|  |  |
| --- | --- |
| **Project Case** | Description: LogoBINUS-University |
| C |
| **Periode Berlaku** Semester Ganjil 2022/2023  ***Valid on*** *Odd Year 2022/2023* | **Software Laboratory Center**  **Assistant Recruitment 23-1** |

***Note:******Please focus on the main logic and main feature!***

*(Splash screen and design are not scored)*

## Soal

*Case*

**C Space Invader**

Space Invaders is a very simple game by modern standards, but it was a technological marvel in its time. Space Invaders was a worldwide success and paved the way for a generation of shooting games that became extremely popular.You as a developer asked to make a Space Invader with C Programming Languange.

* **Main menu**
* This menu contains of 5 menus, which are **New Game**, **Load Game**, **Setting, How to Play, and Exit.**
* **Prompt** userto **input chosen menu**. **Validate** the input must **between 1 and 5 inclusively**.



Figure 1. Home Page

1. If user choose **New Game** (**Menu 1**), then:
   * **Prompt** user to **input username**. **Validate** the data must more than **3 characters exclusively**
   * Validate the **username** must **not exists** on **‘database/player.dat’.**



Figure 2. Input Username

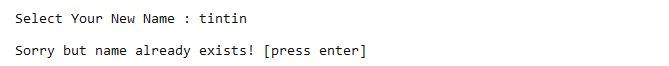


Figure 3. Username Validation

* + **Direct** user to **game page.**

1. If user choose **Load Game** **(Menu 2)**, then:
   * **Show** all available player at **‘player.dat’**
   * **Prompt** user to **input chosen menu**. **Validate** the input must between **1 and total player + back inclusively.**
   * If load data **success, direct** user to **game page**.
   * If user choose **back, direct** to **main menu**.



