DeckDevs: Derivable 1

Divyadeep Maan

Anshul Alpesh Patel

Sane Sunny

Mohemmed Faaiz Shaikh

Sheridan College

Sivagama Srinivasan

SYST17796: Fundamentals of Software Design and Development

June 19th, 2023

Table of Content

Team Contract

SYST 17796 TEAM PROJECT

Team Name: DeckDevs

Please negotiate, sign, scan and include as the first page in your Deliverable 1.

Please note that if cheating is discovered in a group assignment each member will be charged with a cheating offense regardless of their involvement in the offense. Each member will receive the appropriate sanction based on their individual academic integrity history.

Please ensure that you understand the importance of academic honesty. Each member of the group is responsible to ensure the academic integrity of all of the submitted work, not just their own part. Placing your name on a submission indicates that you take responsibility for its content.

Team Member Names (Please Print)	Signatures	Student ID
Project Leader: Anshul Alpesh Patel	DocuSigned by: Anshul Patel 7AC986ED47C649A...	991710958
Divyadeep Maan	DocuSigned by: 5BBDC2FFC2A3425...	991705288
Sane Sunny	DocuSigned by: 2BB9750B8D3647E...	991710519
Mohemmed Faaiz Shaikh	DocuSigned by: Mohemmed Faaiz shaikh C9FC93A70B0D4F8...	991704798

For further information, read Academic Integrity Policy here :

https://caps.sheridancollege.ca/student-guide/academic-policies-and-procedures.aspx

By signing this contract, we acknowledge having read the Sheridan Academic Integrity Policy

Responsibilities of the Project Leader include:

- Assigning tasks to other team members, including self, in a fair and equitable manner.
- Ensuring work is completed with accuracy, completeness and timeliness.
- Planning for task completion to ensure timelines are met.
- Notifying the professor of any issues in a timely manner so that corrective measures can be taken.
- Any other duties as deemed necessary for project completion.

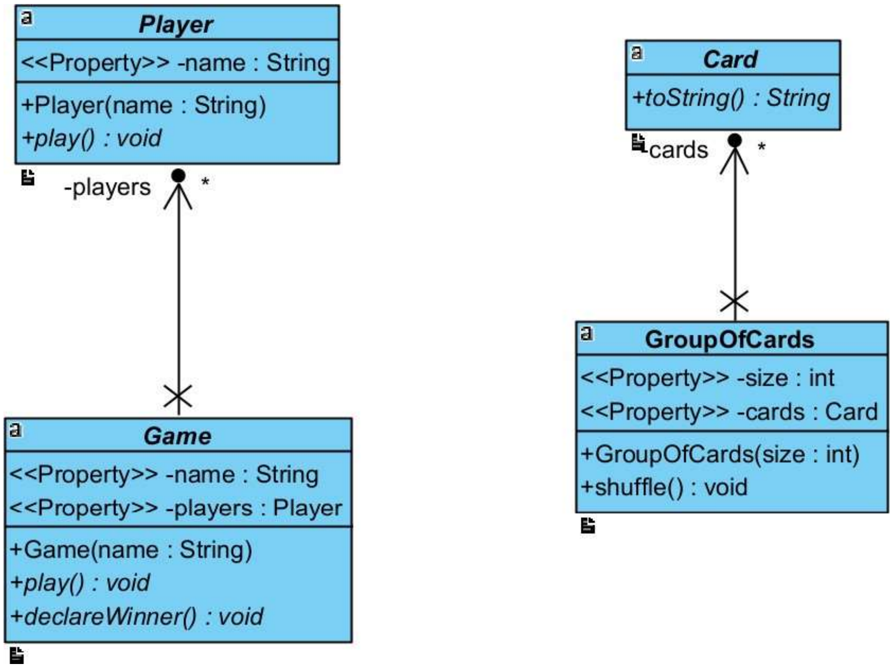
What we will do if . . .

Scenario	**Accepted initials**	**We agree to do the following (Put an X corresponding to your choice in each box)**
Team member does not regularly attend team meetings and/or does not respond to communications in a timely manner.	AP SS DM MS	Project leader emails the student citing the concerns and cc's the professor so they are aware of the situation at the very onset _X_ (**Mandatory**). a) Won't write their name in the submission of the document _X_
Team member does not deliver component on time due to severe illness or extreme personal problem.	AP SS DM MS	a) Team absorbs workload temporarily _X_ b) Team seeks advice from professor ___ c) Team shifts target date if possible ___ d) ___Other (specify):

Scenario	Accepted initials	We agree to do the following (Put an X corresponding to your choice in each box)
Team member has difficulty delivering component on time due to lack of understanding or ability.	AP SS DM MS	a) Team reassigns component ___ b) Team helps member _X_ c) Team member must ask professor for help __ d) ___Other (specify):
Team member does not deliver component on time due to lack of effort.	AP SS DM MS	a) Team absorbs workload ___ b) Team member(s) ask professor to request a Participation Form from all team members. This *may* result in individualized grades being awarded for a deliverable ___ c) Both a. and b. above _X_ d)___Other (specify):
Team cannot achieve consensus leaving one or more member(s) feeling that their voice(s) is/are not being heard in a decision which affects everyone.	AP SS DM MS	a) Team agrees to abide by majority vote _X_ b) Team seeks advice from the professor ___ c) ___Other (specify):
Team members do not share expectations for the quality of work on a particular deliverable.	AP SS DM MS	a) Team members will draw on each other's strengths to help bring the quality of the deliverable to a minimal acceptable level _X_ b) Team votes on each submission's quality ___ c) Team member(s) ask professor to request a Participation Form from all team members, which may result in individualized grades being awarded for a deliverable ___ d)___Other (specify):

