Alex Li (alexl3)
Game Design Document: Can't Touch This

Project Description
This is a platform based projectile dodging game that tests player's abilities to dodge different types and speeds of projectiles. The platforms may be moving, but the level height and width is static. Increasing levels will have increasing difficulty.

Version History
No versions completed yet.

Characters
Equilateral triangle - the player
Square - archnemesis, will send various projectiles including other polygons

Story
You are an equilateral triangle, and many enemies (which are just different projectiles that may or may not be other polygons) want to "break" you because of your strength. You desperately avoid them desperately to survive and prove your strength.

Gameplay
WSAD to move, P to pause.
There will be a menu screen and (if time permits) a level selector.
There will be a mini cutscene to give some background.
Level consists of platforms to jump on and projectiles appearing in random locations along the screen's edges and flying towards the player with random trajectories.
Falling off the platforms and into the bottom of the screen results in death and losing screen.
Being hit by a single projectile results in death and losing screen.
Levels progress based on time, and higher levels will mean less or moving platforms and unique and more projectiles.
There is a final level where the square appears and swings itself towards the triangle many times and other projectiles may appear.

Artstyle
Polygons and simple geometry-dash artstyle will be used. Particle effects will definitely be used to supplement the simple art style.

Music and Sounds
Builtin or royalty free unity sounds and music will be used, 8 bit sounds and music preferred. Player jump (boing sound) and land (landing sound) will have sounds, and projectiles shooting (pew) and crashing (mini explosion) will have sounds. Player being hit will have sounds (explosive sounding). Menu screen buttons will have sounds.

Technical Description

The game will have very low technical requirements, and mainly the Unity engine will be used. Scripting will be at a minimum, and many prefabs will be used to replicate projectiles. Different projectile behavior may require some scripting. The game will be 2D.