|  |  |
| --- | --- |
| **Onsite Case** |  |
| CH4S2 |
| **Periode Berlaku** Semester Ganjil 2020/2021  ***Valid on*** *Odd Year 2020/2021* | **Software Laboratory Center**  **Assistant Recruitment 21-1** |

## Materi

*Material*

* Shortest Path

- Dijkstra

- A-Star (A\*)

## Soal

*Case*

**SUDUT and Precious Star**

**SUDUT and Precious Star** is a console game application which the game to help SUDUT (SW16-2) collected his precious star. The game consists of the following features: **play game**, **build map,** and **game setting**. You, as a **trainee** of **21-1 Generation** from **Software Laboratory Center**, are asked to create this program using **C Programming Language** with these following requirements:

* In the beginning the program will:
* **Play** “**nar.wav**” song and **repeat** when the song finished
* **Read** “**map.sw**” file for the default map in the game

|  |  |
| --- | --- |
| Value | Information |
| 0 | Represent the floor |
| 1 | Represent the wall |

* **Show** the game main menu that consists of:
* The main menu UI Interface:

|  |  |  |
| --- | --- | --- |
| Position | UI Screenshot | Information |
| Left Side |  | * Currently active map * Use light green color for the wall |
| Right Side |  | * Game Title and Menu * Consist of 5 Menus:  1. Play Game 2. Buiild Map 3. Default Map 4. Game Setting 5. Exit |

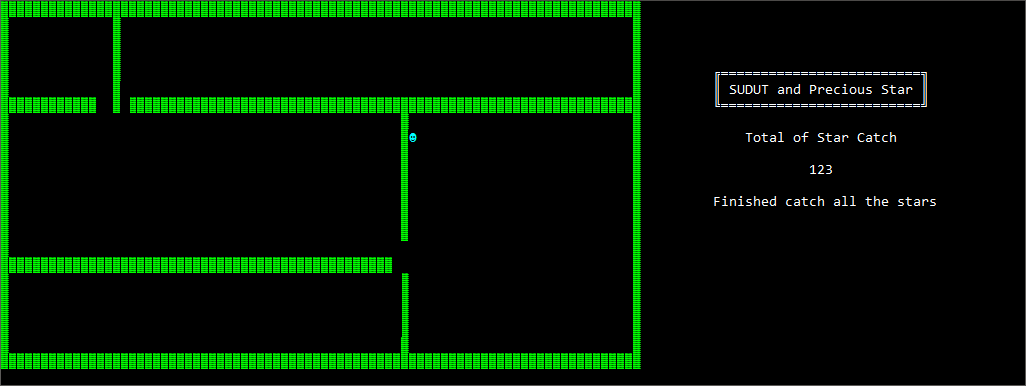
* If the users choose **menu 1** “**Play Game**”, then:
* The Play Game UI Interface:

|  |  |  |
| --- | --- | --- |
| Position | UI Interface Screenshot | Information |
| Left Side |  | * Game map area |
| Right Side |  | * Game Title * Number of total star catch |

* **The detailed game tile** information:

|  |  |  |  |
| --- | --- | --- | --- |
| Tile | Image | Color | Initial Count |
| Wall |  | Light Green | Depends on the map |
| Floor |  | Black | Depends on the map |
| Player |  | Light Aqua | One (1) |
| Star |  | Light Yellow | Between 80 and “Maximum Start” game setting value |
| Target Star |  | Light Red | One (1) |

* **The detailed gameplay** information:
* **Generate a star position**, which the start position must be **inside the game map area** and **on the floor location**. The number of total stars is **between 80** and **the** “**Maximum Star**” **game setting value**
* **Generate a player position**, which the player position must be **inside the game map area** and **on the floor location**
* **The game movement** information:
* **Change** the target star color to **light red color**
* The player **movement** using **Dijkstra or A\* Algorithm** to catch the star
* **Each time catch the star**, ~~then~~ **increase** the number of total star catch value
* **Repeat** until the player catches all stars on the map
* After catch all the stars, then show “**Finished catch all the starts**” message and **redirect** to the main menu