Scenario	**Accepted initials**	**We agree to do the following** **(Put an X corresponding to your choice in each box)**
Team member behaves in an unprofessional manner, e.g. being rude, uncooperative and/or making one or more member(s) feel uncomfortable.	AP SS DM MS	a) Team agrees to avoid use of all vocabulary inappropriate to a business/college setting __ b) Team attempts to resolve the issue by airing the problem at a team meeting _X_ c) Team requests a meeting with the professor to discuss further ___ d) ___Other (specify):
There is a dominant team member who insists on making all decisions on the team's behalf leaving some team members feeling like subordinates rather than equal members	AP SS DM MS	a) Team will actively solicit consensus on all decisions which affect project direction by asking for each member's decision and vote __ b) Team will express subordination feelings and attempt to resolve issue _X_ c) Team seeks advice from the professor ___ d) ___Other (specify):
Team has a member who refuses to participate in decision making but complains to others that s/he wasn't consulted	AP SS DM MS	a) Team forces decision sharing by routinely voting on all issues _X_ b) Team routinely checks with each other about perceived roles ___ c) Team discusses the matter at team meeting __

UML Class Diagram



Design Document

Project Background and Description

The creation of a useful application for the UNO card game is the aim of this project. Players take turns matching cards in their hands with the top card on the discard pile in the well-known card game UNO. Up until one player has no cards left, the game continues. The UNO Rules website (https://www.unorules.com/) contains the UNO rules. The game's current beginning base code, which was built in Java, offers a simple framework for implementation.

The base code consists of the following classes:

Card Class:

- Abstract class representing a card.
- It has an abstract method toString() that needs to be implemented by the child classes.

Game Class:

- Abstract class representing a game.
- It contains the game name and an ArrayList of players.
- It provides methods to get and set the players, play the game, and declare a winner.
- It has an abstract method play() that needs to be implemented by the child classes.
- It has an abstract method declareWinner() that needs to be implemented by the child classes.

GroupOfCards Class:

- A class representing a group of cards in the game.
- It has an ArrayList of cards and a size attribute.
- It provides methods to get the cards, shuffle the cards, get the size, and set the size.

Player Class:

- Abstract class representing a player in the game.

- It has a name attribute.
- It provides methods to get and set the player name.
- It has an abstract method play() that needs to be implemented by the child classes.

Project Scope

The team members and their roles for this project are as follows:

- Divyadeep Maan: Creation Design Document
- Anshul Patel: Editing Design Document
- Sunny Sane: Creation of UML Class Diagram
- Mohemmeh Faaiz Shaikh: Handling Git Repository

The technical scope of the project includes:

- Include a player registration feature that enables each player to sign up for the game.
- Put the game's logic into action by matching cards, drawing cards, and announcing a winner by UNO regulations.
- Create a user-friendly interface that shows pertinent data including player scores, the cards currently in play, and game progress.
- Implement alerts when the game is won or lost.
- Make sure the game follows the UNO rules and gives the participants a fun gaming experience.

The project will be considered complete when the following high-level requirements are fulfilled:

- Players can register with the game.
- The game can communicate the outcome (win or loss).
- Players can view their status (score) during the game.

High Level Requirements

- Player registration: Make it possible for players to sign up for the game.

- Game logic: Apply the rules of the UNO game, such as matching cards, drawing cards, and deciding who wins.
- User interface: Create a simple user interface that shows player standings, the cards currently in play, and game progress.
- Alerts: Send players alerts when they win or lose a game.

Implementation Plan

GitHub:

- Git repository URL: https://github.com/pate5080/DeckDevs
- Expected use: Each developer checks in their code at the end of each day/week. Text files are stored under a separate directory, code, and UML diagrams have their own folders.

Coding standards:

- The team will follow a set of coding standards and conventions for consistency and maintainability.

Tools:

- Development Environment: NetBeans
- Version Control: Git

Design Considerations

Encapsulation

- Encapsulation is used in the "Game" class using private instance variables "name" and "players," which have a public getter and setter methods for accessing and changing them.
- The 'cards' and size private instance variables in the 'GroupOfCards' class are encapsulated, with getter and setter methods to manage access.

Delegation

- The 'play()' and 'declareWinner()' methods in the 'Game' class are both marked as abstract, indicating that the implementation would be left to the child classes.
- The 'play()' method in the 'Player' class is declared abstract, allowing other sorts of players to override and offer their implementation.

Flexibility/Maintainability

- The base code makes use of abstraction and inheritance to offer a flexible structure for implementing various game kinds.
- The 'Game' class serves as a model for certain game implementations, making it simple to extend and modify.
- The 'GroupOfCards' class offers a generic representation of a group of cards that may be used for various card games with variable card sizes.

The provided design considerations are based on the existing base code and can be further enhanced or modified based on the specific requirements and goals of the UNO game implementation.