Game Design Document

Team: Mazeland

# **Team**

| Name | Email | Spurpunk TD review |
| --- | --- | --- |
| Shibo Zhang(Team Captain;Enemy Control) | shibozha@usc.edu | https://youtu.be/Kke2brSLXVE |
| Mingdong Lyu(Product manager; Analytics) | mingdonl@usc.edu | https://youtu.be/HX8jvAXe5ts |
| Zifeng Lin (Version Control) | zifengli@usc.edu | https://www.youtube.com/watch?v=iwWn0\_tOTyk |
| Xiujing Huang (User Interface) | xiujingh@usc.edu | https://www.youtube.com/watch?v=Hu2N8fc77CU |
| Tianding Zhang(programmer) | tianding@usc.edu | https://youtu.be/bNQvSRG7MDE |
| Yintang Yang (Programmer) | yintangy@usc.edu | https://youtu.be/exPrTCbu-ME |
| Yang Zhang（Game Feel, programmer ） | yangz673@usc.edu | https://youtu.be/X2eXk2\_PhGY |
| Chuanshi Zhu(programmer) | zhuchuan@usc.edu | https://youtu.be/\_71f3Zo2sL0 |
| Ruize Zhang(Product manager,Analytics) | ruizezha@usc.edu | https://www.youtube.com/watch?v=LEytUpuILrM |
| Minzhi Zhan(User Interface) | minzhizh@usc.edu | https://youtu.be/dsA9ySEOY-s |
| Jerzy Ramos Chen(Programmer) | ramosche@usc.edu | https://youtu.be/0amXbrHk1d8 |

# **Github and WebGL Link**

1. <https://github.com/Samberg-0808/Mazeland>
2. <https://mazeland.itch.io/divisible>

# **Game Overview**

Original Prototypes:

[Prototypes](https://docs.google.com/document/d/14tmkzz_W47w2EiTJmOj0cRHUVd3-8Sr9LyW5t0xK2cA/edit#)

## Genre

Survival, Collecting, Exploration, Math

## Goal

Survive until the last second and collect score to reach goal

## Mechanics

Use WASD to move in 2D plane

Player has 3 lives in total and each damage loses one life.

Player can eliminate enemies if the number of the enemy is the factor of the number on the player

Get scores by collecting green circles or by killing enemies.

Players need to avoid the black circle and cannot collide with the black circle.

The black circle will shrink as time goes by.

Players can collect green circles with different scores to increase the player’s total score.

Collect enough Scores to reach the next level.

