Assignment Case	
CH2	BINUS UNIVERSITY
Periode Berlaku Semester Genap 2022/2023 ***Valid on*** *Even Year 2022/2023*	**Software Laboratory Center** **Assistant Recruitment 23-2**

Note: Please focus on the main logic and main feature!
(Splash screen and design are not scored)

Materi
Material

- Review Data Structure
- Multiple Linked List
- Hash Table

Soal
Case

JeDi's Adventure

JeDi's Adventure is a puzzle type 2D gacha game. Player can will have to move around solving puzzle and then finding all the treasure in some kind of dungeon. The story is about some adventurer that wants to follow the step of a Legendary Adventure named "Jedi". This legendary adventure left all his treasure inside the dungeon that will be uncovered by the player of this game.

You as one of the developers of this game, will have to made the program from scratch. Here are the detail of the game:

➢ **Main Menu**

- The Game will have **4 main menu** that are **"Start New game",** "**How To Play**", "**High Score**" and "**Exit**".

- **Prompt** user to **input chosen menu**.


```
JeDi's  Adventure!
==================
1. Start New Game
2. How To Play
3. High Score
4. Exit
>> _
```

Figure 1. Main Menu

1. If user choose **Start New Game** (**Menu 1**), then:

   - **Prompt** user to **input difficulty level** either **Beginner**, **Normal**, or **Hard**. Users will have to input either **1 (Beginner Level)**, **2 (Normal Level)**, **3 (Hard Level)**, or **4 (Go Back to Main Menu)**.

```
Choose Difficulty
==================
1. Beginner [You can move freely] [Less Score & Item]
2. Normal [You will lose after certain move] [Normal Score & Item]
3. Hard [You need to be more carefull to move] [More Score & Item]
4. Back to Main Menu
>>
```

Figure 2. Choose Difficulty

   - Each **difficulty level** will have certain conditions that will be explained later.
   - After that **prompt** user to **input name**. **Validate** the input must be between **5 – 50 Characters**.

```
Input Your Player Name [5-50 characters]: The One and Only▄
```

Figure 3. Input Name

   - After input the **player's name**, the game will **Start**.
   - There is **9 Map** for this game that the player can explore in game. You will need to **read** every **.txt map file** in the "**Maps**" folder and then store it using **Multiple Linked List**. Here are the connections for each map that you will need to make.
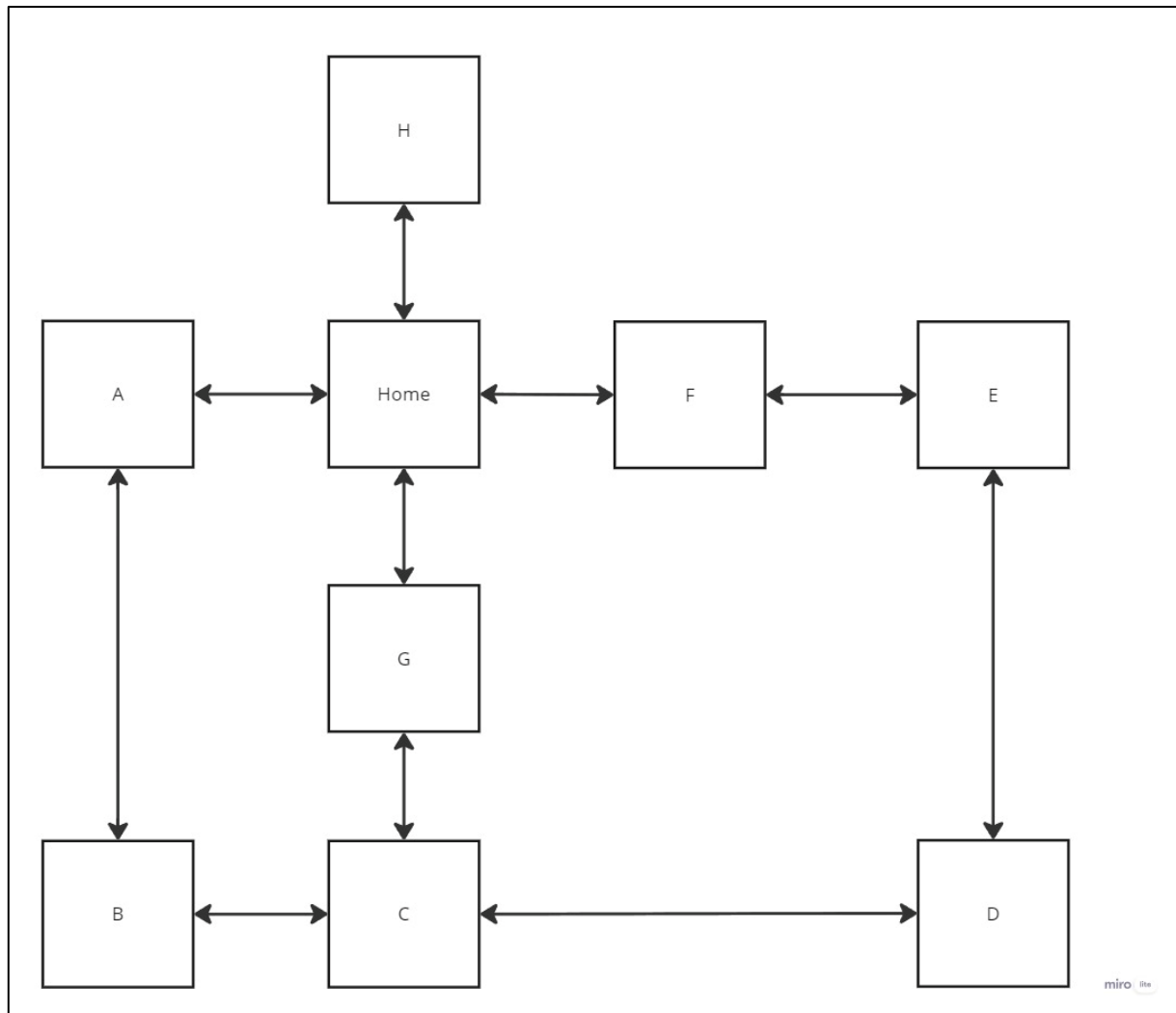
Figure 4. Map Multiple Linked List Overview