Figure 4. Load Data

1. If user choose **Setting (Menu 3)**, then:
   * **Show** all available **input**.

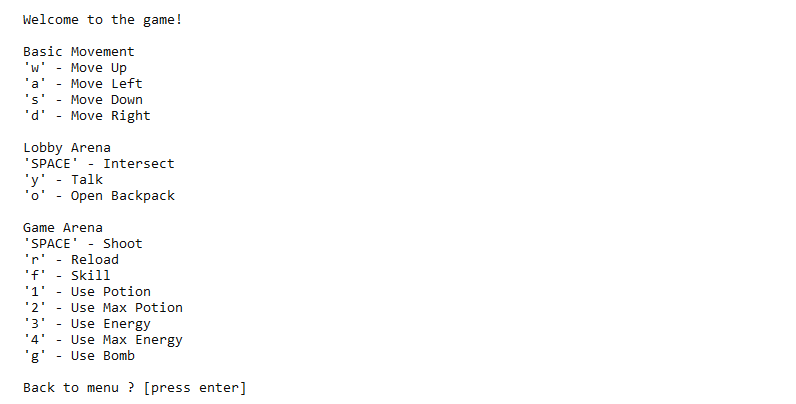


Figure 5. Setting

1. If user choose **How to Play (Menu 4)**, then:
   * **Show** main objective of the game.

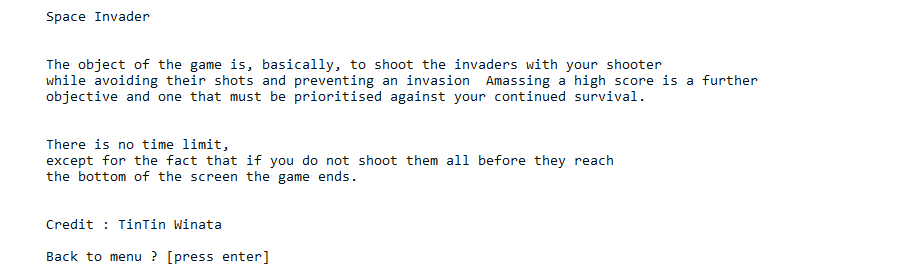


Figure 6. How to play

1. If user choose **Exit (Menu 5)**, then:
   * **Display** Logo and motto

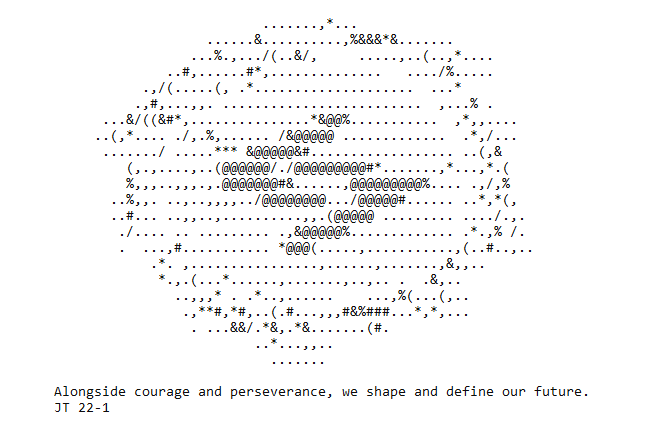


Figure 7. End Screen

* **Gameplay**
* **Lobby**
  + Spawn player at the **center** **of bottom of pillars**.
  + Display **lobby map, player position, player information**.
  + **Validate** if the **player level** is **less than 10**.
  + If the **player level** is **less than 10**, then **close** the **spaceship** door.
  + If the **player level** is more **than equal 10**, then **open** the **spaceship** door.
  + This is player information below :

|  |  |
| --- | --- |
| Player Attribute | Value |
| Default XP | 0 |
| Default Level | 0 |
| Default Money | 0 |
| Symbol | P |
| Default HP | 100 |
| Default Energy | 50 |
| Default Armor | 1 |
| Max HP | 300 |
| Max Energy | 500 |
| Max Armor | 30 |

Table 1. Player Information

* + Print **NPC** with the given symbol.
  + If user stands the **middle of lobby map** (fountain), then :
    - Tell **user** can do **interact** by **typing ‘SPACE’ key.**
    - **Player** can go to the **game arena**
* Below is some information of map symbols.

|  |  |
| --- | --- |
| Symbol | Meaning |
| | | Board |
| \ or / | Door |
| - | Closed Door |
| # or = | Wall |
| . | Pillar |
| P | Player |
| U | Upgrade Shop |
| I | Item Shop |
| W | Spaceship Station |

Table 2. Map information

* Prompt user to input **key W A S D** (Case Insensitive) to **move**.
* Validate that player **cannot pass through walls and pillars**.
* If user type key **O** (Case Insensitive), display player’s **backpack information**.

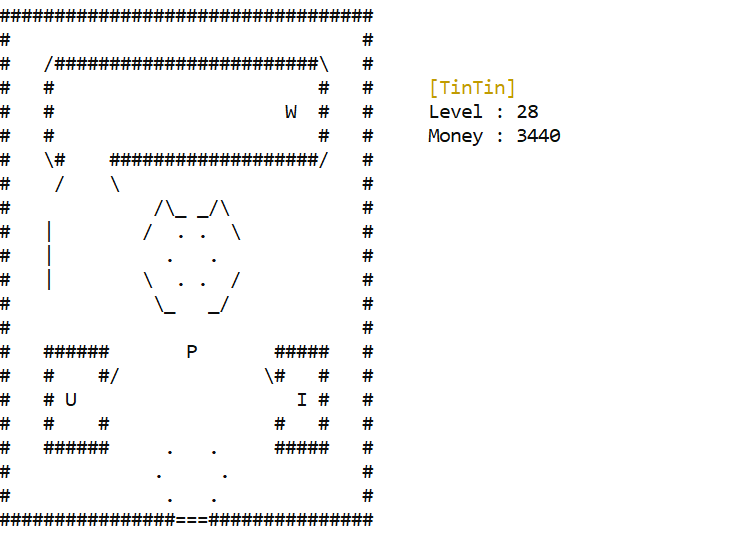


Figure 8. Lobby (Spaceship Door Open)

A picture containing text

Description automatically generated

Figure 9. Lobby (Spaceship Door Closed)

Background pattern

Description automatically generated with low confidence

Figure 9. Open Backpack

* + If users **stand in front** of **closed door**, then display a **message need to level 10**

