

# 1. Refined requirements specification - team description of What the game is not How it will be developed (Max 2 pages)

Whist is an Internet-based game. Whist can be played with teams or individually. If playing in teams of 2, partners are seated across from each other. Players playing on their own

The dealer; a player that gives out cards to each player. The cards given are faced down. Each player will have thirteen cards. The final card, belonging to the dealer, will be revealed to every player. The revealed card determines the trump of the game. A trump card is a suit that outranks all other suits.

Each player selects a card from their hand and reveals it on the table to everyone.

To determine the winner of the round:

If the same suit is played, the rank of the card determines the winner of the round

If only one trump card is played, then that card outranks the other cards.

If there are two trump cards played, the trump card's rank will decide the winner of the round

Winner of the round gets a stack of four cards, played by the players of that round. It is also known as a trick.

The game ends when all the cards have been played or the person/team gets more than 6 tricks.

Players can play any card.

(If playing in teams of 2, partners must sit opposite each other.)

The suit rank from weakest to strongest:

2 3 4 5 6 7 8 9 10 J Q K A

Default setting:

Individual

To determine the winner of the game, the first player gets six tricks wins the game.

**Variations:**

## Partners

To determine the winners of a game, the team with the highest number of tricks combined wins the game.

## Artificial Intelligence

Placeholder for a player will have similar behaviour to what a real player would play.

## 2. Scenarios - First description of playing game (Informal - text) (max 2 pages)

Informal Scenarios:

Current system state:

The System State consists of 4 players each at their starting locations, Players are situated at opposite ends of each other that will correspond with Player1, Player2, Player3, Player4

In a 4 player game, it will be played in teams of 2, in this game it is agreed that Dealer and Player2 in one team and Player3 and Player4 is one team.

They will receive a hand of 13 cards by the dealer player one card at a time starting from the dealer's left side.

The last card dealt will be revealed to all players and its suit will become the trump suit for this trick.

After all the players acknowledge the trump suit, the card will remain revealed until it's the dealers turn, in which case the card goes into the dealer's hand

The first trick will be initiated by the player on the dealer's left

Informal Scenario 1, Beginning of game:

Trump card was revealed to be the ace of spades.

Player2 plays the 9 of hearts which was Player2's highest card in hearts, Player3 plays 8 of hearts which was not Player3's highest valued card,

Player4 plays the 2 of hearts which is the loPlayer2 value hearts card Player4 has, and the Dealer takes the revealed card into his/her hands and plays the ace of hearts the highest valued card in the suit, winning the trick,

Dealer gather's the trick (which is 9 of hearts, 8 of hearts, 2 of hearts and ace of hearts), the dealer receives one point towards his/her score

And the next trick will be first played by the winning of the last trick, in this case, the dealer, but just before that Player3 has requested to see the last trick, and the dealer will reveal the last trick to Player3, and start the next round.

Informal Scenario 2, Mid game

After a few tricks, Player3 is in the lead.

This trick starts with Player4 since Player4 won the last trick,

Player4 plays the king of Diamond, Dealer plays the 2 of Diamond since The Dealer has no other spades left, Player2 plays 5 of Diamond, and Player3 plays the Queen of diamonds, Player4 won this trick because the king of diamonds is the highest-ranking card and takes the trick.

Player4 will start the next trick.

Informal Scenario 3, Late game

Nearing the end of the game, all players only have a few cards left.

This trick starts with the Dealer, Dealer plays the 2 of spades,

Player2 discard a card since Player2 have no spades left in hand and spades is the trump,

Player3 plays the Ace of Spades, the highest card in the trick, and wins the trick.

Informal Scenario 4, End game

After all the players have no cards left, the team with tricks more than 6 wins the round.

Informal Scenario Disconnections

When a player disconnects from the game, the game will display a timer to show the time the game will pause to allow the player to reconnect. If the player rejoins the game resumes, if the player fails to join back to the game, the game ends.