o   The **player** will start at the Spawn Point (**'S'**) of map "**Home**".

o   In the game, you must **show** the **current map** with the **player** in it that symbolized by **'P'.**
You must also **show** the **player's score** that will **start by 0** at the beginning.

```
###############-------------###############
#     ^^^      #    ^      ^    #            ^^C #
#    C         #    ^      ^    #             ^^ #    The One and Only
#    ^^^       #    ^      ^    #          ###### Score: 0 pts
#########OO#########    #######      ####    |
#                                    #       |    Chest Left: 40
#                                    #OO######
#                                           #
#                                           #
#                                           #
#                #    #                      #
#                 P                          #
#                #    #                      #
#                                           #
#                                           #
|                                           #
|                                           #
|                                           #
#############################################
#^^##    ##          ##       ^C#    O   #        #
#C^#   O   #########    #^^^^^##   O    #        #
#^^   OCO                 O             #  O     #
#^^#   O   #########    #^^^^^####### #        #
#^^##    ##          ##       ^C#C   ^#        #
###########################################-----####
Press 'Escape' to pause
```

Figure 5. Start Game

- o When the game starts, you also need to prepare **the list of items** that the player can get from obtaining **chest**. **The list of items** is obtained by **reading** the **ListItem.csv** file and you must **store** it using **Hash Table Chaining**. The **Hash Table** of **List Items** will have the following details:

  - Each **item** in **ListItem.csv file** has **name**, **rarity**, and **score**.
  - The **hash key** will be determined by the **rarity of each item** with the following **format**.

Rarity	Key
Common	0
Uncommon	1
Rare	2
Epic	3

Legend	4

- The **hash table size** must be **5** and each **hash key** will have **chaining** using a **single linked list** or **double linked list**.