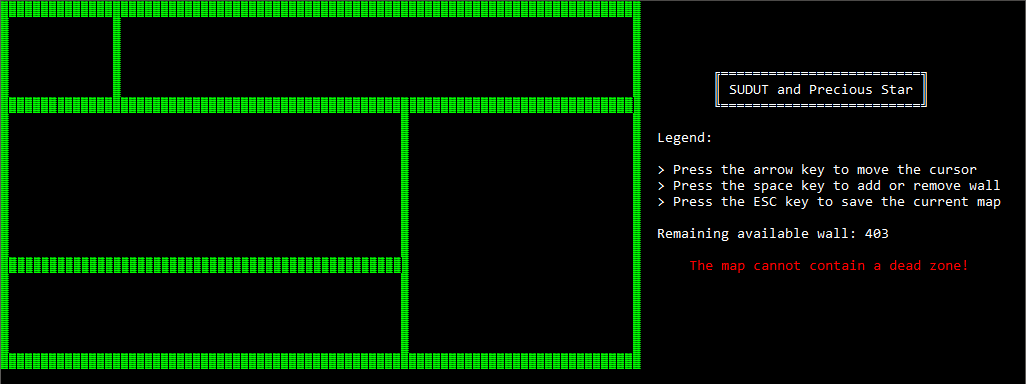
**Figure 1. After catch all the stars**

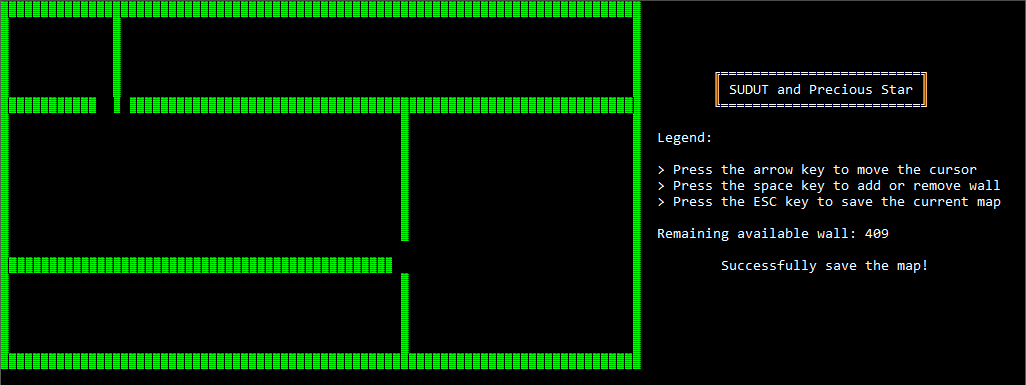
* If the users choose **menu 2** “**Build Map**”, then:
* The Build Map UI Interface:

|  |  |  |
| --- | --- | --- |
| Position | UI Interface Screenshot | Information |
| Left Side |  | * Created map area * Use light green color for the wall color |
| Right Side |  | * Game Title and Legend * Remaining available wall status |

* **The detailed key information** for the build map:

|  |  |
| --- | --- |
| Key | Information |
| Arrow Key | * Move the cursor position based on the arrow key press * Validate the cursor must be inside the created map area |
| Space Key | * On floor location, then fill with a wall and decrease remaining available wall * On wall location, then remove a wall and increase remaining available wall * Validate only fill a wall with maximum count based on “Maximum Wall” game setting value |
| ESC Key | * Validate the created map cannot contain a dead zone (using Flood Fill algorithm) * Save the created map to the active map and redirect to the main menu |

**Figure 2. The map contain a dead zone**



**Figure 3. Successfully save the map**

* If the users choose **menu 3** “**Default Map**”, then:
* **Change** the game active map using “**map.sw**” file
* **Redirect** back to the main menu
* If the users choose **menu 4** “**Game Setting**”, then:
* The Game Setting UI Interface:

|  |  |  |
| --- | --- | --- |
| Position | UI Interface Screenshot | Information |
| Left Side |  | * Available Game Setting |
| Right Side |  | * Game Title and Legend |

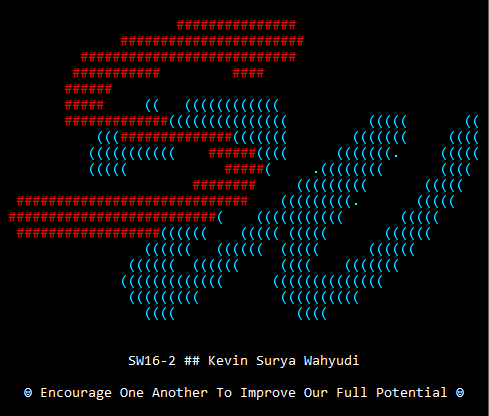
* **The detailed information** for the game setting:

|  |  |  |  |
| --- | --- | --- | --- |
| Game Setting | Value | Default Value | Information Setting |
| Maximum Star | Between 100 and 500 (Inclusive) | 300 | Determine maximum star available in the play game menu |
| Maximum Wall | Between 400 and 700 (Inclusive) | 550 | Determine maximum wall available in the build map menu |
| Maximum Speed | Between 50 and 100 (Inclusive) | 80 | Determine player speed movement in the play game menu |
| Music Status | Between “Play” and “Stop” | Play | Determine music status for play or stop the game background music |

* **The detailed key information** for the game setting:

|  |  |
| --- | --- |
| Key | Information |
| Up Arrow Key | * Move the cursor position up * When the current cursor position on the first game setting then move the cursor to the last game setting |
| Down Arrow Key | * Move the cursor position down * When the current cursor position on the last game setting then move the cursor to the first game setting |
| Left Arrow Key | * Decrease the game setting value * For music status game setting change the value between “Play” and “Stop” depend on the current state |
| Down Arrow Key | * Increase the game setting value * For music status game setting change the value between “Play” and “Stop” depend on the current state |
| ESC Key | * Save the current game setting and redirect to the main menu |

* If the users choose **menu 5** “**Exit**”, then:
* **Show** splash screen



**Figure 4. Exit splash screen**

* **Terminate** the program

**Referensi**

*References*

* “**nar.wav**” music song created by an assistant from 15-1 generation

**Please run the EXE file to see the sample program.**

**Komponen**

*Component*

|  |  |  |
| --- | --- | --- |
| **No.** | **Criteria** | **Percentage** |
| **1.** | **Play Song** | 3% |
| **2.** | **Read Map File** | 3% |
| **3.** | **Coloring Tile, Smooth Effect, and Pretty UI Interface** | 5% |
| **4.** | **Play Game** | |
| Game Logic | 10% |
| Shortest Path (Dijkstra or A\* Algorithm) | 45% |
| **5.** | **Build Map** | |
| Key Event | 2% |
| Fill or Remove Wall | 4% |
| Flood Fill Algorithm | 20% |
| Save Map | 2% |
| **6.** | **Game Setting** | |
| Key Event | 2% |
| Validation | 2% |
| Save Configuration | 2% |
| **Total** | | **100%** |