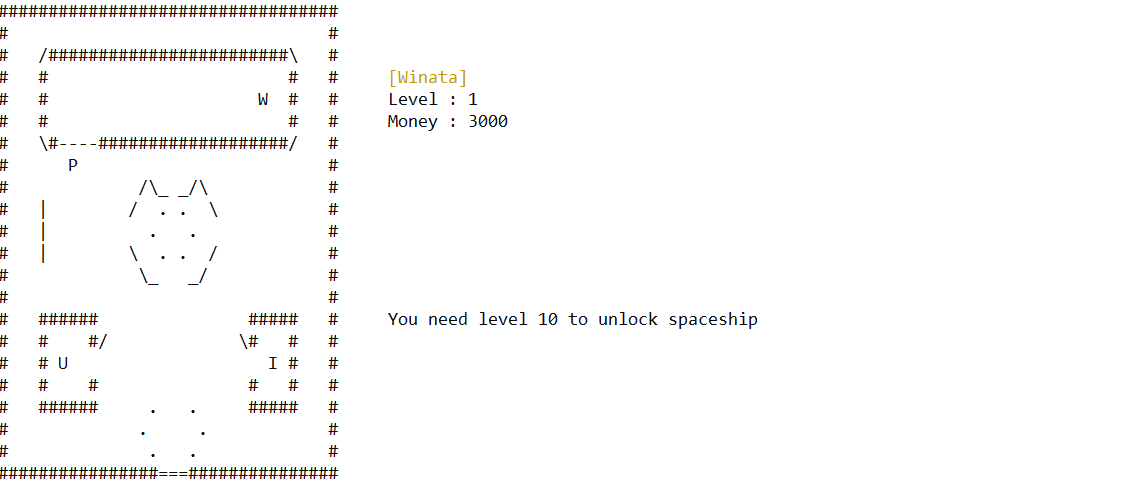


Figure 10. Closed Door Message

* + If users stand in the front of **Board**, then :
    - Tell **user** can do **interact** by **typing ‘SPACE’ key.**
    - If users input **‘SPACE’** then:
      * Open **‘database/score.dat’**
      * Read all the data in **score.dat** with this following format :

Format 1. Score.dat

playername#score

* + - * Sortthe **name and score** on score.dat **based on the score**, and **display** it on the prompt.
      * For **sorting algorithm** you can choose one of algorithm below :
        + Bubble Short
        + Quick Sort
        + Merge Sort
      * If user press **‘ENTER’** back to **lobby**.

A picture containing graphical user interface

Description automatically generated

Figure 11. Scoreboard

* If player position is at **U (upgrade Shop**), then:
  + Tell user can do interact by typing **‘SPACE’** key.
  + If user press **‘SPACE’**, then user will interact to **NPC**. **NPC** will ask user to go the shop menu.
  + If user press **‘y’**, then:
    - **Display** user **upgrade shop menu**.
    - **Prompt** user to **input chosen menu. Validate** the input must **between 1 and 4 inclusively**.
    - If user choose upgrades attribute **validate user money is enough and validate attribute is not at the maximum level**.
      * If **money is not enough** or **attribute is not at the maximum level**, **display error message**.
      * If **money is enough** and **attribute is not at the maximum** decrease the **money**,increment the **player attribute by 1**,and **display succesfull message**.
    - If user choose **4**, then **back to lobby**.
  + Remember please validate if you can’t press **‘y’** to open shop if you never interact with NPC.

Text

Description automatically generated

Figure 12. NPC will ask to go shop menu

A screenshot of a computer

Description automatically generated with low confidence

Figure 13. Upgrade shop menu

A picture containing graphical user interface

Description automatically generated

Figure 14. Successfully upgrade player attributes

A picture containing text

Description automatically generated

Figure 15. Ugrade shop invalid validation

* If player position is at **I (Item Shop)**, then:
  + Tell user can do interact by typing **‘SPACE’** key.
  + If user press **‘SPACE’**, then user will interact to **NPC**, **NPC** will ask user to go the shop menu.
  + If user press **‘y’**, then:
    - **Display** user **item shop menu**.
    - **Prompt** user to **input chosen menu. Validate** the input must **between 1 and 6 inclusively**.
    - If user choose items, then:
      * **Prompt** user to **input the quantity** to buy.
      * **Calculate total price** by **multiplication** of price and quantity. **Validate** user money is **enough to pay** the **total price**.
      * If user **money is enough**, **reduce user money** by **total priceadd items to** user’s data, and display **succesfull message**. Then **back to lobby**.
      * If money is **not enough**, display **error message.** Then **back to lobby**.
    - If user choose **6**, then **back to lobby**.
  + Remember please validate if you can’t press **‘y’** to open shop if you never interact with NPC.