- You will need to make **2 hash tables** for the **overall List Item** and for the **Player Inventory**. These **2 hash tables** will have the **same structure**.

o The player can **move around** by **typing** the **arrow keys in keyboard** to go **up (↑)**, **down (↓)**, **left (←)**, and **right (→)**.

```
# P #
   S
#    #
```
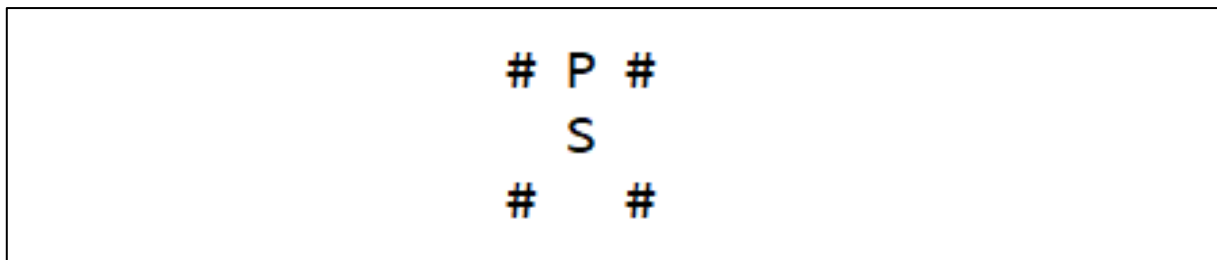
Figure 6. Player Move Up

```
#    #
  PS
#    #
```
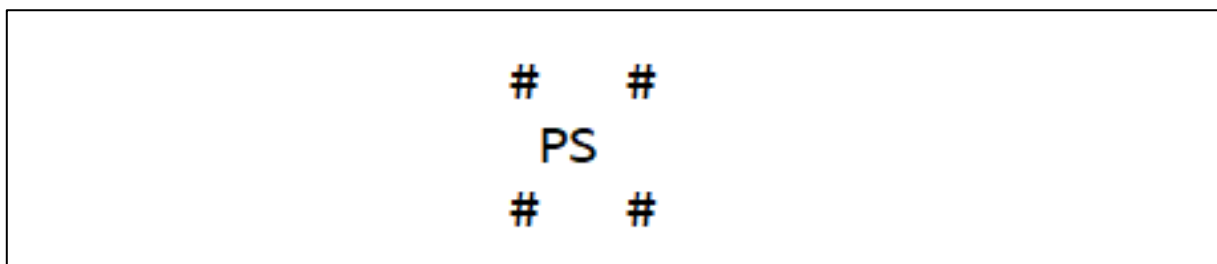
Figure 7. Player Move Left

```
#    #
   S
# P #
```
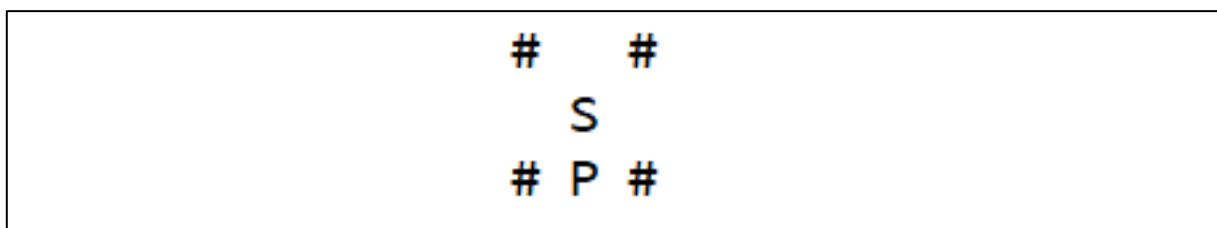
Figure 8. Player Move Down

```
#    #
  SP
#    #
```
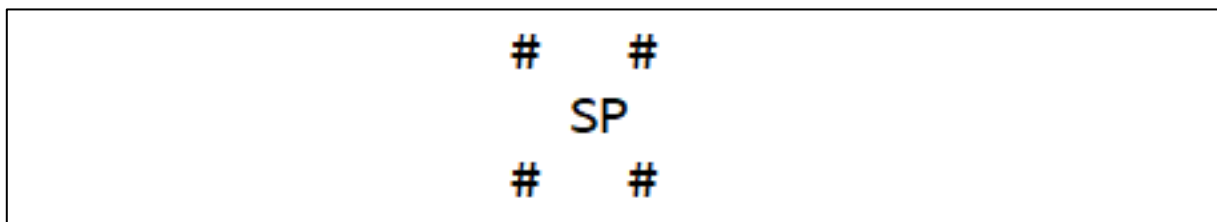
Figure 9. Player Move Right