### 3. Primary class list (tends to be domain-based) (Max 10 plus or minus 3)

- Game
- Player
- Server
- Table
- Trick
- Card
- AI
- Deck
- Team

### 3a. Class Responsibility Collaboration Descriptions of your Classes (max 10)

<b>Class Name:</b> Game	<b>ID:</b> 1	<b>Type:</b>
<b>Description:</b> Representation of the game itself. It keeps track of the players and keeps the game running.		<b>Associated Use Cases:</b> Player joins game session. Player/AI assigned to team. Team wins game. Team loses game.
<b>Responsibilities</b>	<b>Collaborators</b>	
Keeps track of the players in the game	Player	
Keeps the game running	Server	
Initialises the round		
Initialises the cards	Card	
Initialises the table	Table	
Displays player menu		
Updates the score		
Starts game	Server	
Ends game	Server	
Displays trick	Trick	
Displays trump	Card	
<b>Attributes</b>		
Players		
NumPlayers		
GameInProgress		
Dealer		
Scores		
CurrTrick		
CurrTrump		
Deck		
<b>Relationships</b>		
<b>Generalisation (a-kind-of)</b>		

<b>Aggregation (has-parts)</b>	
Player	
Table	
Deck	
<b>Other Associations</b>	<b>Collaborators</b>
“Initialises”	Card
“Contains”	Trick

<b>Class Name:</b> Player	<b>ID:</b> 2	<b>Type:</b>
<b>Description:</b> Represents a user playing the game.		<b>Associated Use Cases:</b> Player joins game. Player shares link. Player assigned to team. Player deals cards. Player plays card. Player leaves game.
<b>Responsibilities</b>	<b>Collaborators</b>	
Tracks if player is dealer		
Player can join game	Game, Server	
Player can quit game	Game, Server	
Player can get cards	Deck, Card	
Player makes a move		
Tracks players cards	Deck, Card	
<b>Attributes</b>		
Name		
ID		
IsDealer		
Hand		
<b>Relationships</b>		

<b>Generalisation (a-kind-of)</b>	
<b>Aggregation (has-parts)</b>	
Card	
<b>Other Associations</b>	
“deals”	Card
“takes”	Trick
“is on”	Table

<b>Class Name:</b> Server	<b>ID:</b> 3	<b>Type:</b>
<b>Description:</b> Keeps the game running and allows players to join game.		<b>Associated Use Cases:</b> Server starts game session. Server generates share link. Player shares link. Server starts game. Server updates game session. Server ends game session.
<b>Responsibilities</b>	<b>Collaborators</b>	
Starts game connection	Game	
Ends game connection	Game	
Allows players to join	Player	
Allows players to leave	Player	
Generate share link		
<b>Attributes</b>		
ListUsers		



WaitingList	
<b>Relationships</b>	
<b>Generalisation (a-kind-of)</b>	
<b>Aggregation (has-parts)</b>	
<b>Other Associations</b>	
“connects”	Player

<b>Class Name:</b> Table	<b>ID:</b> 4	<b>Type:</b>
<b>Description:</b> Displays information about the game.		<b>Associated Use Cases:</b> Player displayed on table. Player places card on table. Card displayed on table.
<b>Responsibilities</b>	<b>Collaborators</b>	
Maintains information about Cards and Players on table	Card, Player	
Maintains information about table dimensions		
<b>Attributes</b>		
Size		
Colour		
Cards		
<b>Relationships</b>		
<b>Generalisation (a-kind-of)</b>		

<b>Aggregation (has-parts)</b>	
Trick	
Card	
<b>Other Associations</b>	

<b>Class Name:</b> Trick	<b>ID:</b> 5	<b>Type:</b>
<b>Description:</b> Contains information about the cards in trick.		<b>Associated Use Cases:</b> Card added to trick. Player wins trick. Player takes trick.
<b>Responsibilities</b>	<b>Collaborators</b>	
Stores information of each card	Card	
Compares each card	Card	
Winning player takes trick	Player	
Contains winning card in trick	Card	
<b>Attributes</b>		
CardCount		
WinningCard		
Trick		
<b>Relationships</b>		
<b>Generalisation (a-kind-of)</b>		
<b>Aggregation (has-parts)</b>		
Card		
<b>Other Associations</b>		

"won by"	Player
----------	--------

<b>Class Name:</b> Card	<b>ID:</b> 6	<b>Type:</b>
<b>Description:</b> Represents a card in a game and can be Spades, Clubs, Diamonds, or Hearts. Card also has a rank.		<b>Associated Use Cases:</b> Card shuffled in deck. Player deals cards. Card marked as trump. Player plays card. Card added to trick.
<b>Responsibilities</b>		<b>Collaborators</b>
Maintains information about card		
Ranking based on Rank and Suit		
Card can be a trump card		
<b>Attributes</b>		
Rank		
Suit		
Type		
<b>Relationships</b>		
<b>Generalisation (a-kind-of)</b>		
<b>Aggregation (has-parts)</b>		
Player		
Trick		
<b>Other Associations</b>		
"dealt from"		Deck
"displayed on"		Table

