SYST 17796 TEAM PROJECT

Team Name: Remote Coders

Please negotiate, sign, scan and include as the first section 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 honesty 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: Sonam Matharu	Sonam	991648370
Abhinav Khullar	Abhinav	991662951
Gurjeet Singh	Gurjeet	991666306

For further information read Academic Honesty Policy on AccessSheridan or visit the faculty office and speak with the Program Support Specialist.

By signing this contract, we acknowledge having read the Sheridan Academic Honesty Policy as per the link below.

https://policy.sheridanc.on.ca/dotNet/documents/?docid=917&mode=view

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
- Any other duties as deemed necessary for project completion

What we will do if . . .

Scenario	**Accepted** **Y/N** **+ initial**	**We agree to do the following**

Team member does not deliver component on time due to severe illness or extreme personal problem	A G S	a) Team absorbs workload temporarily Y b) Team seeks advice from professor __ c) Team shifts target date if possible __ d) Other:
Team member cannot deliver component on time due to lack of ability	A G S	a) Team reassigns component __ b) Team helps member Y c) Team member must ask professor for reference material __ d) Other:

Team member does not deliver component on time due to lack of effort	A G S	a) Team absorbs workload Y b) Team "fires" team member by not permitting his/her name on submission __ c) Other:

Scenario	**Accepted** **Y/N** **+ initial**	**We agree to do the following**
Team member does not attend team meeting	A G S	a) Team proceeds without him/her and will assign work to the absent member __ b) Team doesn't proceed and records team member's absence Y c) Team proceeds for that meeting but "fires" member after __ occurrences __

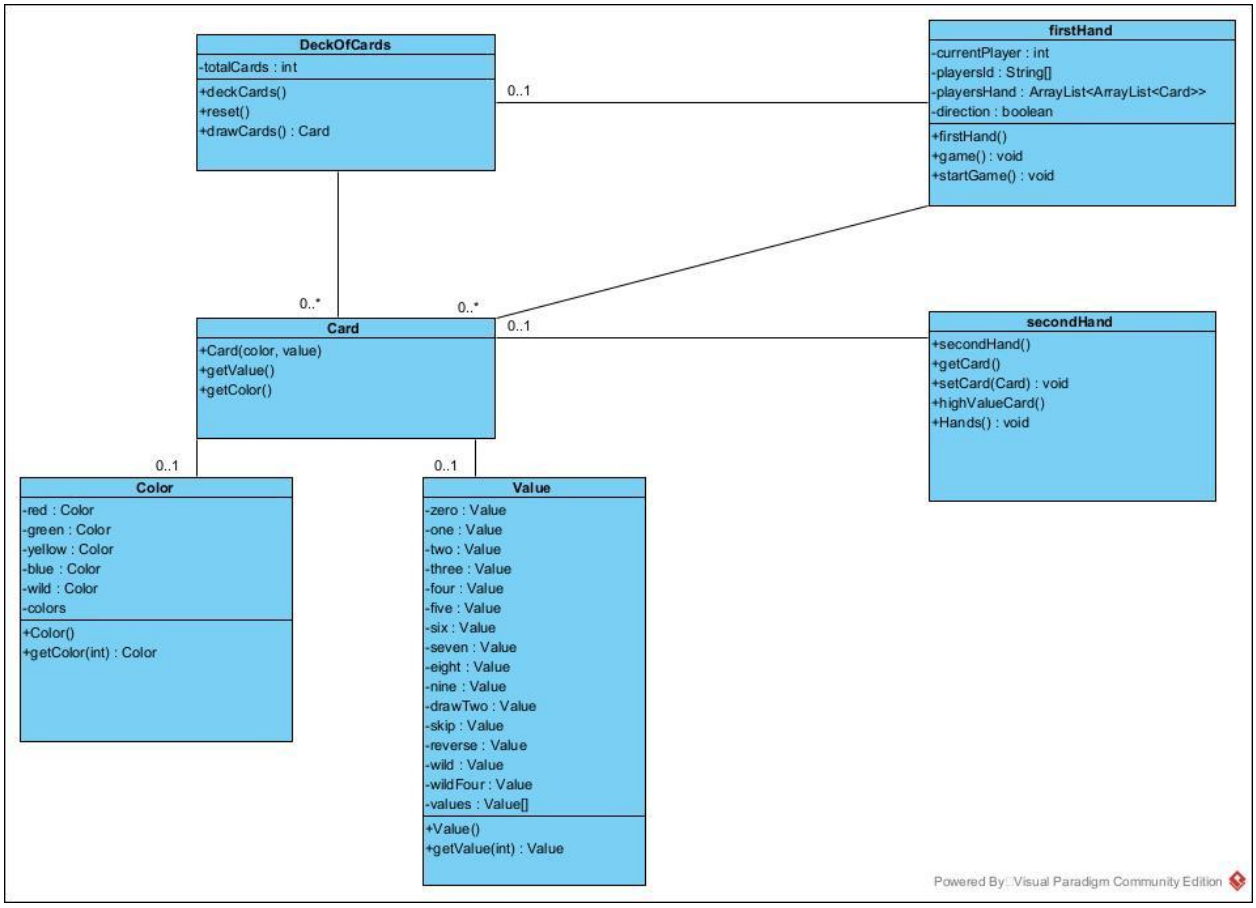
A piece of production equipment fails such as a printer, disk drive, or laptop	A G S	a) Backup copies will be made and kept in the college __ b) A locker or "share" directory will be used for joint access __ c) A photocopy and duplicate disk of all deliverables will be made Y d) Other:
An unforeseen constraint occurs after the deliverable has been allocated and scheduled (a surprise test or assignment)	A G S	a) Team meets and reschedules deliverable __ b) Team will cope with constraint Y c) Other:
Team cannot achieve consensus leaving one member feeling "railroaded", "ignored", or "frustrated" with a decision which affects all parties	A G S	a) Team agrees to abide by majority vote Y b) Team flips coin __ c) Other:

Team members do not share expectations for grade desired	A G S	a) Team will elect one person as "standards-bearer" who has the right to ask that work be redone __ b) Team votes on each submission's quality Y c) Team will ask for individual marking and will identify sections by author __ d) Other:

Scenario	**Accepted** **Y/N** **+ initial**	**We agree to do the following**
Team member behaves in an unprofessional manner by being rude or uncooperative	A G S	a) Team attempts to resolve the issue by airing the problem at team meeting Y b) Team requests meeting with professor to problem-solve __ c) Team ignores behaviour __

		d) Team agrees to avoid use of all vocabulary inappropriate to the business setting __
Team member assumes or requests that his/her name be signed to a submission but has not participated in production of the deliverable	A G S	a) Team agrees that this is cheating and is unethical __ b) Friends are friends and should help each other __ c) Team will submit with signature but will advise professor who will take action Y
There is a dominant team member who is content to make all decisions on the team's behalf leaving some team members feeling like subordinates rather than equal members	A G S	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 Y c) Other:
Team has a member who refuses to participate in decision making but complains to others that s/he wasn't consulted	A G S	a) Team forces decision sharing by routinely voting on all issues __ b) Team routinely checks with each other about perceived roles Y c) Team discusses the matter at team meeting __

UML DIAGRAM


DeckOfCards
-totalCards : int
+deckCards()
+reset()
+drawCards() : Card
0..1
firstHand
-currentPlayer : int
-playersId : String[]
-playersHand : ArrayList<ArrayList<Card>>
-direction : boolean
+firstHand()
+game() : void
+startGame() : void
0..*
0..*
0..1
Card
+Card(color, value)
+getValue()
+getColor()
secondHand
+secondHand()
+getCard()
+setCard(Card) : void
+highValueCard()
+Hands() : void
0..1
0..1
Color
-red : Color
-green : Color
-yellow : Color
-blue : Color
-wild : Color
-colors
+Color()
+getColor(int) : Color
Value
-zero : Value
-one : Value
-two : Value
-three : Value
-four : Value
-five : Value
-six : Value
-seven : Value
-eight : Value
-nine : Value
-drawTwo : Value
-skip : Value
-reverse : Value
-wild : Value
-wildFour : Value
-values : Value[]
+Value()
+getValue(int) : Value
Powered By: Visual Paradigm Community Edition

Design Document Template

Overview

1. Project Background and Description

UNO is a multi-player card game in which the goal is to be the first person to discard all of their cards.Each player is handed seven cards and takes turns drawing from the deck. A player can play a card that matches the card they drew from the deck if they have it in their hand (helping to reduce the total cards in their hand). If the player does not have a matching card in their hand, they must draw another card from the deck, increasing the total number of cards they must discard in order to win. When a player has only one card remaining in their hand, they must cry "UNO!" however if an opponent player discovers before the initial player can say "UNO!" they must draw four more cards. Code is written in java and made 6 different classes for all the parameters which is included in uno. All the classes are initialized with their constructor and all the attributes are included and stated as either private or public. Many different functions are being called which perform a particular game pattern.

2. Project Scope

There are 3 member in group in which Sonam Matharu is leading team. Sonam was working on guiding the team and distributing the work. He was more focused on the documentation part and came across to select uno for this because it is the most played game among people.

Abhinav was working on the UML Diagram by the discussion of all the attributes and functions which have to be used. He build the UML Class Diagram in Visual Paradigm.

Gurjeet was more focused on the coding part so he build and implemented the UNO program using Java programming language.

3. High-Level Requirements

Person will get a deck of cards and each card have two arguments that is color and value
When a person will done with all the cards he will won the game otherwise loss

As the number of cards are getting decreased as soon as the player is towards winning the game.

4. Implementation Plan

https://github.com/SonamMatharu/UnoGame

Used Eclipse for performing Java Program

5. Design Considerations

Class: We have build multiple classes and have connected them through a particuular multiplicity. There are about 5 classes present and all have their own attributes and functions.

Object: To call a function in a particular class we have used objects in that case.

Abstraction: This abstract modification is only applicable to classes and methods, not variables. It just gives you method prototypes, not implementations.

Polymorphism: One task can be done in many different ways that is to be termed as polymorphism.