

# **Kingdom - Chosun Dynasty**

**2023.12.12.**

**MSc Human and Social Data Science**

**Jungyeon Lee**

**Candidate Number :** 

The game title is "Kingdom - Chosun Dynasty," inspired by the Netflix Korean drama "Kingdom." In the game, the player's goal is to save the princess from the king, who has turned into a zombie. To achieve this, the player must successfully navigate through a total of 9 storyline maps and acquire a key to access the final boss map.

Figure1. Game Map

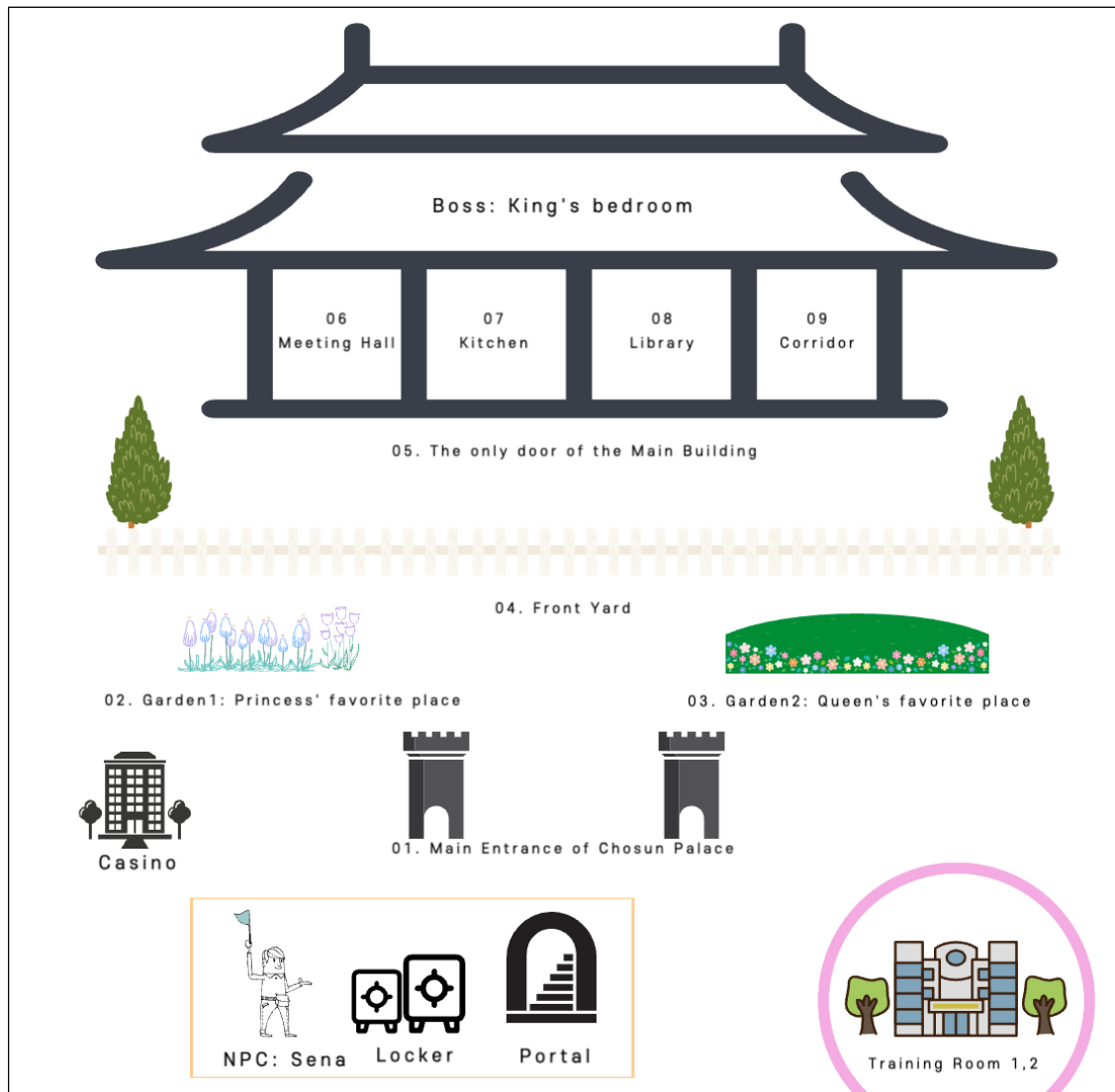


Table1. Game Map List

No	Name	Reward	Requirement (ST)
1	Main Entrance of Chosun Palace	£5	50
2	Garden1: Princess' favorite place	£10	100
3	Garden2: Queen's favorite place	£20	100
4	Front Yard	£30	150
5	The only door of Main Building	£40	200
6	The First Section: Meeting Hall	£50	300
7	The Second Section: Kitchen	£60	450
8	The Third Section: Library	£70	600
9	The Last Section: Corridor	£80	700

