# *Rogue-like chess*

## 1. Project Overview

This is a 2D rogue-like game built on pygame. There will be 3 waves, 2 of which the player has to fight enemies, black chess pieces, and collect an item to move on to the lastwave, the boss wave. In the first wave only players use ranged attacks while enemies use melee. Then in the second wave both the player and enemies will use guns and in the last wave it will be player vs boss only. After each wave, players will collect money and buy items with strategy to help proceed: wherever to be upgrading themselves (turning a rook into a queen).

## 2. Project Review

This project is inspired by the soul knight game. The key change is that this game will only have 3 waves, various attacks according to the chess rule types of enemies and shops in the game. Another notable difference is that all characters are chess pieces and might have some of the abilities they have in chess, for instance the boss, black queen will move around and shoot a bullet in 8 directions every 5 seconds.

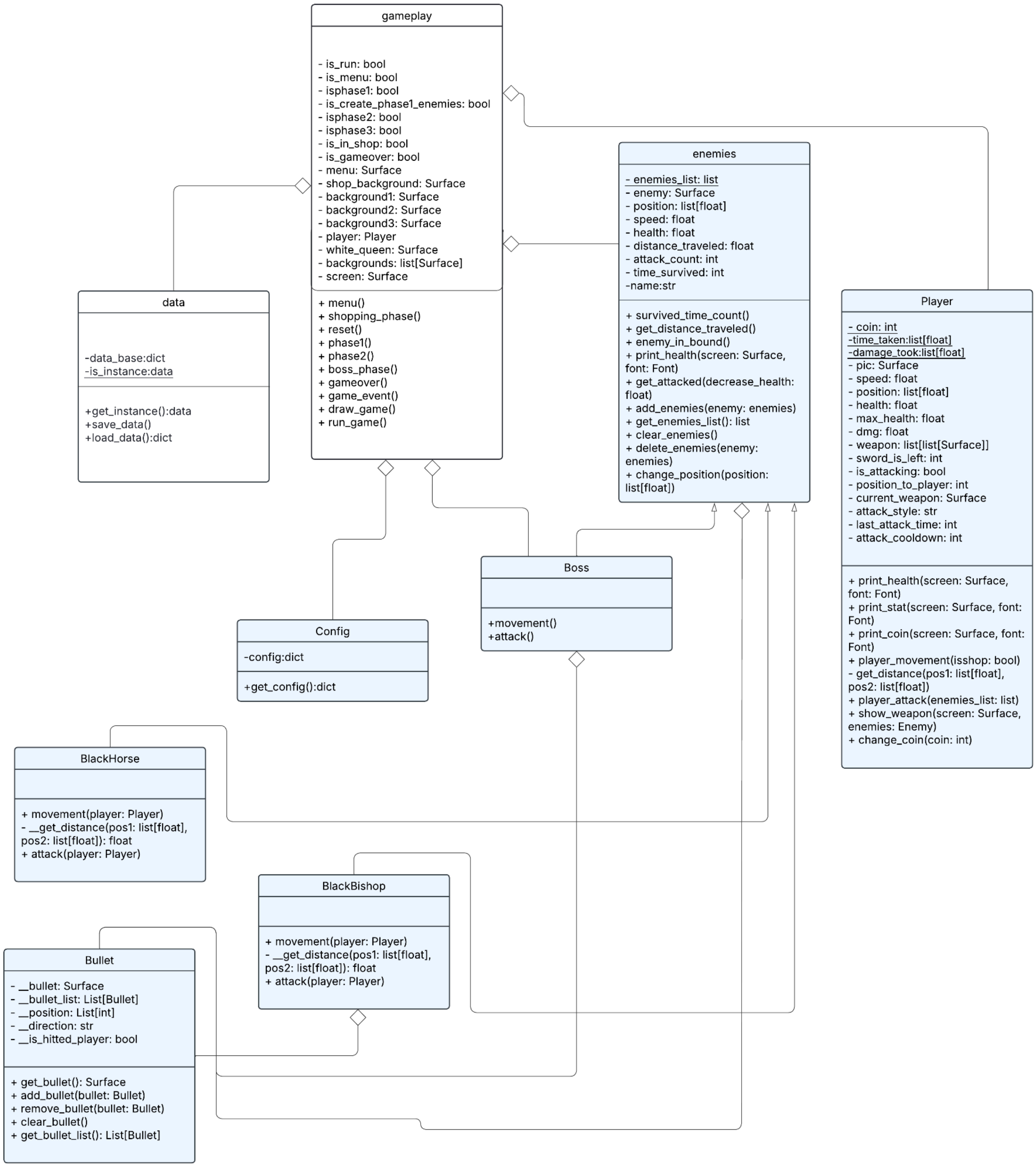
## 3. Programming Development

### 3.1 Game Concept

The concept of the game is chess 2D role play, meaning that now chess pieces can actually fight each other freely without turn rules in chess; pieces can fight without waiting for another one to move. The key highlight is that chess pieces will retain some of their abilities in regular chess.

