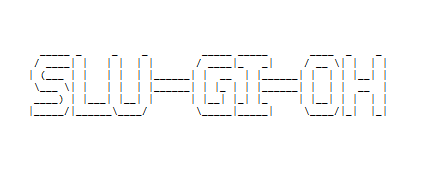
|  |  |
| --- | --- |
| **Mandatory Case** |  |
| Java2Mandatory |
| **Periode Berlaku** Semester Ganjil 2015/2016  ***Valid on*** *Odd Year 2015/2016* | **Software Laboratory Center**  **Assistant Recruitment 16-1** |

## Soal

*Case*

**SLU-GI-OH**

SLU-GI Company hire you to make a game for his company. The game is SLU-GI-OH which is almost identical with Yu-Gi-Oh. SLU-GI-OH hire you to make this game **using Java Programming Language** .





1. **Main Menu**

Program has 3 **menus**:

1. **Login**
2. **Register**
3. **Exit**
4. **Login**

* Program will ask **player** to input **username** and **password.** 
  + **If username** and **password** is “**admin**” in case insensitive. Then you will be logged in as **admin**.
  + Validate **username’s** length must not **less than 5** or **more than 40**. Otherwise display error message “**Username’s length must between 5 and 40**”.
  + Validate **username** must **contains 3** or more words. Otherwise display error message “**Username must be 3 words or more**”.

D:\pre-training-img\pre-training\Login\Username.PNG

* + Validate **password** must not **empty**. Otherwise display error message “**Password cannot be empty**”.
  + Validate **password**’s length must more than 8 characters. Otherwise display error message “**Password’s length must more than equals 8 characters**”.

D:\pre-training-img\pre-training\Login\pass.PNG

* + Validate **username** and **password must match** with **User**.**txt**.Otherwise display error message “**User not found**” and program will ask if you **want to continue login or not**. Validate input **must be “Y” or “N”.**
    - If input is “**Y**”, the program will ask player to **input** **username and** **password again**.
    - If input is “**N**”, the program will **back to main menu**.

**D:\pre-training-img\pre-training\Login\notfound.PNG**

* If player **logged in as** **admin**, the program will display **Admin Menu**.
* If player **logged in** **as user**, the program will display **User Menu.**

1. **Register**

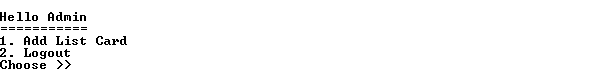
* First the program will **generate an id**. The format is “**SLxxx**” whereas xxx is **auto-incremented number**, it will **increase** every time when the user success **registered**.
* Program will ask user to **input username**:
  + - Validate **username** cannot be **empty**
    - Validate **username** must be **3 words or more**
  + Program will ask user to input **Email** :
    - Email must be **ended with .com**
    - Email cannot be **empty**
    - **‘.’** cannot be **beside ‘@’**.
  + Program will ask user to input **Password** :
    - Validate **password** cannot be **empty**
    - Validate **length of password** must be **8 characters or more**.
  + Set the **money** of player to **2000** bits.
  + **Generate a deck** for new **user** from “**Deck.txt**”
    - Validate the **maximum** of a deck when generating is **40 cards**.
    - A card cannot be **added twice** or **duplicated** card in a deck.
  + After that, write the user’s data to the file “**User.txt**” with format below :

|  |
| --- |
| **#Id#Name#Email#Password#Money** |

1. **Exit**

The program will be **closed automatically**.

1. **Admin Menu**



**Admin menu** contains 2 **menu**:

1. **Add List Card**
2. **Logout**
3. **Add List card**



* First program will ask to input **card type**.
  + **If card type is Monster :**
* Program will ask user to input **monster name. Validate the name cannot be empty.**
* Program will ask user to input **monster star. Validate the star must between 1 and 12.**
* Program will ask user to input **monster attack, monster defense and monster price.**
* **Generate the id** which is begin with **MN** then **followed by the total card + 1**. Example : total card is 80, then the id is MN081
* **After admin have been input all the required field**. The input will be **insert** into **Deck.txt** with format below :

|  |
| --- |
| **#id#name#star#attack#defense#price#** |

* + **If card type is Spell:**
* Program will ask user to input **spell name. Validate the name cannot be empty.**
* Program will ask user to input **spell description. Validate the description at least 5 words.**
* Program will ask user to input **spell damage and spell price.**
* Program will ask user to input **spell target.** Player can choose target to :
  + **Hero**
  + **Enemy**
  + **All Target**
* Then, program will ask again about **spell target**. Player can choose target to :
* **Monster**

If user choose **Hero** or **Enemy, and Monster spell target.** Program will ask again if the target is **single** or **multiple**

* **Life points**
* **Generate the id** which is begin with **MG then followed by the total card + 1**. Example: total card is 80, then the id is MG081.
* **After admin have been input all the required field**. The input will be **inserted** into **Deck.txt** with format below :

|  |
| --- |
| **#id#name#description#damage#price#target(allTarget/users/enemy)#target(Monster/lifepoints)#target(Multiple target / single target)** |

