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. Player plays 6 -> Opponent plays 7 -> Opponent takes both cards and adds to their pile -> Repeat cycle until wincon

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.

GITHUB REPO: <https://github.com/17796-GROUP4/SDF_TERMPROJECT.git>

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. I would not want my player (or computer) to suddenly gain a wincon advantage due to faulty coding logic!

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.

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.