- o When moving around, you must add below conditions:
  - **Validate** the player **cannot go through the wall ('#')**.
  - If the player **moves** into a **boulder ('O')**, the **boulder** will also **move straight to the player move direction**. You need to also **validate** the **boulder cannot go** through **the wall** and **chest ('C')**

```
###############-------------###########
#      ^^^      #    ^      ^     #
#    C          #    ^      ^     #
#    ^^^    O   #    ^      ^     #
#########O ########      #######      ###
#          P                         #
```
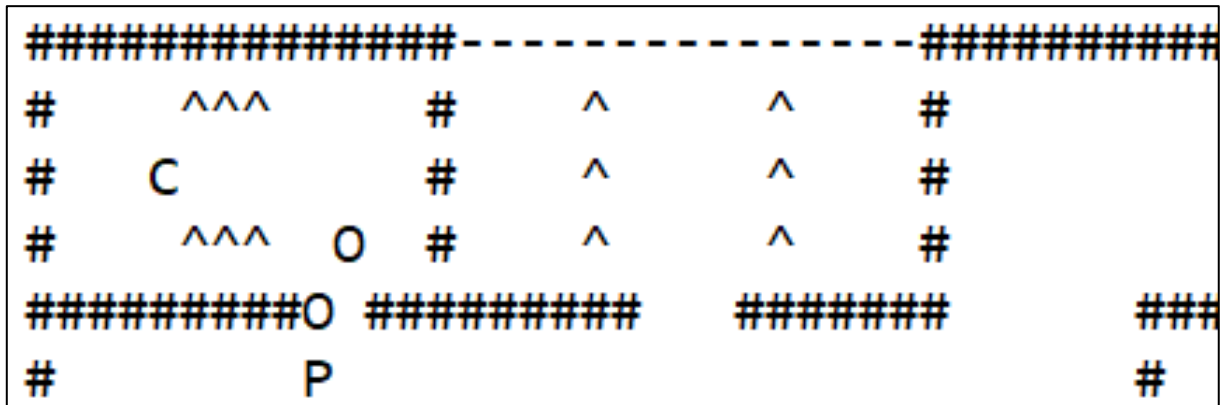
Figure 10. Player Move Boulder (Before)

```
###############-------------###########
#      ^^^      #    ^      ^     #
#    C          #    ^      ^     #
#    ^^^   OO   #    ^      ^     #
#########P ########      #######      ###
#                                    #
```
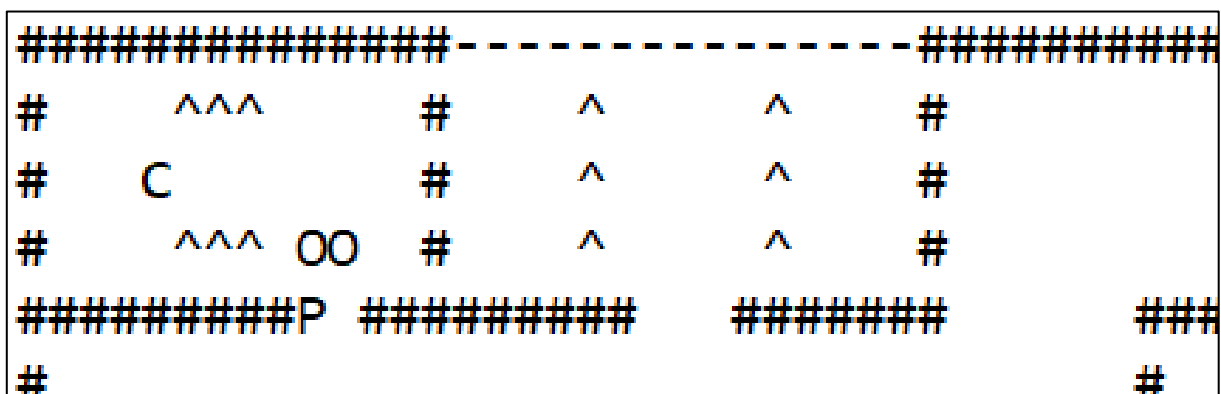
Figure 11. Player Move Boulder (After)