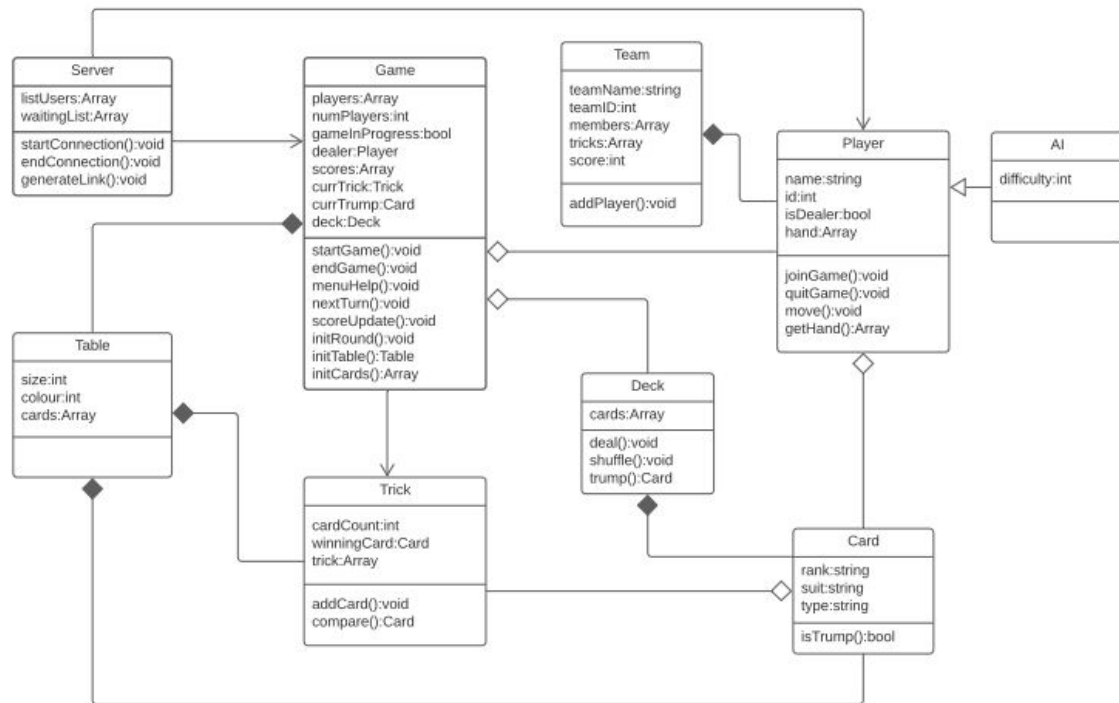
<b>Class Name:</b> AI	<b>ID:</b> 7	<b>Type:</b>
<b>Description:</b> Represents a fake player if real player leaves game. AI plays based on difficulty level set.		<b>Associated Use Cases:</b> AI joins game. Player sets AI difficulty. AI assigned to team. AI deals cards. AI plays card.
<b>Responsibilities</b>	<b>Collaborators</b>	
Maintains information like a real Player	Player	
Plays based on difficulty level		
Tracks if AI is dealer		
AI makes a move		
Player can get cards	Deck, Card	
Tracks AI cards	Deck, Card	
<b>Attributes</b>		
Difficulty		
<b>Relationships</b>		
<b>Generalisation (a-kind-of)</b>		
Player		
<b>Aggregation (has-parts)</b>		
<b>Other Associations</b>		

<b>Class Name:</b> Deck	<b>ID:</b> 8	<b>Type:</b>
<b>Description:</b> Maintains information about all 52 cards in the game.		<b>Associated Use Cases:</b> Deck shuffles cards. Player deals cards. Card marked as trump.
<b>Responsibilities</b>	<b>Collaborators</b>	
Stores each card	Card	
Shuffles cards	Card	
Cards dealt from deck	Card, Player	
Trump card revealed from deck	Card	
<b>Attributes</b>		
Cards		
<b>Relationships</b>		
<b>Generalisation (a-kind-of)</b>		
<b>Aggregation (has-parts)</b>		
Game		
Card		
<b>Other Associations</b>		
"stacked on"	Table	

