

De Elit BRUINS

Project Deliverable 1

SYST17796 - Software Development
Guided under - Professor Amandeep Sindhu

Group Members: Steven (Group Leader), Jasmine
Daniel and Zawad

Content

Cover page	1
Content page	2
Group Contract	3-9
UML Diagram	10
Design Document	11-15

Team Contract

SYST 17796 TEAM PROJECT

Team Name:DE ELIT BRUINS.

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: Steven Kefila		991569077
Danylo Perepelitsyn		991566760
Jasmine Kaur		991540298
Steve ZAWAD HOSSAIN		991588646.

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

1/21/2020

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	Y Y ZH Y Y	a) Team absorbs workload temporarily <u>Y</u> b) Team seeks advice from professor — c) Team shifts target date if possible <u>Y</u> d) Other:
Team member cannot deliver component on time due to lack of ability	Y ZH PD JK	a) Team reassigns component __ b) Team helps member <u>Y</u> c) Team member must ask professor for reference material __ d) Other:
Team member does not deliver component		a) Team absorbs workload <u>Y</u>

<p>on time due to lack of effort</p>	<p>SK ZH PD JK</p>	<p>b) Team "fires" team member by not permitting his/her name on submission _</p> <p>c) Other:</p>
--------------------------------------	--------------------------------	----------------------------------------------------------------------------------------------------

Team Contract SYST 17796

Scenario	Accepted Y/N + initial	We agree to do the following
Team member does not attend team meeting	Sk ZH PD JK	a) Team proceeds without him/her and will assign work to the absent member <u>Y</u> b) Team doesn't proceed and records team member's absence __ 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	Sk ZH PD JK	a) Backup copies will be made and kept in the college __ b) A locker or "share" directory will be used for joint access <u>Y</u> c) A photocopy and duplicate disk of all deliverables will be made __ d) Other:
An unforeseen constraint occurs after the deliverable has been allocated and scheduled (a surprise test or assignment)	Sk ZH PD JK	a) Team meets and reschedules deliverable __ b) Team will cope with constraint <u>Y</u> c) Other:

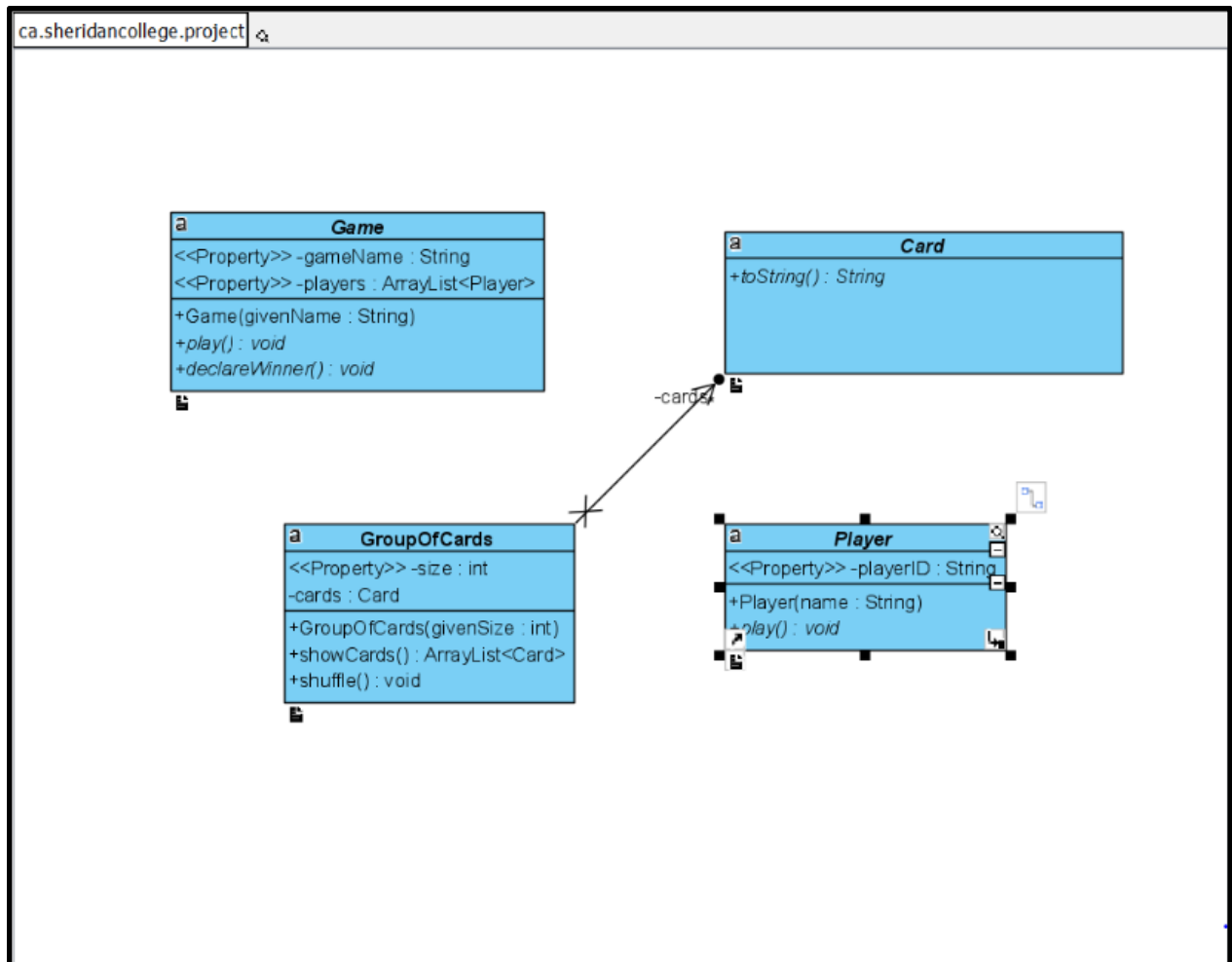
<p>Team cannot achieve consensus leaving one member feeling "railroaded", "ignored", or "frustrated" with a decision which affects all parties</p>	<p>SK ZH PD JK</p>	<p>a) Team agrees to abide by majority vote __ b) Team flips coin __ c) Other: <i>Consult the professor</i></p>
<p>Team members do not share expectations for grade desired</p>	<p>SK ZH PD JK</p>	<p>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 __ c) Team will ask for individual marking and will identify sections by author __ d) Other: <i>Consult the professor</i></p>