* + **Exit** 
    - Program will send you back to the **Admin menu.**

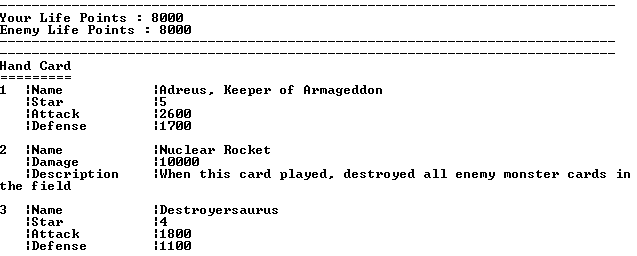
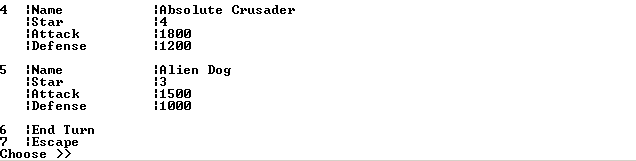
1. **Logout**
   * Program will send you back to **Main menu**
2. **User Menu**



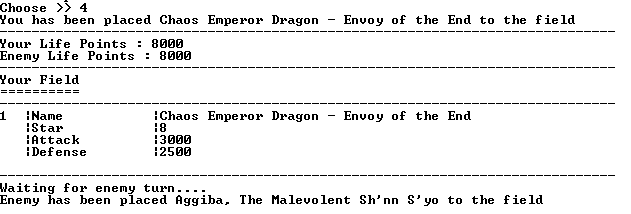
* Program Contains 6 menu :

1. **Play Game**
2. **My Card**
3. **Buy Card**
4. **Save**
5. **Refresh**
6. **Logout**
   * **Money** will be **increase** every 30 seconds by **200**.
     1. **Play Game**

* Program will **random the deck from list of card that user’s** have in **(userId).txt** Example: if user id is **SL003**, the name of the file is **SL003.txt.**
* **Enemy’s deck** will be **random from Deck.txt** which is only inputted by admin.
* Validate that **yours and enemy’s hand card cannot have more than 5 card**.
* At the first game, set **yours and enemy’s life points is 8000 pts**.
* Validate that **you** and **enemy cannot attack at the first turn**.
* When you or enemy **draw a card**, **reduce a card** from deck.
* Program will show **your life points**, **enemy life points, and your hand card** 
  + If the card type is **spell card**, **show name, damage, and description**
  + If the card type is **monster card**, **show name, star, attack, defense**

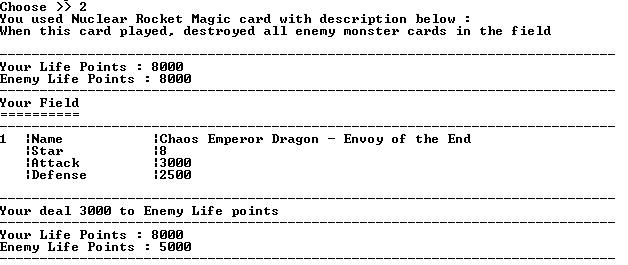
****

* Validate that you can only **choose one of the card**, **End Turn**, or **Escape**.
  + If user choose **end turn**:
    - If user **choose it at the first round**, it will be **the enemy’s turn**.
    - If user **choose it not at the first round**, user will move to **attack phase**.
  + If user choose **escape**, program will back to the **user menu**.
  + If user choose a **card**:
    - If the **chosen card type** is **monster**, then **placed it to the field**.
    - If the **chosen card type** is **magic**, it will **using its effect directly** :
      * **The card’s effect will affect all target, user or enemy.**
      * **The card’s effect will affect life points or monster**
      * **If the card will affect monster and the card’s description contains “destroyed” word, then destroy the monster**
      * **If the card’s description contains “increased” word, then increase monster’s defense / life points.**
      * **If the card’s description contains “reduced” word, then reduce the monster’s defense / life points.**

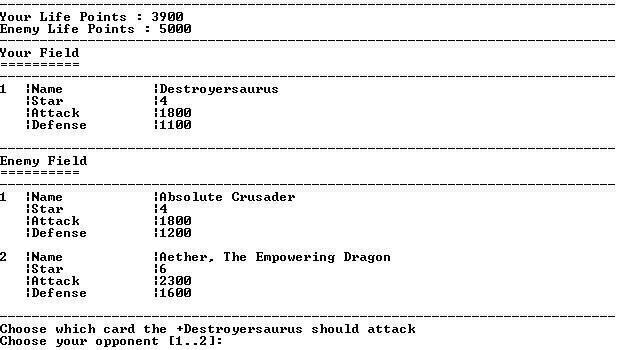
****

* If **enemy’s field does not have any monster card**, **your monster will attack to enemy’s life points directly then reduce the enemy’s life**.
* If **your field doesn’t have any monster card, enemy monster will attack to your life points directly then reduce your life points** with the formula below :