<b>Class Name:</b> Team	<b>ID:</b> 9	<b>Type:</b>
<b>Description:</b> Maintains information about a team of players.		<b>Associated Use Cases:</b> Player/AI assigned to team. Team wins game. Team loses game.
<b>Responsibilities</b>	<b>Collaborators</b>	

Stores team name	
Stores team ID	
Stores list of players in team	Player
Maintains number of tricks won	Trick
Maintains score of team	Game
<b>Attributes</b>	
TeamName	
TeamID	
Members	
Tricks	
Score	
<b>Relationships</b>	
<b>Generalisation (a-kind-of)</b>	
<b>Aggregation (has-parts)</b>	
<b>Other Associations</b>	
“owned by”	Player

4. Class diagrams - the initial associations - name them, and add some (c.4) attributes and (c4.)methods/functions. (1)



## 5. Use case descriptions (Max 5)(based on Cockburn's template) and One Use Case Diagram



- **Start Game**
  - Description
    - A Player would like to start the game. This creates a server and a link for other players to connect to.
  - Actors
    - Player
  - Flow of events
    - The use case starts when the player requests to start a new game by opening the website at the root address.
    - The Server starts the game.
    - The Server redirects the Player to the new link for the game that has started.
    - The Player connects to the game as usual.
- **Join Game**
  - Description
    - A Player would like to connect to a game using a hyperlink.
  - Pre Conditions



- Another player has already started the Game and generated a link to the game, which this player possesses.
- Actors
  - Player
- Flow of events
  - The use case starts when the player requests to join a game by opening the website using a special link that contains an access token.
  - The Server finds the game in the access token and connects the player.

## 6. Result of “Structured walk-through” i.e. a walk through your complete game start to finish using (Max 2 pages) your Use Cases and [Classes](#)

### **Purpose-**

To cover all possible requirements and scenarios if occurred.  
It is a technique to detect errors to improve the product's quality.

### **Should there be a tutorial for the first-time players?**

Yes, a text-based menu is shown by the Table

### **When the server is trying to search for players to pair up with:**

The server creates a Game, which Players are created with and connected to. If there are insufficient Players when the Game begins, AI Players will be added.

### **Game Display:**

- ‘Shareable link to invite people’ at the top bar
- Team score will be displayed at the top bar
- Players can only see their hand of cards.
- Opponents’ names will be anonymous.
- Can only see other players’ cards when they play it
- Update the trump suit every round once the trick has been played
- Current trick + current trump
- Cards will be dealt out with no animation

### **Top Bar:**

- If a player is selecting a card: shows current player + is selecting a card (highlight player choosing the card)
- Not enough players: shows share link
- Before the game starts: the dealer is handing out the cards
- When a team/player has a trick?: update the score numbers

### **Start of the game:**

4 Players (represented by Player classes) will be seated across from each other.

Dealer randomly assigned.

Cards are given in a clockwise manner to all Players.

Each player gets 13 cards.

Dealer’s last card will be revealed to everyone. The suit of the card will be the trump of the round.

Each player plays out any card of any suit.

1 round completed is when all players have put down their card of choice.

The player with the highest rank card gets 1 point and gathers all the Trick which is given to the Team the player is a part of.

The trick is in the discarded pile and the cards are removed from the Players

If two cards of the trump suit are played, the higher rank card wins the round.

The game ends when the first Team gets to 6 tricks.

Server disconnects Players from the WebSockets.

**If a player leaves the game abruptly:**

If a player doesn't reconnect during a certain amount of time, the game stops.

**End of the game:**

The game finished.

Players disconnect from the game

Server ends the connection with the players that end the game.

## 7. Minutes/notes of team meetings (Max paragraph - just action items)

Date: 9th October 2020      Time: 9.00pm - 9.50pm

Participants: Gytis, Alan, Andrey, Melanie and Vincent.

To assign roles to everyone, we told everyone what we did over the summer. Alan prefers algorithms and assets.

Melanie chose to be a note-taker, class diagrams and write-ups. Andrey prefers Networking and can do any other work. Vincent can do Networking and class diagrams. Gytis can take care of the server and client stuff.