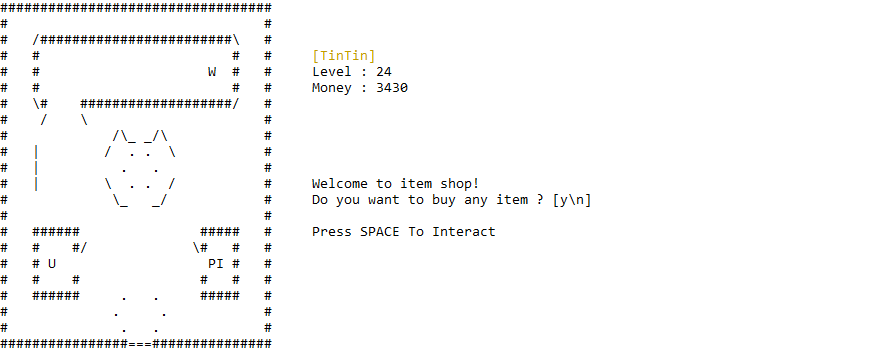


Figure 16. NPC will ask to go item shop menu

A screenshot of a computer

Description automatically generated with medium confidence

Figure 17. Item shop menu

Graphical user interface, text, application

Description automatically generated

Figure 18. Successfully purchase item.

Text, letter

Description automatically generated

Figure 19. Invalid validation item shop.

* If player position is at **W (Spaceship Station)**, then:
  + Tell user can do interact by typing **‘SPACE’** key.
  + If user press **‘SPACE’**, then user will interact to **NPC**, **NPC** will ask user to go the shop menu.
  + If user press **‘y’**, then:
    - **Display** current **spaceship**.
    - User can change current spaceship by input **‘A’ to (Left)** and **‘D’ to (Right)**.
    - Remember, there’s **no min or max** **spaceship** validation, if user input **‘D’** when **the current spaceship is last** then **back to first spaceship**.
    - User can **choose the spaceship** by input ‘ENTER’. And return to **lobby**.
  + Remember please validate if you can’t press **‘y’** to open shop if you never interact with **NPC**.

Background pattern

Description automatically generated with low confidence

Figure 20. Choose Spaceship

* If player position is at **bottom of the map** then:
  + Tell user can do interact by typing **‘SPACE’** key.
  + If user input **‘SPACE’** then :
    - **Open ‘database/player.dat’**
    - Below is the player format on **player.dat**.

Format 1. Player.dat format Player

name#money#xp#level#hp#energy#armor

* + - Validate if user exists in **player.dat**
    - If user **is exists** :
      * **Update the user on player.dat**
    - If user **is not exists** :
      * **Make a new user line in player.dat**
    - Display **main menu**.
* Below is some information about **Upgrade Shop**.

|  |  |
| --- | --- |
| Attribute Name | Price |
| HP | $50 |
| Energy | $50 |
| Armor | $30 |

Table 3. Upgrade Shop Information

* Below is some information about **Item** **Shop**.

|  |  |
| --- | --- |
| Item | Price |
| Potion | $5 |
| Max Potion | $20 |
| Energy Drink | $3 |
| Max Energy Drink | $15 |
| Bomb | $10 |

Table 4. Item Shop Information

* There are any **cheat codes** that’s run in the lobby**, to activate the cheat code**, user needs to **run buffers** below :

|  |  |
| --- | --- |
| Buffers | Cheat Code |
| W + A + S + D + W + A+ S + D + W + A + S + D + W + A + S + D | Bomb = 99 |
| A + S + D + A + S + D + A + S + D + A + S + D | All Item (Except bomb) = 99 |
| W + S + W + S + W + S + A + D + A + D + A + D | Player max attribute |
| D + A + S + D + A + S + W + D + A + S + W + D + A + S+ W | Player max level |

Table 5. Cheat Code

* **Shooter Arena**
  + **Display shooter information** (spaceship name, score, bullets, hp, energy).
  + Every 3seconds **Random 3 Enemies** with this following criteria :
    - Randomize **1 – 3** for the enemy level.
    - Every **4 seconds** enemy will **move** left - right **and** right - left **continously**.
    - Every **3 – 4 seconds** (Randomize) enemy will **shoot** a bullet.
  + Below is some information about **Enemy.**

|  |  |  |  |
| --- | --- | --- | --- |
| Level | Sprite | Damage | HP |
| 1 | \v/ | 10 | 1 |
| 2 | \x/ | 20 | 1 |
| 3 | \o/ | 25 | 1 |

Table 6. Enemy Information

* + Display player shooter with this criteria :
    - **Display sprite** based on their **referencee on Spaceship Station**. The default spaceship is ‘Default Spaceship’.
    - Below is some information about **shooter movement.**

|  |  |
| --- | --- |
| Key (Case Insensitive) | Shooter Event |
| R | Reload |
| W | Move Up |
| S | Move Down |
| D | Move Right |
| A | Move Left |
| SPACE | Shoot |
| F | Skill |
| 1 | Potion |
| 2 | Max Potion |
| 3 | Energy |
| 4 | Max Energy |
| G | bomb |

Table 7. Shooter Information

A picture containing graphical user interface

Description automatically generated

Figure 21. Shooter Arena

* + - Every **20 milliseconds**  increment **energy by 0.1**.
    - When user reloading, **reset the bullet to total bullet**.
    - When user **shoot**, then :
      * **Validate** if shoot **have any bullet left.**
      * If **there are any shoot left**, make **a new bullet with given symbol**.
      * **The direction of the bullet is always go up**.