- If the player **moves** into a **spike ('^')**, then the player will be **teleported** to the **last visited spawn point ('S')**. **Validate** if the **last visited spawn point** has **boulder** on **top** of it, when player **teleported to it**, **end the game**.

```
####################        ####################
#                              ^^^^^^C#
#                              P^^^^^#
###############-------------###############
```
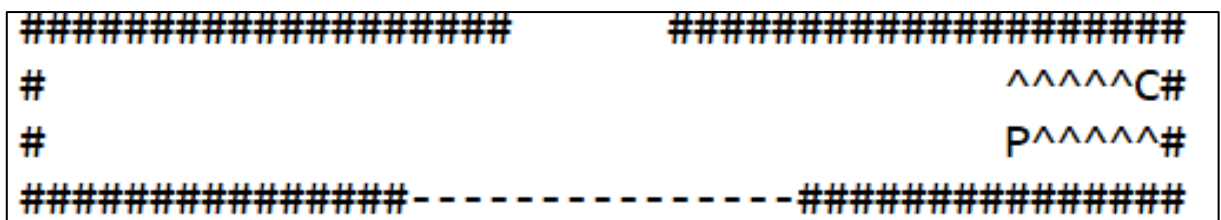
Figure 12. Player Move to Spike (Before)

```
###############----------------################
#    ^^^       #     ^       ^      #           ^^C #
#    C         #     ^       ^      #            ^^ #
#    ^^^       #     ^       ^      #         ###### 
#########OO##########     #######      ####    |
#                                          #       |
#                                          #OO######
#                                               #
#                                               #
#                                               #
#                  #     #                      #
#                     P                         #
#                  #     #                      #
#                                               #
#                                               #
#                                               #
|                                               #
|                                               #
|                                               #
###############################################
#^^##    ##           ##        ^C#    O   #        #
#C^#   O  #########   #^^^^##  O    #              #
#^^   OCO               O            #  O          #
#^^#   O  #########   #^^^^####### #               #
#^^##    ##           ##        ^C#C   ^#          #
#####################################-----####
```

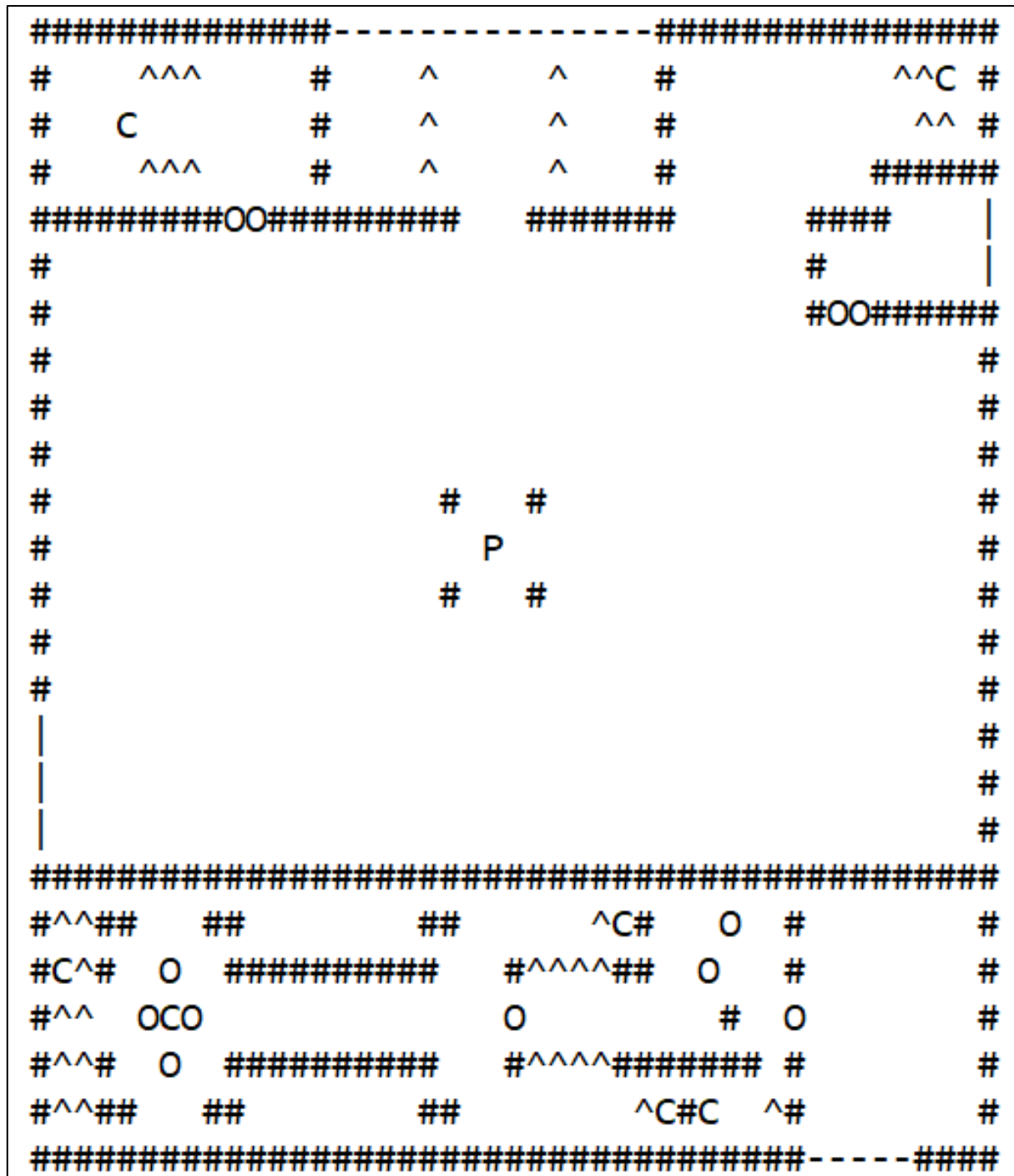*Figure 13. Player Move to Spike (After)*