## Math

Controlled random generation for point orbs, enemies, and assigned numbers

## 

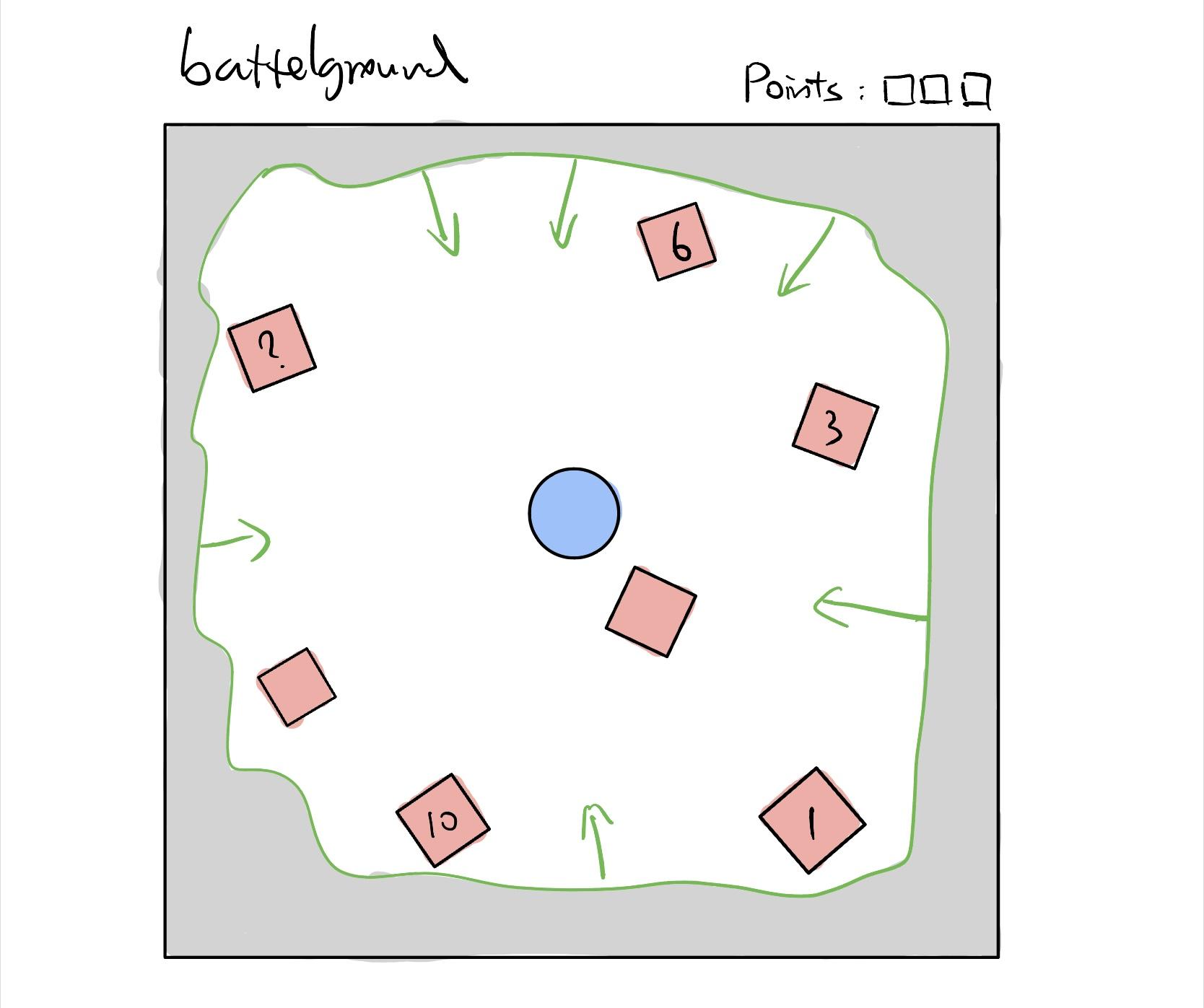
## Game Description

Last-man-standing gameplay with the survival, exploration and scavenging elements of a survival game

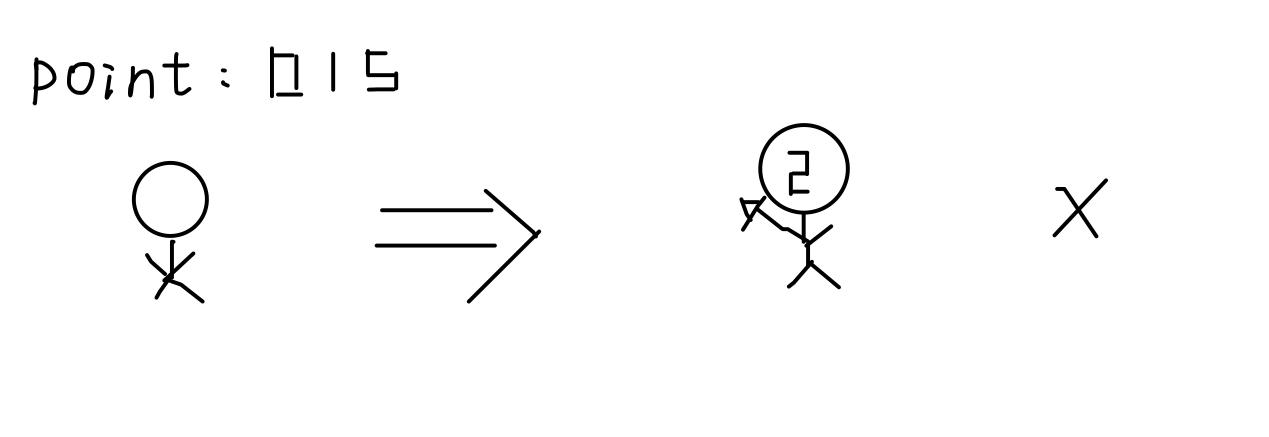
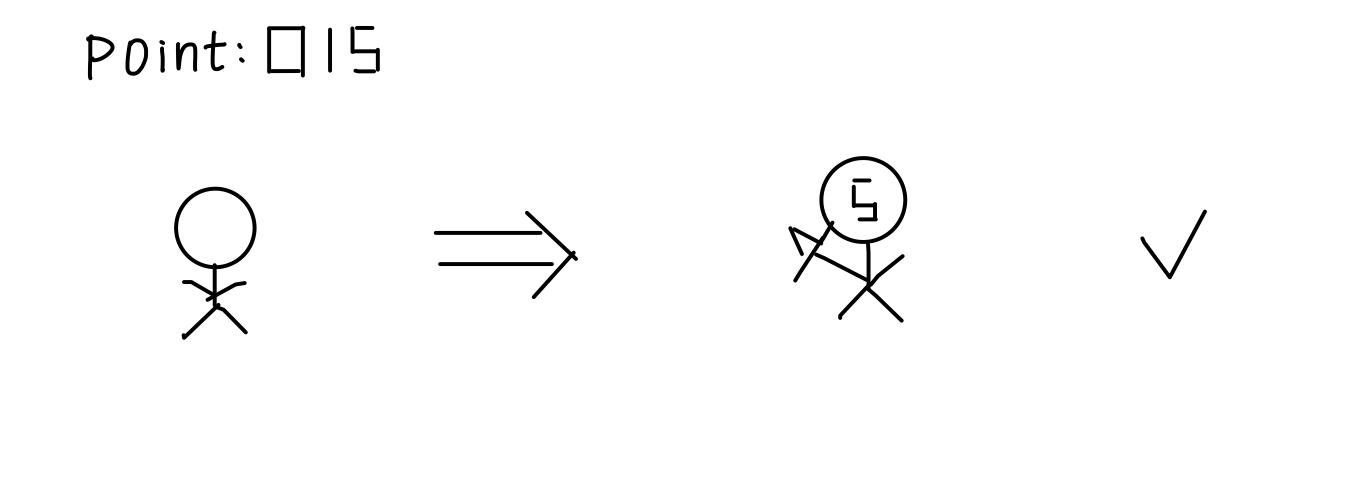
### Detailed Idea:

Move the player orb to collect generated point cubes to increase total score. The player is able to eliminate certain cubes by collision. In the meanwhile, the battleground will gradually shrink to a random size. The play will have to stay in the battleground in order to be alive and reach certain points to level up.

### Drawing:



The player can move by using WASD. Collect points by colliding with cubes. Try to collect as many points as possible to level up. Always stay in the safe zone to avoid loss of points.



For example, if the player has point 15, the player can kill enemies with number 5 on it. However, the player cannot kill enemies with number 2 on it since 15 is not divisible by 2. If the player with points 15 hits enemies with number 2, the player will get penalties like decreasing two of the total points and generate an enemy with a number which is the reminder of it.

# 

# 

# 

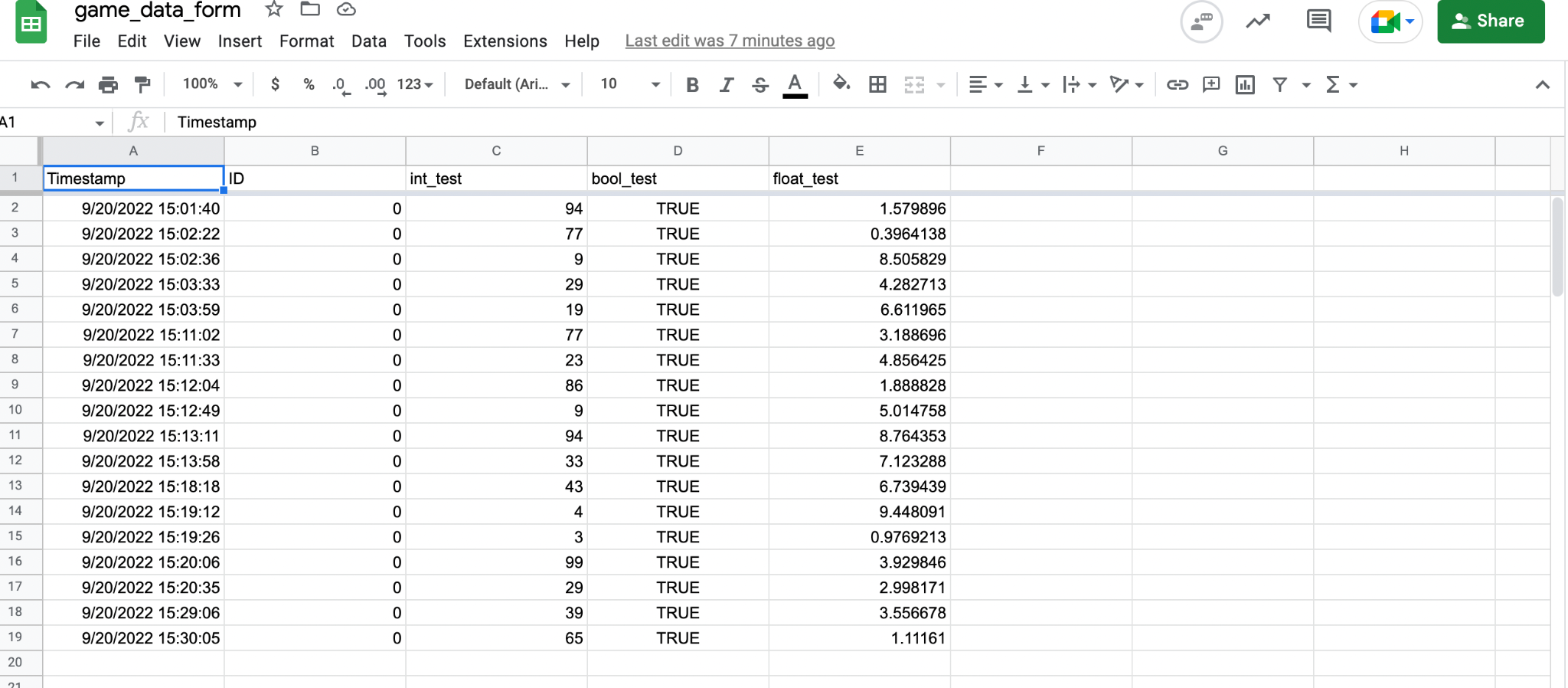
# 

# 

# Analysis

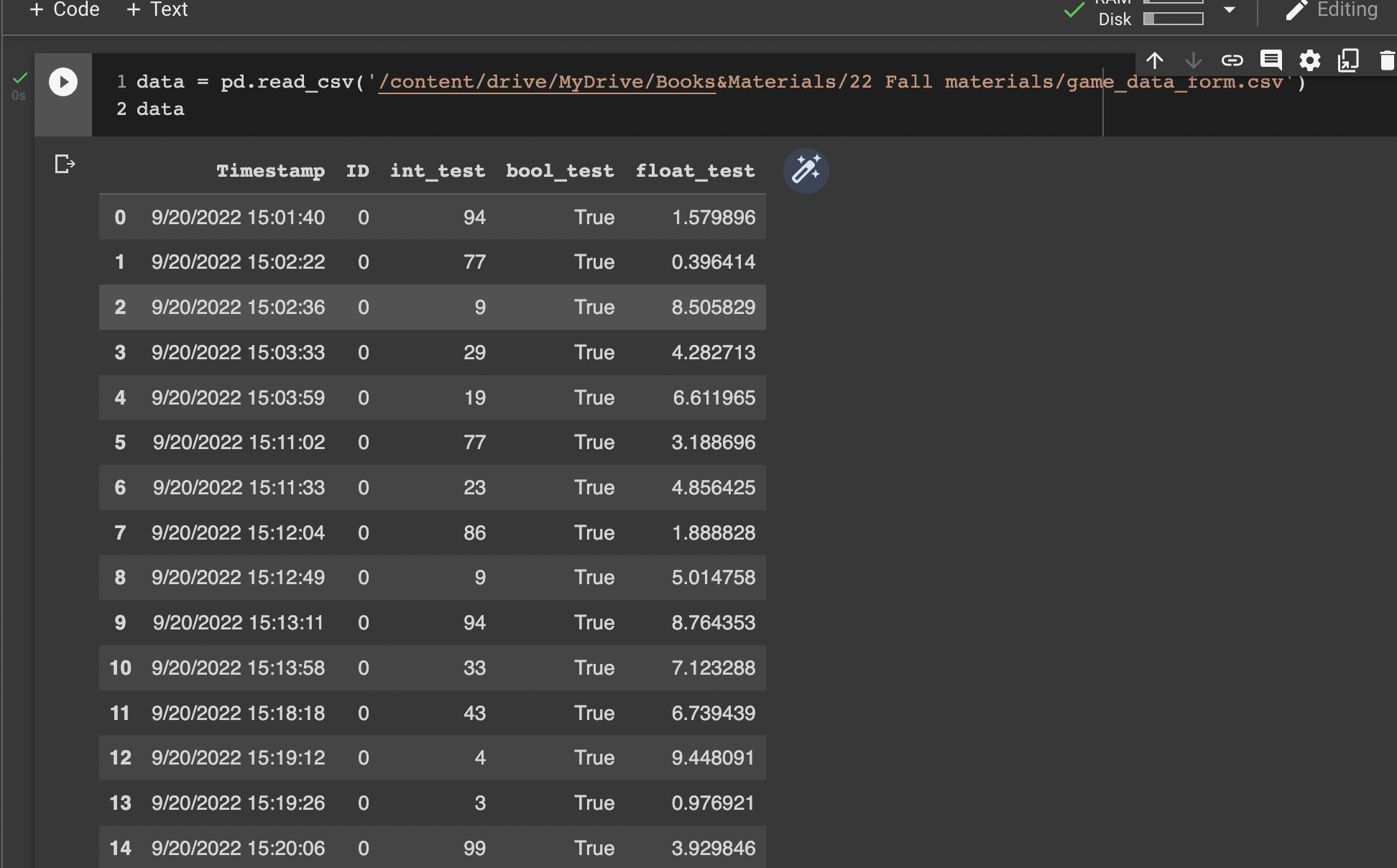
Data collection:

* We send our metric to a Google Form through Unity. We can then get a spreadsheet with all the metrics tracked.

