


Assignment Case	
CH3	
Periode Berlaku Semester Genap 2022/2023 Valid on Even Year 2022/2023	Software Laboratory Center Assistant Recruitment 23-2

Materi

Material

- AVL Tree
- Minimum Spanning Tree (Prim and Kruskal)

Soal

Case

Grocktunnel

Golden, a creative young entrepreneur, started as an eager gamer and eventually led his own successful gaming company. Known for his fresh ideas and passion for unique games, his latest venture, **Grocktunnel**, is set to be another exciting breakthrough in the gaming world.

Golden has chosen you as the head developer to create **Grocktunnel's program**, an engaging exploration game. Players will explore a constantly **changing maze**, designed by the mythical Grock the Earth Shaper. The maze's layout, driven by either **Prim's or Kruskal's algorithm**, **changes each time**, making every game a new adventure.

➤ Home Page

- This menu contains 4 menus, which are **Login**, **Register**, **Highscore**, and **Quit**.
- **Prompt** user to **input chosen menu**. **Validate** the input must be **between 1 and 4 inclusively**.

```

Grocktunnel
Welcome to Grocktunnel,
-----
1. Login
2. Register
3. Highscore
4. Quit
>>

```

Figure 1. Home Page

1. If the user chooses **Login (Menu 1)**, then:
 - **Prompt** user to **input name** and **password**, user can go **back** by **inputting 'q'** (case sensitive). **Validate** the **name** and **password** must be **valid**.
 - If the **credentials** are **valid**, **redirect** the user to **Choose Maze Generator Page**.

```
Login
-----
Input name ['q' to go back]: golden
Input password: |
```

Figure 2. Login Page

```
Login
-----
Input name ['q' to go back]: golden
Input password:
Invalid Credential!
Enter to continue...|
```

Figure 3. Invalid Credential Login Page

- To **validate** the **inputted credentials**, you must **read all the user's data** from the **"user.txt"** file, the user data is stored in this format.

```
Name#Password#Highscore
```

Figure 4. User Data Format

2. If the user chooses **Register (Menu 2)**, then:
 - **Prompt** user to **input name** and **password**, **validate** the **name** must be **unique** and the **password** must be **minimal 5 characters** long.

```
Register
-----
Input name [must be unique]: golden
Name is not unique !
Input name [must be unique]: janganbohonglagiyaa
Input password [minimal 5 char]:
Invalid password !
```

Figure 5. Invalid Register Page

- If the inputted **data** is **valid**, **write the new user's data** into the **"user.txt"** file using this **format**, with the **new user's high score** set to **0**. Then **redirect** the user back to **Home Page**.

```
Name#Password#Highscore
```

Figure 6. User Data Format

```

Register
-----
Input name [must be unique]: janganbohonglagiyaa
Input password [minimal 5 char]:
Register Success !
Enter to continue...

```

Figure 7. Successful Register Page

3. If the user chooses **Highscore (Menu 3)**, then:
 - **Show** all **users' name** and their **high scores**, make sure to **sort** them in **ascending order by their high score** and it must be **sorted** by using the **heap sort algorithm**.

No.	Name	Highscore
1.	golden	474
2.	halo	369
3.	josua	363
4.	kevin	304
5.	jesi	222
6.	okeoke	0
7.	ciu	0
8.	janganbohonglagiyaa	0
9.	asdasdasd	0

Enter to continue...

Figure 8. Highscore Page

4. If the user chooses **Quit (Menu 4)**, then show some goodbye message and **exit** the program.

➤ Choose Maze Generator Page

- This menu contains **3** options: **Prim**, **Kruskal**, and **Exit**.

```

Hello golden !
Choose your preferred maze generator,
-----
1. Prim
2. Kruskal
3. Exit
>> |

```

Figure 9. Choose Maze Page

1. If the user **chooses Prim or Kruskal**, generate a maze using the **chosen algorithm** with the **size of 51 x 31** (width x height), you are required to represent the **wall** of the maze with **block symbols (ASCII 219)** and the **path** with **space (' ')**. Also, generate **flashlights** represented with **('* ')**, the flashlights will be **generated randomly across the map**, with a **5.88% chance (1 in 17 available paths will be replaced with a flashlight)**. Then continue to **Gameplay Page**.
2. If the user **chooses Exit**, **redirect** the user back to the **Home Page**.