- If the player **moves** into a **new spawn point** (the one that is **not the last visited**), then update the **last visited spawn point** into that **spawn point**.

```
#####################------#########################
                                    #            #
   #############                         PS      #
   O^O^O^O^O^O^O                    #            #
   #############                         #### #####
```

Figure 14. Player Move to New Spawn Point (Before)

```
Spawn Point Updated!
Press Enter to Continue...
```

Figure 15. Update Spawn Point Message

```
#####################------#########################
                                    #            #
   #############                         P       #
   O^O^O^O^O^O^O                    #            #
   #############                         #### #####
```

Figure 16. Player Move to New Spawn Point (After)

- If the player moves the **boulder** ('**O**') into a **spike** ('**^**'), those two objects will **collide and disappear.**

```
###############--------------------###############
   ^^^ O    #     ^        ^     #              ^^C #
   C        #     ^        ^     #         PO^^ #
   ^^^      #     ^        ^     #         ######
########   #########   #######    ####        |
```

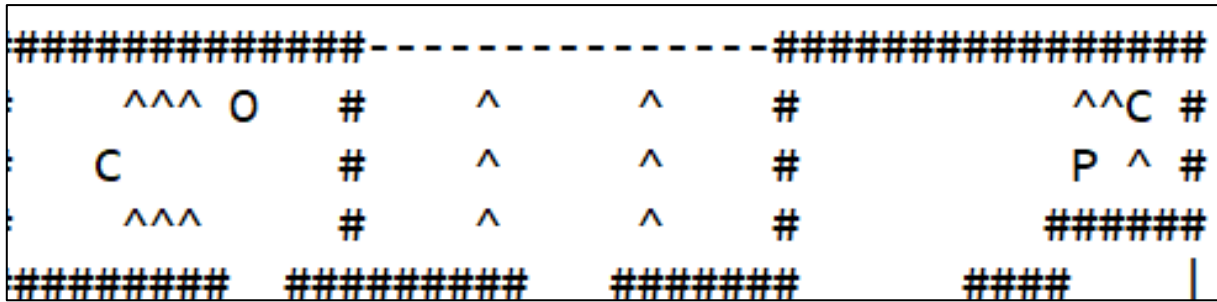*Figure 17. Boulder and Spike Collide (Before)*

Figure 18. Boulder and Spike Collide (After)

- If the player **moves** into **a chest ('C')**, the game will then **random** an **item** from the **List Item Hash Table**. The item will then be **stored** into **Player Inventory Hash Table**. **Validate** that the player will **not get the same item** in their **inventory**. After inserting the **item**, **add** the **player score** with the **item total score** that **obtained** from the **multiplication of item score** with **the score multiplier**. The **rarity chances** and **score multiplier** for each **difficulty level** will be **different** with the detail below.

