**Game Rules**

The game chosen for the project is Durak, a traditional Russian card game.  
In this game the goal is to be the first player without cards. This will result in the opposite player to be the “Durak” or “fool” as per the meaning of it.

The application to be created will be a 2-player game where the other player can and cannot be Computer AI.

A standard 52 deck card is used but 2 to 5 cards are removed, leaving 36 cards.

The ranking of the cards in the trump suit changes and the highest trump card is Ace.

Each player will receive 6 cards and the bottom card of the deck is turned over to reveal the trump suit for the hand.

The game is turn based and when one player is attacking the other will be defending and vice versa.

The attacking player selects a card and shows it on the table.

The defending player is supposed to put forward the same suit if possible or else can play any card.

If the attack is not defended then the defender must pick the attacking cards.

If it is defended then the defender can use a higher card of the same suit or the trump card to defend the attack, upon success the attacking cards remain on the table.

After every attack and defence, the players will draw new cards from the deck until they have 6 cards.

The turn passes to next player and becomes the attacker.

The game continues until one player gets rid of all cards, this player is the winner and the opponent losses being the “Durak”.  
  
**Computer AI Logic**  
  
The AI player should be setup in such a way that it takes care of the following cases when playing:

* Based on the cards in the hand of the player
* Based on the cards played by the opponent
* The trump suit card

When attacking the AI player should select a card based on the value, suit and some strategies defined.

The AI player should also be able determine the strength or order of the card played by the attacking player and must choose the defence card based on current suit and trump suit.

**Requirement Specifications**

Must Have

* Game logic with attacking and defence mechanisms
* Graphical User Interface (GUI) for the game
* Computer AI capable of attacking and defending
* Deck management which includes shuffling and dealing of cards

Should Have

* Game option to select difficulty
* Sound effects for the game events and also animations for better experience
* From the game data show player analytics
* User guide

Could Have

* Multiplayer mode both online and offline
* Advanced AI for the computer player which has adaptive learning
* Include theme options to change GUI

Will Not Have

* Social Media Integration for authentication of sharing purposes
* Step by Step tutorial intro for the game
* No store to buy in game related themes or any resources whatsoever
* No AR (Augmented Reality) or VR (Virtual Reality) experience
* No Support for multiple languages

**Class Diagram**

The classes or entities for the game project are as follows:

* Player – represents a player in a game and has the attributes like name and methods like playCard() etc.
* Card – represents a card in a deck and contains properties like suit and value
* Deck– represents the deck of cars used and contains functions like shuffle(), deal() etc.
* Game – represents the game logic and state, this class will have functions like start(), end() etc.
* AI Player, Human – represents the types of players in the game and might have specific logic to play

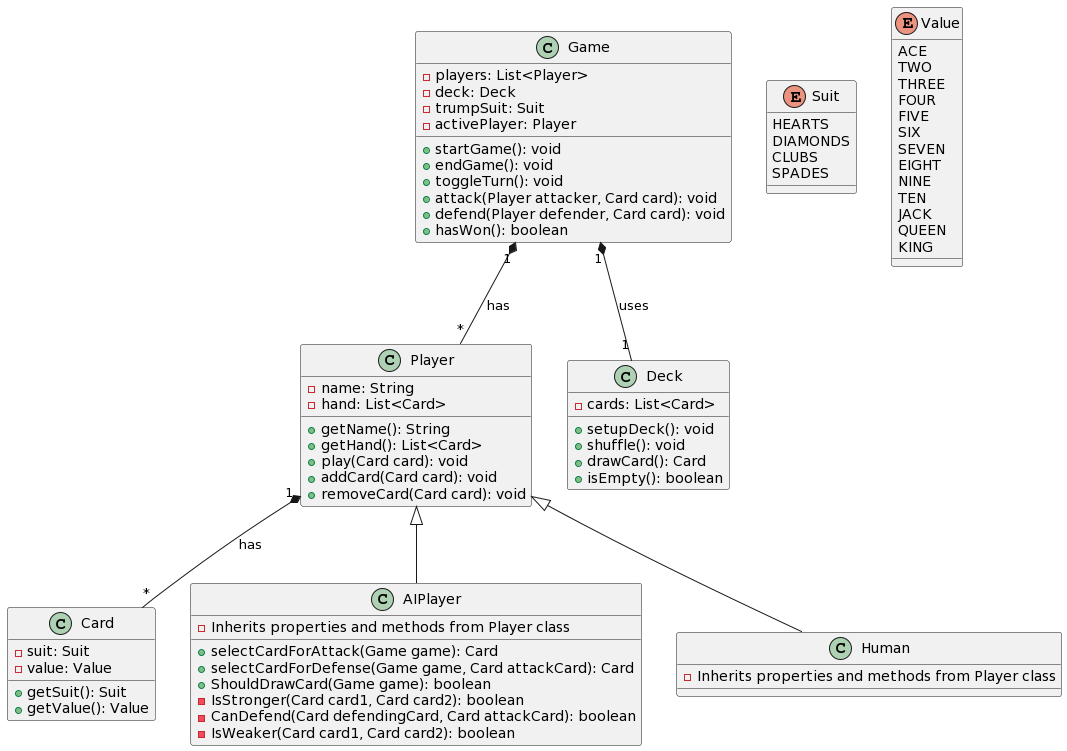
Relationships

* Player has a hand of cards – Composition
* Game has players – Association
* Deck contains cards – Aggregation
* AI Player and HumanP layer extends Player – Inheritance

Enumerations

* Suit – represents the suits in a deck like hearts, clubs etc.
* Value – to represent the value of the cards in the deck like K, Q, 2 etc.

Below is the class diagram



**Sequence Diagram**

The Player’s action is to attack or defend

Once the game starts the deck is initialised and cards are drawn from deck for both players

The game has AI player which is set to act or play accordingly

Below is the sequence diagram