### 3.2 Object-Oriented Programming Implementation

There will be 6 classes for now: config, enemies, boss, player, gameplay and data. The config class only stores the background size and all the pictures for it as for these 3 waves the background will be changed to make it looks like the player is teleported to another battle field. The enemies class is used for black pieces in the first 2 waves to make them move and fight the player. The boss class will be exclusive for the boss only and also it will inherit some of the enemies classes attributes. The player class is used for the player's movement and actions in the game. The gameplay will be for the game loop menu and end game. Lastly the data class will be used to store and interact with all data.



### 3.3 Algorithms Involved

Algorithms involving path finding for enemies and boss, because shooting bullets of a boss and player will be partially based on chess rules. This means that a bishop can only move diagonally and a rook can only move in 4 directions. Random weight algorithms will be involved in the boss phase as the boss will randomly attack harder than the normal with 1/10 chance.

## 4. Statistical Data (Prop Stats)

### 4.1 Data Features

| Feature | Why it is good? What can it be used for? | How will you obtain 50 values of this feature data? | Which variable (and class) will you collect this from? | How will you display this feature data? |
|---|---|---|---|---|
| Enemy Survival Time | This data will be used to help see if the enemy in which wave is too difficult or too easy to beat. | The data will be collected within 5 gameplay across 3 waves. The time will be counted using the time library and stop for each one when it is removed from the enemies list. | This data will be collected from enemies class and a variable called time_survived. | Using histogram |
| Enemy Attack Frequency | This will be used to record how many times each enemy shows if the enemy is too aggressive or not. | The data will be collected within 5 gameplay across 3 waves. The variable will be incremented each time the attacking function is called. | Enemies class will be used to record this data in a variable called attack_count. | Using bar graph |
| Time taken between wave | Show player's speed and effectiveness of playing style | The data will be collected within 15 gameplay across 3 waves. | Player class from variable call time_taken as a list. | Box plot |

| | | | | |
|---|---|---|---|---|
| Damage taken each wave | This will help if the wave is too hard or not | It will be recorded in 15 games in a class list | From Player class in a class variable call damage_took as a list | Table |
| Distance Traveled by Enemy in Each Wave with in 10 sec | This is good because we are able to see enemies' effectiveness. | It will be recorded every time enemies move in 5 gameplay. | From distance travel in enemies class | Scatter plot |

| | Feature Name | Graph Objective | Graph type | X-axis | Y-axis |
|---|---|---|---|---|---|
| graph1 | Enemy Survival Time | Show each enemies survival time to see difficulty of the game | Histogram | Survival Time | Number of Enemies |
| graph2 | Enemy Attack Frequency | Show enemies effectiveness | Bar Chart | Enemy’s Random name | Attack_count |
| graph3 | Time taken between wave | See how fast/ good the player is | Box plot | Wave Number | Time Between Waves |
| graph4 | Damage taken each wave | This will help if the wave is too hard or not and to see damage taken | Table | Wave | Damage taken |
| graph5 | Distance Traveled by Enemy in Each Wave with in 10 sec | To see movement pattern’s effectiveness | Scatter plot | Time | Distance Traveled |

## 3.2 Data Recording Method

The game will store these data using a data class and csv file in a dictionary type using singleton pattern. The data will be recorded in 5 gameplay.

### 3.3 Data Analysis Report

The data will be represented in a line graph, histogram and scatter plot to show these 5 data.

## 4. Project Timeline

| **Week** | **Task** |
|---|---|
| 1 (10 March) | Proposal submission / Project initiation |
| 2 (17 March) | Full proposal submission |
| 3 (24 March) | Finish wave 1 |
| 4 (31 March) | Finish all wave |
| 5 (7 April) | Add miscellaneous: menu and data |
| 6 (14 April) | Submission week (Draft) |

| Checkpoint | Task |
|---|---|
| 16 april | Full game play |
| 23 april | All data collected |
| 11 May | Data report and animation |

## 5. Document version

Version: *1.0*
Date: *4 March 2025*

| Date | Name | Description of Revision, Feedback, Comments |
|---|---|---|
| 14/3 | Pattapon | Good job. I recommend adding an explanation of each chess piece's abilities, maybe as an in-game instruction.<br>Also, you can use bullet points to improve readability of the document. :) |
| 16/3 | Phiranath | Don't forget to remove the italic fonts and reformat some of the sections using a number list or a bullet point, so it would be easier to read. I suggest adding more information about the pieces promoting mechanics, is it player piece or enemy piece. Overall good idea. Good job! |
| 29/3 | Pattapon | Good job. There are some suggestions in my comments. |
| 30/3 | Phiranath | Good job! Nothing much for me to comment on, only some relationships in the class diagram and the data features. |