Difficulty Level	Item Get Chances	Item Score Multiplier
Beginner	Common (60% chance) Uncommon (35% chance) Rare (5% chance)	1
Normal	Common (40% chance) Uncommon (25% chance) Rare (20% chance) Epic (10% chance) Legend (5% chance)	2
Hard	Common (20% chance) Uncommon (25% chance) Rare (30% chance) Epic (15% chance) Legend (10% chance)	3

```
###############-------------###############
#   ^^^    #   ^     ^     #        ^^C #
#  CP  O   #   ^     ^     #         ^^ #  Budiman
#   ^^^    #   ^     ^     #     ###### Score: 0 pts
######### O#########  #######    ####    |
```

Figure 19. Player Get Chest (Before)

```
You Get:
Letter [Common]

Score        5 pts
Multiplier   1x
------------------
Total Score  5 pts

Press Enter to Continue..._
```

Figure 20. Player Get Chest Message with Difficulty Level Beginner

```
###############-------------###############
#   ^^^    #   ^     ^     #        ^^C #
#  P   O   #   ^     ^     #         ^^ #  Budiman
#   ^^^    #   ^     ^     #     ###### Score: 5 pts
######### O#########  #######    ####    |
```

Figure 21. Player Get Chest (After)

- Each time the player **moves**, for **difficulty level** that are **not beginner**, it will **cost a step**. If the **player's total step** left is **0**, then the **game** will be **ended**. This **total step** can be **increased** by **collecting chest** and **colliding** the **boulder** and **spike**. The details are explained below.

Difficulty Level	Starting Steps	Chest Extra Step	Collision Extra Step
Normal	750	80	15
Hard	500	75	15

- If the player **moves** into **the entrances** ('**|**' or '**-**'), then **move** the player into the **entrance of other map**. It will **move** the player to the map that is **connected** based on the **map multiple linked lists.**

```
###############--------------------###############
#       ^^^           #       ^  P  ^        #                    ^^C  #
#    C                #          ^     ^        #                        ^^     #
#       ^^^           #          ^     ^        #                      ######
########OO#########        #######         ####        |
```

Figure 22. Player Below the Top Entrance of "Home" Map

```
#######################################################
#                        #              #                                   #
#      ##########   O          O    ##                            #
#      ##                       O          O    ##                            #
#      ##                       #              #   ##                        #
#      ##  OO  ####      #              #   ##      OO             #
#      ##          ####      #              #   ##                        #
#      ##          C##       #              #   ##C              ##       #
#      ##########   #              #   #############        #
#                        #              #                                   #
#          ^##^          #              #              ^##^             #
#          ^##^          #              #              ^##^             #
#          ^##^          #              #              ^##^             #
#          ^##^          #              #              ^##^             #
#          ^##^          #              #              ^##^             #
#          ^##^          #              #              ^##^             #
#          ^##^          #              #              ^##^             #
#          ^  C^          #              #              ^C  ^             #
#          ^##^          #              #              ^##^             #
#          ^##^          #              #              ^##^             #
#                        #              #                                   #
##################        ####################
#                                                                      ^^^^^^C#
#                               P                                  ^^^^^^#
################--------------    ###############
```

Figure 23. Player Move From "Home" to "H" Map

- o Players can press "**Escape**" **button on the keyboard** to **pause the game**. When the player **pauses** the **game**, **show** the **Pause Menu**.

```
Paused! [Press 'Escape' Again to unpause]
============
1. Item List
2. End The Game
>> ▪
```