Graphical user interface, application

Description automatically generated

Figure 22. Shooter Shoot

* + - When user skills, then :
      * **Validate** if **shooter energy is more than skill cost** and have minimum **3 bullets.**
      * If shooter energy is **more than skill cost** and **have 3 bullets** then **decrement energy** by skill cost, **decrement bullets** by **3 , spawn a three bullet** with 3 direction **(Up, Up-Left, Up-Right)**.

Graphical user interface, application

Description automatically generated

Figure 23. Shooter Skill

* + - **Remember**, bullets damage based on damage on the shooter.
    - When users use **bombs**, then :
      * **Fill map** with **bomb symbol.**
      * If there are any **enemy when filling the map**, then **enemy will die.**
      * Please use **floodfill algorithm** to fill the map.

A picture containing text

Description automatically generated

Figure 24. Before Bomb

A picture containing calendar

Description automatically generated

Figure 25. Bomb

A picture containing graphical user interface

Description automatically generated

Figure 26. After Bomb

* + - When **users use potion,** then **increment shooter hp by 50**, and decrement the potion by 1.
    - When **users use max potion**, then **set shooter hp to max**, and decrement by max potion 1.
    - When **users use energy drink**, then **increment energy by 50**, and decrement the energy drink by 1.
    - When **users use max energy drink**, then **increment energy to max**, and decrement the max energy drink 1.
    - Don’t forget to **validate**, if there’s **no item left**. User **can’t use the item** and **display error message**.

Graphical user interface, text

Description automatically generated

Figure 27. No item left error message.

* + - When use Item don’t forget to **display successfully used item**.

Graphical user interface

Description automatically generated with medium confidence

Figure 28. Succesfuly use item message.

* + - **Please remember** any changes on shooter while game is running, please **update the shooter status.**
    - Below is some **information** about shooter attributes :

|  |  |
| --- | --- |
| Shooter Attribute | Value |
| Reload Time | 1 Seconds |
| Skill Cost | 30 |
| Damage | 1 |
| Total Bullet | 10 |
| Bullet Symbol | ^ |
| Skill Symbol | o |
| Bomb Symbol | . |

Table 29. Shooter Information

* + **Render** all **player bullets** that available with this criteria:
    - **Validate** if player bullet **outside the map** then, **dissapper/remove** the bullet.
    - If player bullets **intersect with enemy**, then :
      * **Dissapper / remove** current bullet.
      * **Decrement enemy hp** by shooter damage.
      * If intersected enemy hp is **below than** 0, then enemy **will die and increment score by 10.**
  + **Render** all **enemy bullets** that available with this criteria:
    - **Validate** if enemy bullet **outside the map** then, **dissapper/remove** the bullet.
    - If enemy bullets **intersect with the shooter**, then :
      * **Dissapper/remove** current bullet.
      * **Decrement shooter hp** by enemy damage, if shooter hp is below than 0, then **stop the game and display Finish Game**.
* **Finish Game**
  + Display **total score** that earned.

Equation 1. Gained XP

Gained XP = Total Score \* 0.75

* + Open ‘**database/score.dat’** then :
    - If the name **isn’t exists** :
      * **Create new line** with **current player name and score**.
    - If the name **already exists** :
      * If the score in **‘score.dat’** is **less than current score**, then **update the score.**
  + Display **gained XP** that earned.
  + **Maximum XP on shooter is 100**, when XP is more than Maximum XP then :
    - Decrement **XP** by maximum XP
    - Increment 1 level to **player**.
  + Display **current level**.
  + If user input **‘ENTER’** then back to **lobby**.

Background pattern

Description automatically generated with low confidence

Figure 30. Finish Game Menu

* **Miscellaneous & Rules**
* Please **don’t make any changes** on folder **assets** or **moving out files from folder assets**.
* You can modify and change files on **database** folder.
* Save format for score.dat is **username#score**.
* Save format for player.dat is **name#money#xp#level#hp#energy#armor**

Please run the EXE file to see the sample program.