Scenario	Accepted Y/N + initial	We agree to do the following
Team member behaves in an unprofessional manner by being rude or uncooperative	SK ZH PD JK	a) Team attempts to resolve the issue by airing the problem at team meeting <u>Y</u> 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	SK ZH PD JK	a) Team agrees that this is cheating and is unethical <u>Y</u> b) Friends are friends and should help each other __ c) Team will submit with signature but will advise professor who will take action __
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	SK ZH PD JK	a) Team will actively solicit consensus on all decisions which affect project direction by asking for each member's decision and vote <u>Y</u> b) Team will express subordination feelings and attempt to resolve issue __ c) Other:
Team has a member who refuses to		a) Team forces decision sharing by

participate in decision making but complains to others that s/he wasn't consulted	SK ZH PD JK	<p>routinely voting on all issues __</p> <p>b) Team routinely checks with each other about perceived roles __</p> <p>c) Team discusses the matter at team meeting <input checked="" type="checkbox"/></p>
-----------------------------------------------------------------------------------	----------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Team Contract SYST 17796

UML Class Diagram for the starter code



1. Project Background and Description

(Describe the project goals and final vision. Include a brief description of how to play the game you have chosen and a reference to the rules of the game you have chosen.

1. Project Goals.

- a. The final goal of the project wasn't introduced to us yet, inasmuch as it's how our course is organized, but speculating of which, we could say that our game project needs to follow the next restrictions:
 - i. It should have all the game rules, and the game mechanics implemented.
 - ii. It should be intuitive and easy to use.
 - iii. We should add our new knowledge about the OOP and design patterns we've learned in class and from the course textbook so far.
- b. Also as a subgoal of completing this project, we will develop our soft skills which include:
 - i. Project management skills.
 - ii. Team organization skills.
 - iii. Communication and problem-solving skills.
 - iv. Project design and structure organization.
 - v. Product presentation.

2. Final Vision.

- a. As a final vision of our project, we need somehow to make our game work.
 - i. It means we need to decide whether it will be just a console game project (1 screen, 1 program ran, several players are waiting for their turns next to just a 1 computer, plus maybe some GUI interface), nor will we achieve something like online game, using network programming and parts of our game software on each machine our users play. Also, we could've used something like javascript for this goal, and I guess it would be easier than using java here, because we can create a site with hosting and so on, but we are still figuring out this question with our team.

The card game we chose for our project is called the UNO. The funny story of UNO started in 1971 Merle Robbins, invented the game while arguing with his son about another card game. Its a shedding type card game meaning whoever plays all the hand cards wins. This game can be played with 2 or more players. The deck has over 100 cards, each player starts with 5 or 7 cards depending on the number of players playing the game. All cards except wild cards have a color assigned to it. It could be blue, yellow, red or green. A player needs to match the color/ number on the card at the top of the pile of cards played. If the player

does not have a card to play and not even a wild card, then the player needs to draw the top card from the deck.

There are three types of cards in the deck

- 1) Number cards - Cards number starts from 0 to 9
- 2) Power cards - Each color has a set of power cards
 - a) Draw Two card - when played the next player draws 2 cards from the deck
 - b) Skip card - when played the next player will miss the turn
 - c) Reverse card - when played it changes the direction of the gameplay, from clockwise to anticlockwise vice versa.
- 3) Wild cards - these cards can be played anytime during the gameplay.
 - a) Wild Color card - It allows the player playing it to change the current color of the cards being played.
 - b) Wild Draw Four card - When played, the next player has to draw 4 cards from the deck.

