the time limit, the player loses a life and the question is either shuffled and drawn again, or it will progress to the next question depending on what feels better for the player. When a player gets a question correct it will also proceed to the next question. Each correct question will give Points = roundup(TimeRemaining \* 1000 / (MaxQuestionDifficulty – Current Question Difficulty + 1). As such if there are 5 difficulty levels of questions and a player answers a level 5 question, they will receive points = time remaining \* 1000. While currently there are plans for only 5 difficulty levels, it is possible we will find ways to implement more going forward.

## Player Goals

The players goal is simple, to achieve the highest score as fast as possible. This makes this game ideal for a competitive or social game and will allow players to compare and compete high scores.

## Components

The game will have several distinct components: Timer, Health, Puzzle and Score.

The timer is designed as a way to impose the fast paced and speed-based nature of the game, forcing the player to think and act fast.

The health is designed as a metric to help end a playthrough. Rather than ending a playthrough after a single mistake, having leeway can help improve the player experience.

The puzzles are the challenges placed in front of the player. Each set will have a theme and will select a set of (4) random options 1 of which will not belong in the set. For more difficult sets we may also have a small set description describing what the set is.

The score is the metric by which the player is rewarded. This is what the player wishes to maximize.

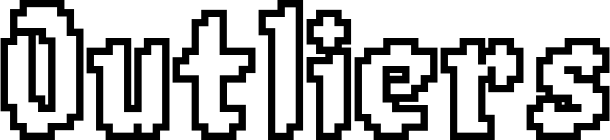
## Controls

The controls are simple, the player will tap the buttons on the device. That being said, I want to provide feedback to the players touches and inputs on the screen. Besides visual feedback for answering. Vibration/Sound can be used to convey an incorrect answer. On the other hand, correct answers may be more subtle and only provide the visual feedback, and perhaps a smaller quieter sound.

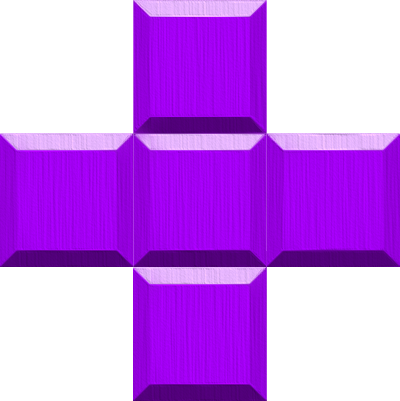
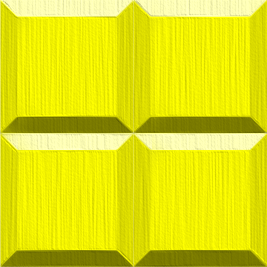
## User Experience

The game should explain itself simply without any text, besides being part of the puzzle. Furthermore, the simple design and easy to view metrics should encourage the player to push themselves to a higher and higher score leading to a simple yet enjoyable experience.

# Visual Style

While most of the application (HUD, UI, Text, Ect)will be in the same art style, 2D pixel art, some puzzles will use different art styles such as vector and 3D art in order to convey different challenges, or as the solution to the puzzles themselves. 

The logo and text will most likely have a pixel structure like the text above.

The puzzles might use slightly different artstyles to allow for greater variation and perhap seven

# Music

?

# Project Goals and Beyond (Game Direction)

Where we want it to be if we can (not in scope)

# Timeline

Self explanatory

## First Checkpoint

This is closer than I expected

## Second Checkpoint

This has become a shitpost

## Final Checkpoint

It really is tho

## Showcase

AHHH

## The Future