10	King's bedroom in Chosun Palace	Victory	1000
11	Training Room 1	£30	50
12	Training Room 2	£60	100

#### [Basic outline of the game]

Throughout each map, the player will receive rewards in the form of money and trophies. These rewards are important for the player to secure enough money, which can be used to enhance their health points (HP) and striking power. The player purchases portions and items from the store located in Sena's room.

It is essential to note that every item in the game has a different weight, and the total weight in the player's backpack cannot exceed 40 kilograms. If the total weight surpasses this limit, the player will be unable to buy additional items or pick up trophies. To manage this constraint, players must utilise the locker in Sena's room, where items can be deposited or withdrawn. Additionally, players have an option to sell or drop items to create more space in their backpack.

However, keys are an exception, as they can be added to the backpack even if there is no available space because the player cannot proceed with the game without the keys.

For a comprehensive understanding, refer to Appendix 1, which includes the UML Class diagrams detailing the game's structure.

#### [How to play this game]

Players can access all maps through the portal in Sena's room. However, if a player does not have a sufficient number of attempts or has not completed the previous stage, the portal will not function. An attempt is deducted each time a player enters a map, and re-entry is not permitted if the player successfully finishes the storyline map.

Each map features different types of zombies and varying difficulty levels. Most maps to the storyline operate in automatic mode. Consequently, if the player possesses the required striking power and ample HP, the map will be spontaneously cleared.

Conversely, if the player's striking power is lower than the map's requirement, overcoming the zombies becomes challenging, resulting in the player's defeat. Subsequently, the player will be expelled from the map, and their HP will be set to 10. To overcome this failure, players must strive to raise their HP and striking power. To help this, players can obtain a key to access Training Rooms 1 and 2 upon completing the maps titled 'Garden1: Princess' Favorite Place' and 'The First Section: Meeting Hall.'

The Training Rooms represent distinct spaces, akin to a multiverse, where players can train by fighting with other zombies. Each Training Room consists of 6 sections, and players must overcome all sections to exit successfully. The player has to manually navigate through these training rooms. Rewards, including money and trophies, are earned upon completing each section within the training room.

Moreover, players also need to manually explore some of the maps where they can obtain a key for the training rooms and the boss map. In these cases, players can choose between the 'Full Attack' and 'Lucky Attack' options. The full attack is a general method, while the lucky attack involves an element of chance. The player's striking power will be randomly set between 0.5 times to 1.5 times the original striking power. If the player wins the battle, the new striking power will replace the original, even if the new striking power is lower. Thus, players must be aware that when selecting the attack option, they may risk losing some striking power. These two attacking methods are applied in the training maps as well.

To enhance the enjoyment of the game, two lotteries automatically occur when the player successfully finishes either one of the storyline maps or each section of the training map. The first is the money lottery, where players have the opportunity to earn seven times more than the rewarded money, but this is only a 1 percent chance. Following this, the trophy lottery is also conducted, giving players a high probability of obtaining a trophy from the zombie on the map. On the other hand, players can also enter the casino to potentially earn significant rewards by betting their money. The ticket price for the casino changes with each game, and the payout is altered every time players enter the casino.

Last but not least, when the final storyline map, 'The Last Section: Corridor,' is cleared, players receive a key to access the boss map, 'King's Bedroom.' But, the king zombie is extremely powerful, and players must fight with various types of zombies as they sail the boss map. Consequently, players must ensure they have sufficient HP and striking power, surpassing the minimum requirement. Players are granted five attempts for the boss map, but if they fail on the first attempt, the next tries become much more challenging due to the health penalty.

Hence, players must find the best combination of weapons within the weight limit and utilise the lucky attack appropriately to enhance their striking power.

Overall, the main concept of this game involves developing striking power by navigating through the storyline maps and training rooms. As a reward for defeating the boss, a small gift awaits to greet players.