- Before the game starts, there will be a **confirmation page** to start. Here, when the user hit **enter**, the **timer** for the game **will start**. And user will be immediately spawned into the map.

```
Ready ? Press enter to start...
```

Figure 10. Confirmation Page

- The **user** will be **spawned** at the **top left corner** of the **maze**, the **user** will be **symbolized** as ('P').
- The **exit** of the maze will be **placed** at the **bottom right corner** of the **maze**, **symbolized** as ('E').

```
#####
#####
##  ##
## p ##
##  ##
##  ##
#####
```

Figure 11. Gameplay (Start)

- By **default**, the **user's vision** of the maze will be **5 x 5**, meaning that the user can only see the maze **within their vision**, and the **user** will **always** be placed **in the center**. The **5 x 5 vision** **doesn't include** the **border**.
- Make sure to **add** the **border** for the user's vision, the **thickness** of the border is **1 character**. The **border** of the user's vision will be **symbolized** as **'#'**.
- **If** the user's **vision exceeds** the **maze/map**, **symbolized** the **outside** of the map with **'#'**.
- The user can **move** by **pressing certain keys** on their keyboard, **'w'** to go up, **'a'** to go left, **'s'** to go down, and **'d'** to go right (case sensitive). **Validate** the **user** so they **can't go through** a wall.
- When the **user moves**, **make sure** they are **always placed** in the **center of their vision**.

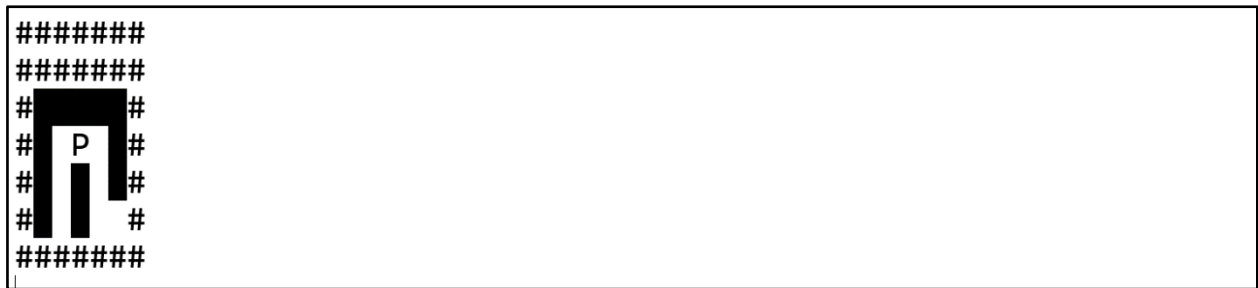


Figure 12. Gameplay (Move)

- There will be **flashlights** scattered everywhere around the map, the **user** can **pick up** those **flashlights**, and when they do, their **vision** will be **enhanced to 9 x 9**. The **flashlight** will **run out of battery after 5 seconds**, so when that happens **the user's vision** will be **reverted to normal**.



Figure 13. Gameplay (Encountering a Flashlight)



Figure 14. Gameplay (Increased Vision)

- The **user** can **activate** a **cheat** when they entered the right secret code. If the **user** **consecutively types 'bluecactus'**, the **cheat will be activated**, when that happens, **show a message**, and **increase the user's vision to 19 x 19**.

User's Time (in seconds)	Multiplier	Point Class
≤ 60	3	60
≤ 120	2	120
≤ 180	1.5	180
> 180	1	User's Time

$$\text{Bonus Points} = (\text{Point Class} - \text{User's Time}) \times \text{Multiplier}$$

Figure 19. Bonus Points Calculation

$$\text{Main Points} = \text{Initial Points} \times \text{Multiplier}$$

Figure 20. Main Points Calculation

$$\text{Total Points} = \text{Main Points} + \text{Bonus Points}$$

Figure 21. Final Points Calculation

- For example, the user finished in 193 seconds.
 - Bonus points = $(193 - 193) \times 1 = 0$
 - Main points = $(100 \times 1) = 100$
 - Total points = $0 + 100 = 100$

```
Main point: 200
Bonus point: 116
-----
Total point: 316
|
```

Figure 22. Points Calculation

- Then **check if the current user's high score is less than the total points**, if the **current user's high score is lesser than the total points**, **update the user's high score** inside the "user.txt" file. After that, **redirect** the user **back** to the **Home Page**.

Please run the EXE file to see the sample program.

Komponen Penilaian

Scoring Component

No	Component	Weight
1	Menu	6
2	File	10
3	Generate Maze	32
4	AVL Tree	20
5	Gameplay	22
6	High score	5
7	Logic	5