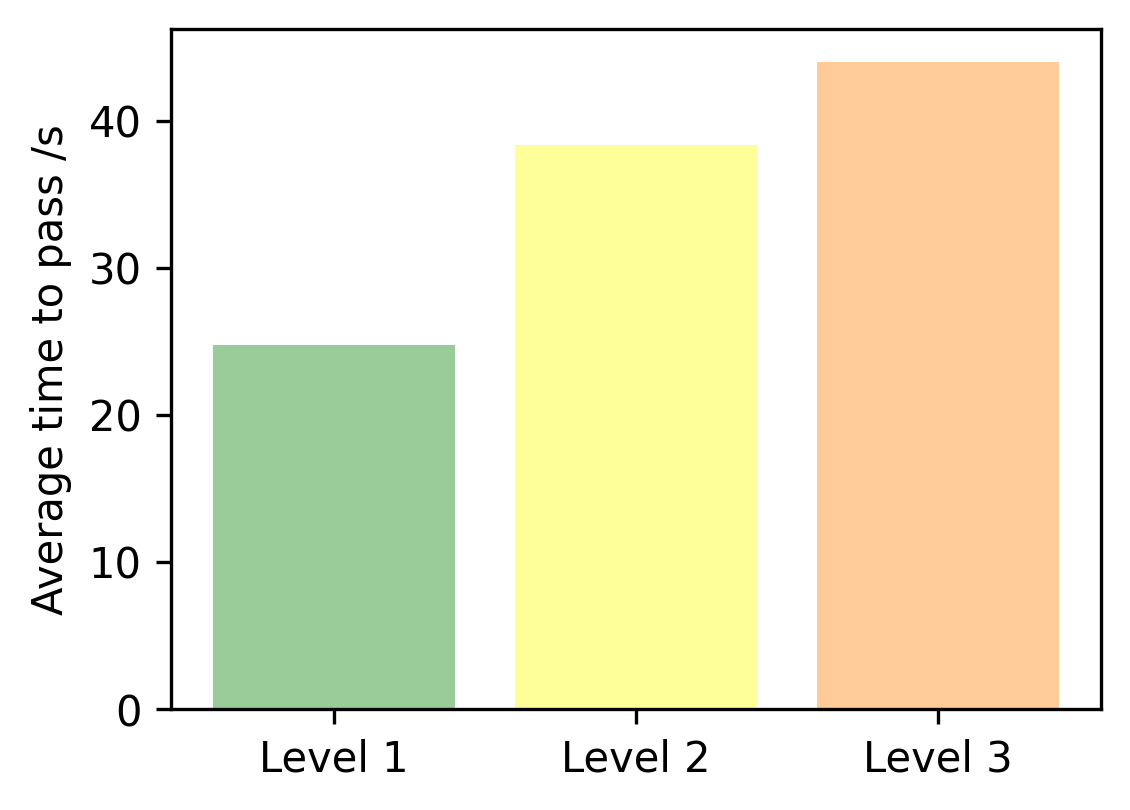


Data analysis:

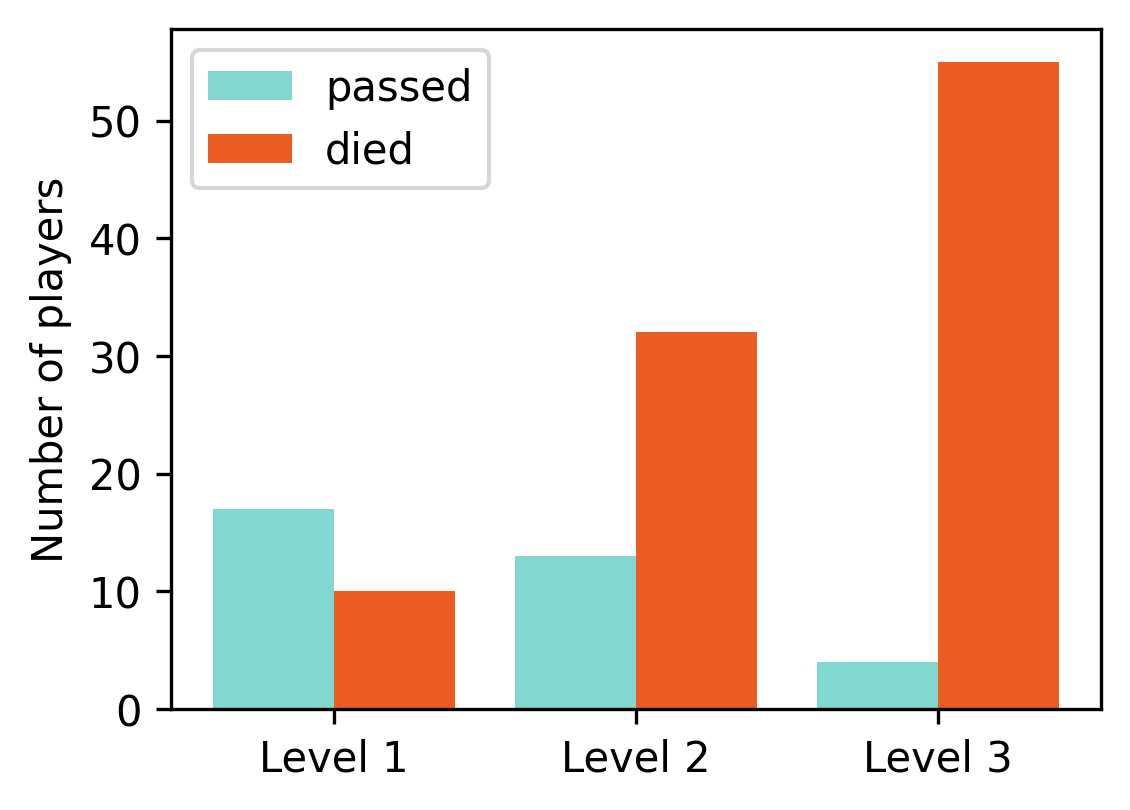
We will save the data from the Google Form as the .csv file. The collected variables are starting time for each level, end time for each level, and the dead level for each player. We will then calculate the average survival time and number of players that pass/fail each level with pandas. The results of the data analysis will be plotted out by matplotlib.