When a player has played his/her second last card he/she has to say 'UNO'. And if the player fails to say 'UNO' he/she is penalized by drawing 6 cards from the deck.

If all the cards from the deck are exhausted and none of the players have played all their cards then the game ends as a tie.

2. Project scope

Our Group members are :

- 1) Steven, the group leader, who will be delegating the task of the project and owner of the Git Hub repository for the project.
- 2) Jasmine will be designing the layout and color scheme of the game environment.
- 3) Daniel is in charge of analyzing the starter code, how to implement it in the project.
- 4) Zawad will be recording the group meeting outcomes and documentation.

All of us will be working together to come up with algorithms that will form the structure of the backbone of our code and finally implement it. We will try to ensure our coding is efficiently written using the proper software in our case it will be the NetBeans 8.2. And for our team collaboration, we have created a Project repository.

A project is thought to be completed when it does whatever it is needed to achieve, so for us, it will be to run properly without any hiccups.

We are hoping that our final game application will be able to be launched smoothly on devices whichever supports Java Framework.

For a desktop version, the user(game player) will use the mouse to click on the desired card to make his/her move. For other devices (mobile /tablet) it will be touching the screen on appropriate locations.

3. High-Level requirements

1. The ability for each player to register with the game
 - a. The registration process will be as follows;
 - i. Each player will sign up with a random name of their choice.
 - ii. Cards will be shuffled and each player will receive their number of cards.
2. The ability for the game to communicate a win or loss
 - a. The first player to finish their cards wins the game but before they can be declared the winner, he/she has to click the finish button. If this player fails to do so, then the game won't stop and another person might win the game.
 - b. When someone wins, the game will display "[Name of the winner] has won the game".
 - c. For the losers, the game will display "The losers are [Names of the losers]".
3. The ability for players to know their status at all times
 - a. During the gameplay, when each player plays a card, their card hand will show the current number of cards they have.
 - b. Players will also be able to see the number of cards their adversaries have.
4. The ability for a player to draw a card randomly from the deck
 - a. During the gameplay, players might run out of cards to play. They would be able to draw cards randomly from the top of the deck.
5. The ability for each player to have their turn to play
 - a. A player won't be able to play when it is not their turn. Our code will display a message to show who's turn it is to play. For example: "Steven's turn to play".
 - b. A player won't be able to play if he/she doesn't have the required card to play. He/she must pick a card from the pile of cards.
6. Tie Game!
 - a. If all the cards from the pile of cards are exhausted and no one has finished playing, the game will display "Oops! It's a tie game. Would you like to restart the game?".

4. Implementation plan

1. Each member of the team has their own responsibilities for this project.
 - a. Each member of the team has their own responsibilities for this project. As was described above we have set our roles at the moment, and by completing them, and moving forward we will face new challenges, and with the time we will rearrange our roles respectively to the problems we face.
2. Project communication.
 - a. In order to share our work, we will be using the GitHub repository, so every member of our team can observe all the changes in the project remotely. (Github repository link <https://github.com/skebila/Deliverable-1.git>). Where we will be storing files for the project like code, text and other files, documentation, UML diagrams, e.t.c, separated by their own directories.
3. Personal communication.
 - a. In order to communicate we usually leave sometime after the class to discuss the project, or we can freely use WhatsUp or Discord where we have our team group chat in order to navigate through the project or asking each other questions related to the project or to get help from our team members.
4. Coding standards
 - a. For our project standardization, we were recommended to use the same IDE for the project, NetBeans of v.8.0.2 or higher and Visual Paradigm software for creating UML graphical representation of the project code structure.

5. Design Considerations

Currently, our code is a structure of abstract classes and methods, which means we don't have the implementation yet. But all these methods already represent the structure of our game where there are represented following dependant classes:

- (Card) class: class of cards which represents a card, and has Enums for its suit and value.
- (GroupOfCard) class: a group of (Cards) shown above ^, that has a |LinkedList| field for storing these cards, method to create all the cards for a deck, and shuffle method (which I guess should be pseudo-randomized), and overall this class is used to create a deck and pass it to the game.

- (Player) class: a player with a unique ID in the game, his own hand, and methods representing basic game mechanics like drawing cards and playing them, and of course he needs to know where are his turns, and when aren't and he should wait for everyone else to finish their turns.
- (Game) class: has a |ArrayList| of players who are taking part in the current game, and in the future should implement, or delegate all the remaining game mechanics.

Overall our game uses inheritance for its classes which should be implemented in real game classes later, will be using polymorphism if we will leave this structure of abstracts when we've done to override some methods, or overload them for some cases if we'll need to do so. All it's classes delegate their responsibilities, which means one class represents and encapsulates one functionality and represents one entity. We don't have our code done so that's it for now.

References

www.unorules.com