Discuss which game engine to run (either Godot or web-scripting). Decided to meet up every Monday and Friday. Every meeting we would be reviewing our work and get feedback from the team and discuss our goals for the next meeting.

Goals for Monday 12th October:

Do our own research to decide between Godot and web-scripting. Investigate Structured Walk Through and Refined Requirements Specification. Decide which programming language we will use for Whist. Figure out what web sockets are and how to implement them in Godot/ web. Read into State Machines. Examine the rules of Whist and explore which algorithms work best. Understand the rules of Whist and see what class diagrams we can derive from it.

Date: 12 October 2020      Time: 9.00pm - 10.00pm

Participants: Gytis, Alan, Andrey, Melanie and Vincent.

Vincent looked at different classes. For game class, could have a player instance, score. Alan looked at state machines, it's about diagrams about games and servers talking to each other. Melanie talked about refined requirements specification and discussed structure walkthrough. Andrey looked at the python option and sockets, investigated the ways the website can be designed and put up using Python libraries like dash and similar modules.

Gytis decided NodeJS was the most suitable language for Whist and web-scripting was chosen over Godot. We analysed the deliverables and divided the work among us. Examined the rules of whist and see what classes we need for the project.

Goals for Friday 16th October

Vincent will do class diagram deliverables and primary class list. Alan will look into scenarios. Melanie will start on refined requirements specification Andrey will work on class diagrams and use case diagrams. Gytis will start on user interface mockups and class skeletons

Date: 16th October 2020 Time: 9.00pm - 10.00pm

Participants: Gytis, Alan, Andrey, Melanie and Vincent.

Alan prompted the question if anyone has any enquiries regarding our tasks. Melanie has enquired about the rules of Whist. Vincent talked about the class list diagrams that would be used in the game; Whist. Andrey initiates talking about his use case diagram. The use case diagram describes the interaction between the actors and the system. Gytis shows his sketch of the user interface. Has a unique approach laid out. Using boxes to see each player layout the cards. Alan shows us the informal scenarios he came up with. Melanie enquired about the refined requirement specification.

Goals for following Monday:

Vincent will work on class list diagrams. Andrey will work on use case diagrams. Everyone looks through Refined Requirement Specification and Structured Walk Through. Alan will investigate object diagrams and sequence diagrams. Gytis will do detailed user interface mockups

Date: 19th October 2020 Time: 9.00pm - 10.00pm

Participants: Gytis, Alan, Andrey, Melanie and Vincent.

At the start of the meeting, we cleared doubts and gave feedback to everyone's progress. Vincent updated last week's primary class list. Andrey updated the use case diagram. Alan looked at object and sequence diagrams. Gytis examined sample user interfaces of Whist and discussed his own user interface for Whist. Melanie went through the structured walkthrough, questioning about the user interface and artificial intelligence implementation in the game.

Goals for Friday:

Gytis will implement design and generating links for the game.

Melanie and Andrey will expand the use case diagram.

Vincent will do the refined class diagram.

Alan will work on scenarios.

Date: 23rd October 2020 Time: 9.00pm - 9.40pm

Participants: Gytis, Alan, Andrey, Melanie and Vincent.

Andrey explained the use case diagram and updated any changes made. Goes through each use case and the interaction with the actors and the system. Alan shows his updated diagram for scenarios. Gytis implemented game server generation, link redirecting and cool card selection animation. Laid the foundations for the server and the user interface. Vincent made a slight modification in the class diagram, instead of having 4 player variables, the players are stored in a list in the game class. Towards the end, we all played Whist. Everyone took turns in playing and we made observations when playing Whist.

Goals for Monday 26th:

Merge all deliverables together in one document and look for inconsistencies in deliverables.

Date: 26th October 2020 Time: 9.00pm - 9.30pm

Participants: Gytis, Alan, Andrey, Melanie and Vincent.

In the meeting, we had a look through the written exam paper. After looking at the exam paper, the plan is for everyone to focus on every diagram so we all get exposure to the diagrams and we just pick and modify the best diagrams drawn by the team.

Goals for Friday 30th:

Melanie is responsible for summarising each meeting into a paragraph.

Andrey will work on state machines.

Alan will quality check everything.

Vincent will add the CRC cards to the main document.

Gytis will submit the first assignment, as well as write the use cases we're missing.