Ping Blast GDD (Game Design Document)

20220925 v1.1

# Game Basics & Technical Specs

## Name:

Ping Blast

## Genre:

Aggressive Pong-like

## Synopsis:

The player will control a small cube that fires smaller cubes as bullets. There will be an enemy on the other side of the game field that will mirror the player’s movements with a delay but fire of its own accord. Powerups will spawn randomly on both sides. The player will attempt to destroy the opponent using a clever combination of moving, firing, and dodging while utilizing powerups to do so.

## Players:

This is a 1-player game.

## Form:

2-D assets and gameplay, minimalist design.

## View:

The gameplay will take place in front of the camera.

## Platform:

PC, Web

## Language:

Primarily C# with some C++

## Device:

Any web capable device with a keyboard

# Gameplay

## Outline:

When the player begins the game, they will be presented with both squares loading into their respective sides of the field. The field itself is split into two equal halves – a black lower half, and a white upper half. The colors of the squares controlled by the Player and the Enemy are inverted to the fields they play on as are the bullets they fire. ~~When a bullet crosses the threshold between the two fields, its color inverts, helping to maintain visibility of the bullet.~~ The Enemy moves with an approximately 2 – 5 second delay that mirrors the player’s movement but fires on its own. The Player will need to move and take shots in such a way that they can avoid being hit but while also striking the enemy. They and the enemy will randomly be provided powerups that must be moved over to be picked up. The gameplay is endless with the player wracking up as high a score as they can before they are destroyed.

## Key Features:

List of entities:

1. ~~Player square~~
2. ~~Enemy square~~
3. ~~Bullet squares~~
4. ~~Black field~~
5. ~~White field~~
6. ~~Game field (i.e. the entirety of both the white and the black field)~~
7. Rapid-fire powerup
8. Spread shot powerup
9. Rapid-fire spread shot powerup
10. Score counter

# Design

## Guidelines:

Color scheme must not be more than 2 separate contrasting colors (i.e. black/white, orange/blue, green/purple) and all entities can only be squares. No rotation of game objects in any fashion. No music. Sound effects should be simple and sound like retro in some fashion. Think Mr. Game & Watch. Score should hold up to a maximum of six digits. Present digits as analog clock style. Controls should be snappy and responsive. Do not allow the player to simulate Rapid-fire powerup by spamming the fire key.

## Definitions:

**Win**: You are winning so long as you are playing.

**Lose**: You lose if the enemy square fires and hits you. Your score is reset to 0 along with the field. New squares load in after 3 seconds.

**Level transition**: There are no separate levels. After the player has destroyed the enemy ship a new one will load in.

## Flowchart:

This space is reserved until a proper flowchart can be made. Include Objects, Properties, and Actions with an index number reference indicating where they exist in the game mechanics document.

# Player

## Definitions:

**Form**: The player will control a white square.

**Health**: There will be no traditional health bar or counter. One shot will destroy the player.

**Actions**:

* **Fire**: This will cause the Player to shoot a small white cube in a linear direction depending on where the Player clicks in the field.
* **Move**: Player movement will be limited to the field they are currently in and will not involve rotations as noted in the *Gameplay Outline*.
* **Pickup**: Picking up a powerup will destroy the game object and apply the specified effect to the Player for the specified amount of time.

## Properties:

This space is reserved until the Player properties can be documented appropriately.

## Rewards:

**Rapid-fire Pickup**: This will apply a multiplier to the fire rate of the Player, allowing them to fire multiple rounds in a linear fashion.

**Spread Shot** **Pickup**: This will apply a multiplier to the number of shots a player can make, ~~with the number of shots spread evenly over the 180° arc in front of the Player.~~

**Rapid-fire Spread Shot Pickup**: This is a rare pickup that will apply both the Rapid-fire and Spread Shot effects to the Player.

# User Interface & Controls

## Design:

The UI should reflect the minimalist design of the rest of the game and mirror the two-tone color scheme chose. Hard corners should be used on buttons to reflect the use of squares throughout the game. Do not apply any 3-D, gradient, shaded, or other such effects. Maintain a flat look that emulates the card style of recent versions of Android OS. UI elements should feel snappy and responsive.

## Controls:

**Menus**: Mouse

**Gameplay**:

* **Movement**: WASD keys in traditional format
* **Firing**: Click left mouse button
* **Pausing**: ESC key

## Actions:

**Menus**:

* **Start**: Clicking this button will allow the player to begin the game
* **Who**: Clicking this button will display a UI window with information about the developer and ways to contact them

**Gameplay**:

* **WASD**: Pressing these keys will allow the Player to move their square around their designated field. Traditional format indicates the following:
  + **W** will move the Player square up
  + **A** will move the Player square to the left
  + **S** will move the Player square down
  + **D** will move the Player square to the right
* **Left Mouse Button**: Pressing the left mouse button will trigger the fire action in the direction of the mouse pointer, with any pickup effects applied appropriately
* **ESC**: Pressing this key will freeze all activity (movement, timers, etc.) within the game and display a UI window allowing the Player to return to the title screen or reset the game.
  + **Main Menu** will reset all elements of the game and return the Player to the main menu.
  + **Restart** will reset the current field by clearing effects, pickups, the score and returning the Player and the Enemy squares to their starting positions.
  + **Return** will return the Player to the game with a short countdown before play resumes.