Assignment 1 – CUI

Callum Geoffrey Clow 20119952 COMP603 Semester 1

Game rules

* First to drop enemy HP to 0 wins
* Take turns playing card
* Can use emperor action off CD as well
* Usually, must beat buildings before attacking emperor
* Rare chance for disaster event
  + Surviving a disaster with over half troops left will empower soldiers (+1/+1)

Average game

1. Welcome message
2. Enter username. Can enter “>X<” to quit
3. Display W/L and ask if they want to play the same emperor as last time
   1. If no, display emperor options and ask them to choose one
      1. They will get more information and a confirmation message
4. Shuffle cards
5. Play! Flip to see who goes first
6. Enter number to choose which card to play. Needed stats are displayed. Can enter 0 to quit.
7. End turn
8. Post game screen with stats
9. Play again?

Classes

* I/O class
* Card Class
  + Each individual card has its own class
* Game run class – final?
* Emperor class
* Deck class
* Board printer class
* Player class
* Stats class?
* Unit class
  + Extends Card class
* Soldier class
  + Extends Unit class
* Building class
  + Extends Unit class
* Equipment class
  + Extends Card class
* Random event class
* Hand class

Interfaces

* Emperor interface
  + Set data
    - Health 20 (changed by + 5HP emperor)
* Card interface
* Soldier interface?
* Building interface?
* Equipment interface?
* Pretty sure I don’t need hand interface or any other interfaces

I/O functionalities:

* Username
  + Linked to W/L ratio and last played emperor
  + Perhaps something to do with the cards?
  + Must use classes from lectures
  + Stats (average game time, # of moves, cards played etc.)

How Game class will work (how the game will work for the user

1. Runs startGame
   1. Makes player
   2. Exit
   3. View stats method
      1. Runs viewStats method
         1. Loads all time and user based stats from file
   4. Runs startMatch method
      1. Makes board
         1. toString
      2. Makes hand for player (maybe one for computer? - probably)
         1. Loads deck class and grabs 4 from deck to load into hand class
         2. toString
      3. Runs user take turn
         1. If deck not empty
            1. Runs select card
            2. Puts card on board
            3. Takes needed actions
            4. Replace card from deck
            5. Choose unit to use
         2. Else choose unit to use
      4. Runs computer take turn
         1. If deck not empty
            1. Runs select card
            2. Puts card on board
            3. Takes needed actions
            4. Replace card from deck
            5. Choose unit to use
         2. Else choose unit to use
      5. Runs disasterRoll
         1. Maybe enacts random disaster, probably not
      6. Eventually runs matchFinish
   5. Runs match end
      1. Match end message
      2. Saves user stats
      3. Asks if they want to view stats

Requirements met

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | E | M | A | Comments |
| CUI |  |  |  |  |
| I/O |  |  |  |  |
| Functionality |  |  |  |  |
| Design |  |  |  |  |

* Abstract – interfaces are abstract classes
* Encapsulation – private variables with getters and setters
* Inheritance – extended card and uni classes
* Polymorphism – playCard function will not place equipment on the board - ??? what this mean ??

TEXT FILES

Stats.txt

* Contains global stats

Users.txt

* Contains user name and last played emperor

Maybe more text files?

Final things to check

* Change all instances of “User” to “Player”
* Delete unused getters and setters ( a lot )
* Put getters and setters at the bottom of the document
* Tick off all hints / suggestions
* Comment each method and specific lines of code that need comments
* Is it a finished product?
* I/O 4+?
* Interfaces?
* Delete code not used
* Smelly?
* Rules from text file maybe?
* Convert to enums

CURRENTLY DOING

FIXING FULL BOARD PRINT