Group 4:
Troy Johnson
Saima Ishaq
Tyler W. Belair

SOFTWARE DESIGN FUNDAMENTALS - TERM PROJECT

PROJECT DESCRIPTION

The card game chosen is War, a replica of Bicycle, a similar card game. The key objective to War is to “win” all of the cards. This is done by dividing the deck evenly between the players (2 at the max) and placing all 26 cards face down (nuance doesn’t matter for coding practices), every turn each player draws 1 card and places it “face up” in front of them, the player with the highest card (ranked/unranked) wins both cards. After a turn is concluded the player that is victorious puts both cards at the bottom of their library, the game continues until either player has collected all 52 cards.

Card game chosen - War

A typical card game involving comparison’s when gameplay is active.

I.e. Game Start -> Player Login -> Player chooses how many other players are playing the game (if 1 is chosen, player will be playing against “AI”) -> Game Begins -> Deck is shuffled and divided in half -> Player turn -> Player draw -> Player plays 7 - Diamonds -> End Turn -> Opponents turn -> Opponent Draw -> Opponent plays 10 - Hearts -> End Turn -> Validation of which card is “larger” -> Opponent wins round -> Opponent collects both cards and they get placed at the bottom of library -> Game determines if win or loss condition has been achieved, if not -> take another turn

PROJECT SCOPE

The project scope will allow for game sustainability, creativity and unique game-interplay as well as enforce strict guidelines aimed at the highest standardised coding practices. To accomplish these goals the group has chosen the IDE IntelliJ, as it offers unparalleled access to libraries and utilities we find suitable to our project’s needs and desires. It is also free to use and easy to learn.

HIGH-LEVEL REQUIREMENTS:

1. Players will be able to “login” via the prompt screen and have access to a valid login system (SYSTEM IS WRITTEN ON RUN-TIME, USER INFORMATION IS NOT SAVED)
2. Stack Engine will computer win/loss con. Based on game conditional requirements (states)

3. The score will be kept via the player’s stack, the amount of cards in their library confer the amount of total points they have accumulated. As a reminder this is a accumulative game, so points have the capability to be both won and lost.

IMPLEMENTATION PLAN

Each developer will check the code at the end of their respective days to ensure code sustainability, to ensure that old/outdated code is not pushed on working code and to ensure that each developer has access, gains the ability to pursue options and coding paradigms as well as create a sustainable organization environment.

The master code will be kept in SDF_TERMPROJECT folder, while the UML, class diagrams, word documents stipulating pseudo-code, will all be stored in their own separate locations pending the naming conventions.

Please find current(not static) GitHub repository that will be utilized for our projects demands.

GITHUB REPO: https://github.com/17796-GROUP4/SDF_WarProject-TermProject

CODING STANDARDS

This project will utilize the IDE IntelliJ to operate, play, debug errors and create enticing interplay between player and game system. IntelliJ is chosen for its in-depth libraries, modules and accessible utilities as well as conforming to a strict encapsulated class-based program.

BASECODE WRITE-UP

Encapsulation - Each module of this game will include as many mutators and accessors as required to provide the most strict Data encapsulation.The player should not experience any in-game errors due to faulty coding, logical run-time errors or, compile errors.

Delegation - Each module of this game will also house 1-2 specific rule sets and no other module will be given a “say” in what each other module does. SImilarly the total function of each module will be sent to the “game stack” which will be separately controlled by a “stack engine” to determine the best course of action regarding game play, card play and rule modifications.

NOTE as it currently sits, the projects only base code is blank.

TODO:

Game hand init
Player init
Card Class Init
Repeatability

POTENTIAL PROBLEMS/TECHNICAL DIFFICULTIES

To my knowledge the entirety of this “game” will take place in a module called “STACK_ENGINE” this module will comprise the “turn base” that will be implemented in this game. The unfortunate part about this is that I am relying on a particular “clock” value during interplay and there may arise difficulties in executing this “clock” without errors.

Outside of this mechanic, I do not predict any truly difficult problem. (maybe netbeans won’t like us hahaha)