## Data Plot 1 & 2:

***How long does the player need to get to the next level?***

***How many players can reach each level?***

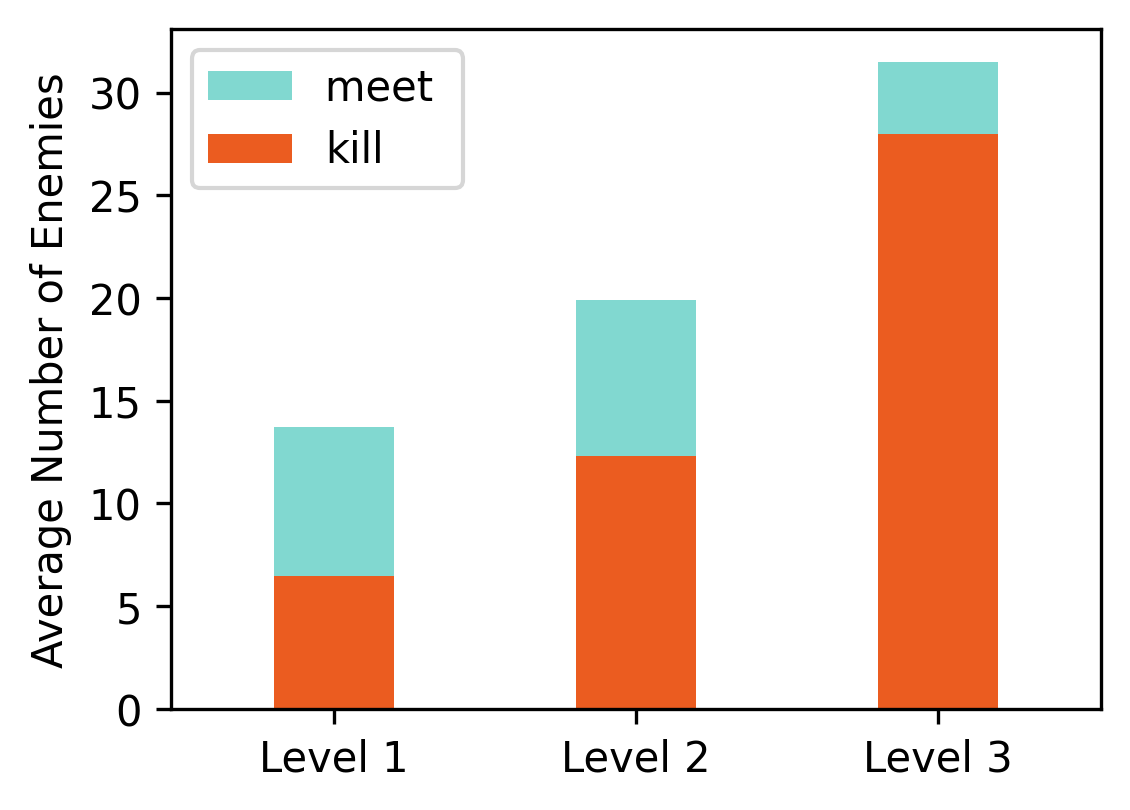


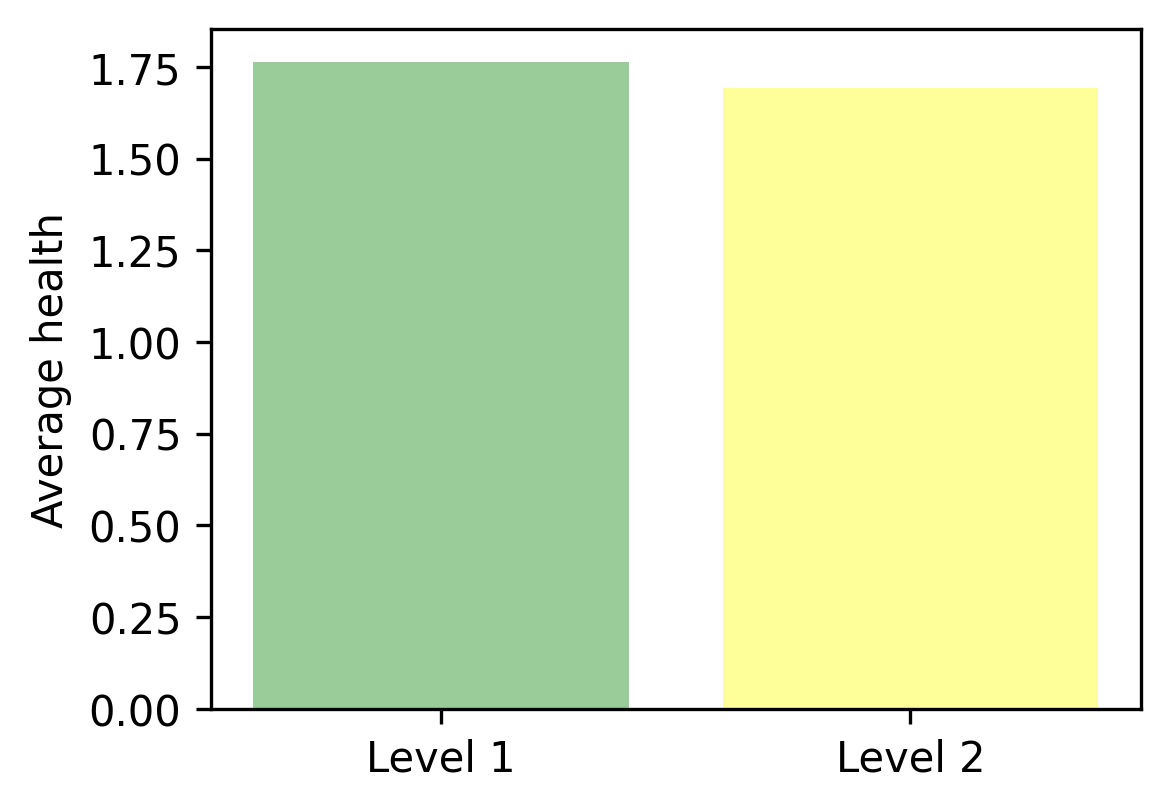