Figure 24. Pause Menu

- o **Prompt** player to **input** the menu with following details:
  - ▪ If the player **input number 1**, then **shows** the **item list** from **Player Inventory Hash Table**.

```
==================================================================
| No | Rarity      | Item Name               | Score |
==================================================================
| 1  | Common      | Letter                  | 5     |
| 2  | Common      | Leather Clothes         | 15    |
| 3  | Common      | Knitt                   | 30    |
| 4  | Common      | Book                    | 35    |
| 5  | Common      | Rope                    | 10    |
| 6  | Common      | Bandage                 | 40    |
| 7  | Common      | Wet Umbrella            | 35    |
| 8  | Common      | Chestplate              | 20    |
| 9  | Common      | Rubber Ban              | 10    |
| 10 | Common      | Coat                    | 40    |
| -------------------------------------------------------------- |
| 1  | Uncommon    | Silver Belt             | 50    |
| 2  | Uncommon    | Wooden Cross            | 40    |
| 3  | Uncommon    | Walky Talky             | 60    |
| 4  | Uncommon    | Christmas Globe         | 40    |
| -------------------------------------------------------------- |
| 1  | Rare        | Gold Sack               | 270   |
| -------------------------------------------------------------- |
| 1  | Epic        | Demonic Gemstone        | 1000  |
| 2  | Epic        | Forgotten Samurai Armor | 450   |
| 3  | Epic        | Extraordinary Bone      | 750   |
| -------------------------------------------------------------- |
| 1  | Legend      | Divine Rapier           | 1750  |
| 2  | Legend      | Ghost Ring              | 1500  |
| 3  | Legend      | JeDi's Diary            | 2000  |
==================================================================
```

Figure 25. Player's List Item

- If the player **input "Escape"**, then **go back** to the **Game**.

- If the player **input number 2**, then **the game** will be **ended**.

o The **game** can also **end** when **all the chest** has been **collected** by the player. After **all the chest collected**, if there are **leftover steps** for the player that chooses **difficulty** other than **beginner**, add the **player's score** with **bonus score** that **obtained** from the **multiplication** of **leftover steps** with the **leftover steps multiplier**. Below are **the leftover steps multiplier** details:

Difficulty Level	Leftover Steps Multiplier
Normal	50
Hard	200

o After the **game ends**, you will need to **reset the map**, **player's score**, **player's position**, and **item list**. You will also need to **write** the **player's score** into **highscore.txt file** with the following **format**.

[PlayerName]#[Score]

2. If user choose **How to Play** (**Menu 2**), then:

o **Show** the information about **how to play the game**, **difficulty details,** and **each symbol meanings**.

o You are **free to create** any kind of **How to Play menu** with all of **those information delivered**.

3. If user choose **High Score** (**Menu 3**), then:

o **Read** the **highscore.txt file** and **show** all **player's score** from the **highest** into **lowest**.

o You must **sort the score** using **cycle sort algorithm**.

```
================================================================================
| No. | Name                            | Score            |
================================================================================
| 1   | Brody                           | 121760           |
| 2   | Jedi's Successor                | 78540            |
| 3   | JokoAja                         | 1090             |
| 4   | Budiman                         | 180              |
| 5   | Budiman                         | 5                |
| 6   | JoshAja                         | 0                |
================================================================================
Press Enter to Continue...
```

Figure 26. High Score Menu

4. If user choose **Exit** (**Menu 4**), then close the program.

➢ **Hidden Command**

- When the player **plays** the **game**, there is a **hidden command** that they can use. If the player press **"Space Bar" on the keyboard**, they will **activate God Mode**. **God Mode** can make the player **go through boulders and spikes**. They can **deactivate** it by pressing "**Space Bar**" again. The **God Mode** will also **reset** when the **game ends**.

```
#################################################
#C^^      #                    P    ^^^^^^C^#
# #^C#    #                    #########  The One and Only
# ####   #####                 ^    O |  Score: 0 pts
# #   O       #########        #####O  |
# #                    #                |  Chest Left: 40
# ########            #        ######### God Mode Activated!
```

Figure 27. God Mode (Before)

```
#################################################
#C^^      #                         P^#
# #^C#    #                    #########  The One and Only
# ####   #####                 ^    O |  Score: 5 pts
# #   O       #########        #####O  |
# #                    #                |  Chest Left: 39
# ########            #        ######### God Mode Activated!
```

Figure 28. God Mode (After)

Please run the EXE file to see the sample program.
Good Luck! 😊

Komponen
Component

Material	Score Percentage
Main Menu	5%
Multiple Linked List Map	25%
Hash Table Item	25%
Game Movement	14%
Chest Logic	10%
Post Game Logic	8%
High Score	4%
Others Logic	9%