## [Recommendations for this game]

This game is a text-based adventure, requiring players to input numbers and strings into the text box. To run this game, players must install the Numpy, Random, and Time libraries in Python. Additionally, for a more enjoyable experience, I recommend playing the game using the 'play.ipynb' file in Jupyter Notebook rather than using the 'play.py' file in PyCharm or Spyder.

Upon starting the game, players set their character's name and choose a difficulty level between easy, normal, and hard. Each difficulty level assigns different base values for HP, striking power, and money to the player. Easy or normal mode is recommended, as hard mode is exceptionally challenging.

## [Debugging and previous issues]

To capture errors in this game, I used the Spyder program to promptly identify any faulty code during the initial development of this adventure. Additionally, I applied the 'try and except' method, specifically utilizing 'Exception as e,' to identify and rectify problematic sections of the code. I also employed the 'Unit-test' on some of the functions to check whether they were working correctly. Moreover, I played through this game more than 30 times to ensure the seamless functionality of all features.

Figure2. Debugging Process

```
In [7]: game_play()

Welcome to King's bedroom in Chosun Palace, you have to command some instructions from now on
[You are now in 1F Corridor]
[You must fight with the creepy eunuch]
[You can choose two attack options]
1. Full Attack (FA) | 2. Lucky Attack (LA): 1
Invalid option. Please choose FA or LA
1. Full Attack (FA) | 2. Lucky Attack (LA): fa
An error occurred: 'Boss_map' object has no attribute 'win_count'
```

```
# -*- coding: utf-8 -*-
"""
Created on Sat Dec 9 20:07:33 2023

@author: ijeong-yeon
"""
import unittest
from Kingdom import *

class Test(unittest.TestCase):

    def setUp(self):
        self.map1 = Normal_map("Main Entrance of Chosun Palace", 5, 3, 1, 10, 1) # map
        self.map2 = Training_map("Training Room 1", 30, 5, 10, 50) # Training map
        self.map2.map_setting(map_list = ["1F Corridor", "2F Corridor", "Chichester1 016", "Chichester1 017", "Chichester1 021",
                                          "Undergraduate Student", 2), ("Postgraduate Student", 5), ("Exchange Student", 7),
                                          ("TA", 12), ("Professor", 14)])
        self.player1 = Player("Jungyeon", 27)
        self.player2 = Player("Omer", 31)
        self.item1 = Weapon("Revolver", 200, 10, 1, 300)
        self.item2 = Key("Training Key1", 1, 0.5, 1, 10)
        self.map2.key_setting(self.item2)

    def test(self):
        self.assertTrue(self.map1.hp_check(self.player1))
        self.player1.hp = 0
        self.assertFalse(self.map1.hp_check(self.player1))

        self.player2.add_item(self.item2)
        self.assertTrue(self.map2.check_key(self.player2))

        self.map2.win_count = 6
        self.assertTrue(self.map2.win_check(self.player2))

        self.map2.add_item(self.item1)
        self.assertEqual(self.player2.striking_power_check(), 300)
        self.assertTrue(self.player2.weight_check(), 10)
        self.assertTrue(self.player2.drop_item(self.item1))
        self.assertEqual(self.player2.striking_power_check(), 0)
        self.assertTrue(self.player2.weight_check(), 0)

    def main():
        a = Test()
        a.setUp()
        a.test()

main()
```

```
IPython 8.12.0 -- An enhanced Interactive
Python.

In [1]: runfile('/Users/ijeong-yeon/
OneDrive - University of Sussex/Programming
Through Python/Coursework/Debugging.py',
wdir='/Users/ijeong-yeon/OneDrive -
University of Sussex/Programming Through
Python/Coursework')
You just got a Training Key1
You just got a Revolver

In [2]:
```

When I developed this game, my focus was on building a smooth display, as it is a text-based game. Initially, it was challenging to create a stable and well-functioning text box throughout all functions and maps. So, I used the 'try and except' code and 'isnumeric' code to maintain the text box. On the other hand, it felt awkward to print numerous sentences without any intervals. Consequently, I discovered and implemented the 'time.sleep()' function to solve this challenge. Additionally, I believe that excessively stacked commands over the text box can disrupt the gaming experience. To fix this, I incorporated the 'clear\_output()' code into most functions to maintain a tidy text box. Nevertheless, it is important to highlight that both PyCharm and Spyder lack support for the 'clear\_output()', 'display' and 'Image' codes from IPython.display library, which is utilised to make an interval between texts, and display images. Therefore, to fully relish the gaming experience, it is advisable to use Jupyter Notebook.

## Appendix 1. UML Class Diagrams