The reason to collect those two data is to measure the difficulties of each level. We can adjust the difficulty level based on the data.

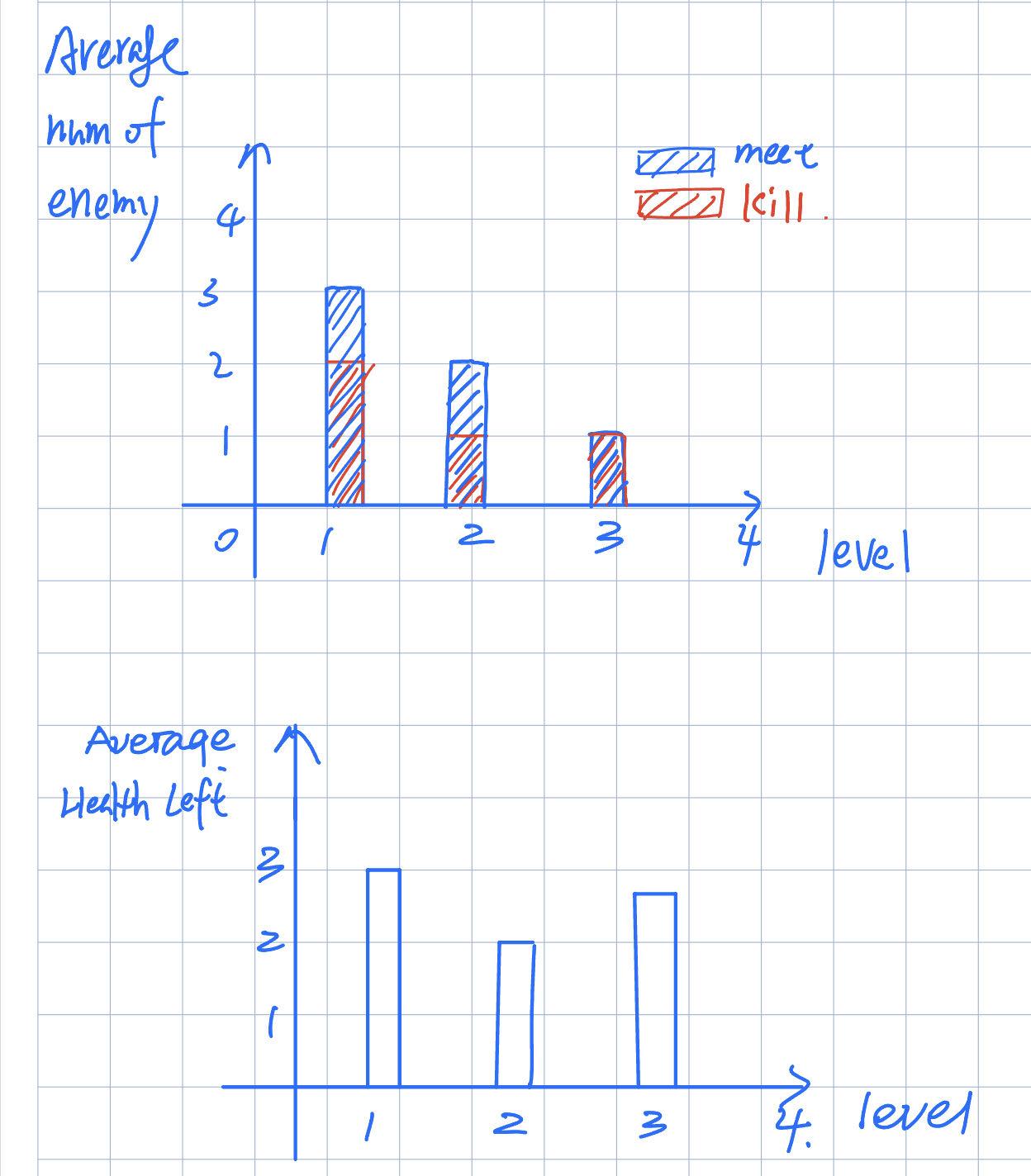
## Data 3 & 4:

How many enemies a player kills vs total number of enemies they meet in each level.

What is the average health of player in each level?







# Game Design Weekly Feedback

## Weekly Goal:

### Goal for 09/22

1. Player:
   1. Effect design
   2. Player health
2. Small circle
   1. Change size with score
3. Enemy:
   1. State showing
   2. Collision effect
   3. React with player
   4. Design enemy number
4. Start menu(Xiujing Huang)
5. BGM and sound effect (Shibo Zhang)
6. Level design (tutorial)
7. Data analytics

Level1: tutorial (Fixed enemies, points, and circle)

Level2: move enemies and circle

Level3: add props

### Goal for 09/27

1. Complete level 3
2. Add props/items(Chuanshi Zhu)
3. Enemy moves toward the player (Yang Zhang)
4. Add point to Player(Minzhi Zhan)
5. Limited green points and enemies(Tianding Zhang)
6. Adjust music effects (Shibo Zhang)
7. Add data analytics(Ruize Zhang; Mingdong Lyu)
8. Add level information (Yintang Yang)
9. Modify tutorial information and add some rules on level 1(Xiujing Huang)
10. Improve green circle / Add number on green circle (Mingdong Lyu)
11. Change level background (Zifeng Lin)
12. Change enemy score(Minzhi Zhan)
13. Modify GDD(Mingdong Lyu)
14. Fix Bug (Yintang Yang - Fixed game not end bug)
15. Adjust shrinking circle
16. Seperate the collision cs file

## Feedback And Improvement

**9/6 Prototype Feedback**

* Short descriptor for mechanics to explore, new things.
* New element, sound/dance/colors. Combine it with somethings that has nothing to do the [genre.us](http://genre.us)

**Update:**

* Designed new game prototype
* Delete the all well-known elements
* Change the game mechanics and genres
* Create the game sprites by hand drawing

**9/13 2nd Prototype Feedback**

For other teams:

* Goal needs to be clear, the player needs to know what to do. (Exit sign/door instead of blue square)
* Break up teams and individually think of little ideas/mechanics to add to the game. Vote for best ideas.
* Communication to the player is important through guidance instead of text instructions.

For our team:

* Hard time understanding role, missing validation of doing the right thing.
* No obvious reward/danger.

Update:

* Add progress bar to remind player’s status of each level
* Add more information on screen to help players understand the game concept.
* Add sound effects to demonstrate the reward/danger
* Change the enemy color based on the division
* Use different colors to separate enemies with different scores.
* Add game menu and User interface

**9/20 Feedback**

* Assumes boxes are enemies
* Not sure how to divide enemies
* Not sure what ‘divide’ means, thought cut in half
* Understands not to touch the outer ring
* Still doesn’t understand how the division works
* Does not understand what progress bar refers to
* Likes the concept, thought the limit was short but didn’t realize

**Update**

* Change the enemies’s sizes to separate different kinds of enemies rather than use different colors.
* Add one tutorial level to help new players understand our game
* Update the enemies colors when they can be eliminated
* Change all friendly objects including points and enemies to be killed to green.
* Add health system to illustrate the penalty more stragefowardly.
* Add interaction for players collecting points as well as getting damages.

**9/26 Feedback**

Presentation tips

* Speak from the audience point of view
* Elevator pitch: give a clear description that explains the game thoroughly
* Sell the point of the game
  + Don’t assume knowledge from user
  + Say things like: It’s a math game visualized, the mechanics include…

**Improve GDD**

1. Lack of mechanic detail
2. Update Google sheet
3. Github main branch update
4. Take notes and put it on GDD
5. Add change based on feedback

**Update**

* Complete Level3
* Add props to have some special effects
* Delete background music
* Adjust the number of enemies
* Adjust the shrink circle speed
* Limit the total number of enemies
* Visualize the total score on player