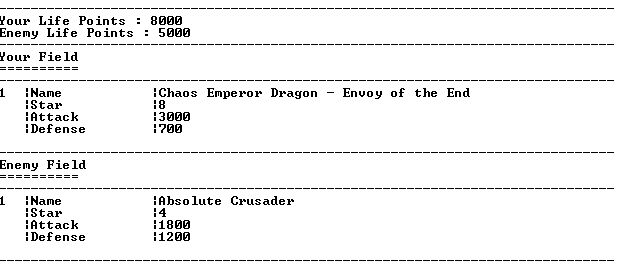
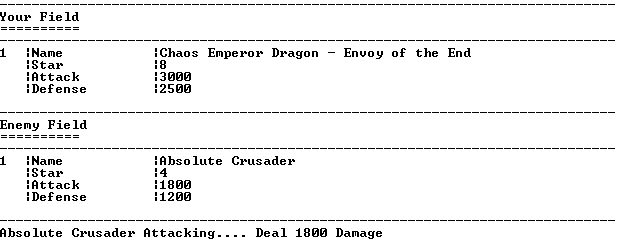
|  |
| --- |
| **Life points = Life points – Monster Attack** |

****

* If **enemy’s field have monster card**, the **first monster in user’s field will choose to attack one of the enemy’s monster card**.

****

* If **user’s field** have **monster card**, the **first monster in enemy’s field** will **attack** to **one of user’s card** which is **defense** **below the monster’s attack**. If there’s **no monster defense below its attack**, then **randomly choose one of them**.

****

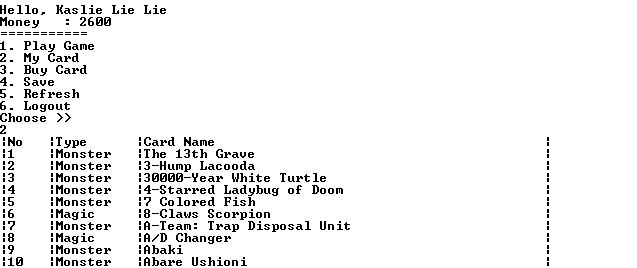
* If a **monster card attack**, reduce the **defense** with the formula below :

|  |
| --- |
| **Your Defense = your defense – enemy attack** |

* If a **monster card’s defense** is below **equals 0**, then **remove the cards from field**.
* If **user’s life points** **is** **below 0**, show the message “**You Lose**”, and then end the game.
* If **enemy’s life points below 0**, increase the money by 500, show the message “**You win, you get 500 bits**”.

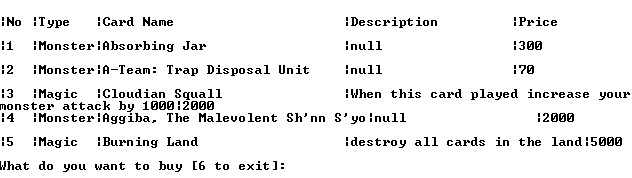
1. **My Card**

* The program will show all user’s cards.

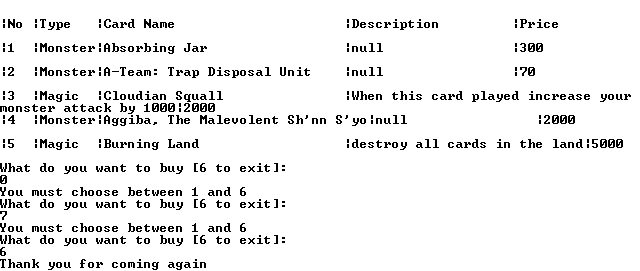


1. **Buy Card**

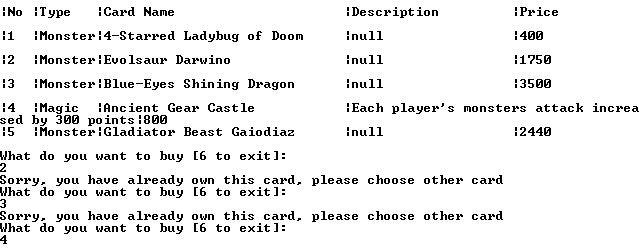
* The program will be show **5 cards random** from **“Deck.txt” every time user select this menu.**



* Validate **input** must **less than** 6.
* If **user input 6**, the program will exit to **user menu**.

****

* Validate if in the user’s deck **already has that card**, then **user can’t buy it anymore**.

****

1. **Save**

* **Player’s status** will be saved such as **list card, and money**.

****

1. **Refresh**

* Refreshing the **current game**, it will **affect user’s money**.

1. **Logout**

* The game will be ended, and back to the **Main Menu**.