Start Document

Chessn't

Version

Version	Changes	Date
0.1.0	Creation of Start Document	31.01.2023
0.1.1	Added Idea and some mock up	06.02.2023
0.1.2	Added planning, Gantt chart, screen's description	08.02.2023
0.1.3	Added rule from dice, update Figma link	08.02.2023
0.1.4	Added Application description	09.02.2023

Version	Changes	Date
0.1.5	Added UML	21.02.2023

1. Group Roles

Tung Do Xuan - Front-End

Robin Michael Visser - Back-End

Sander Siimann - Back-End

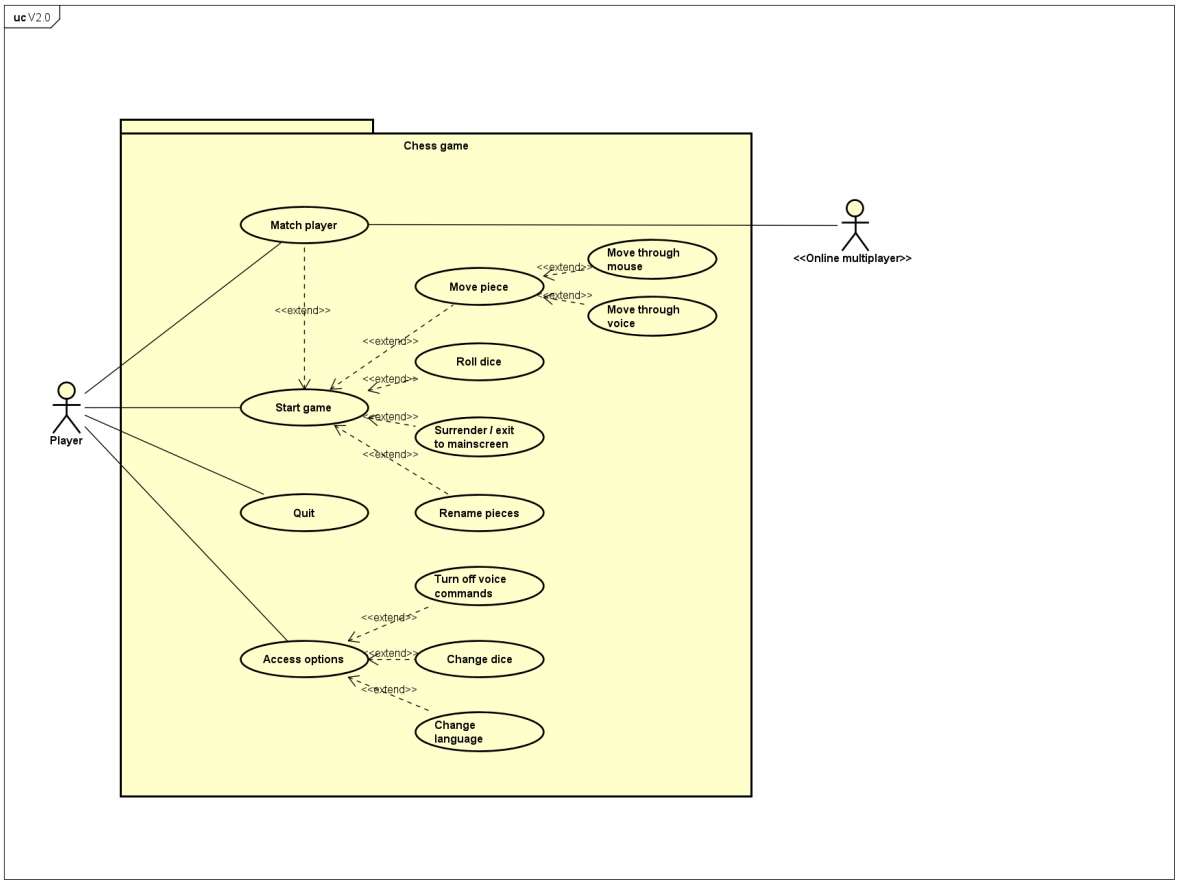
Levente Stieber - Front-End

2. Application Description

A group of student wants to play chess that is different from normal. Students want to play chess using voice commands and use a random set of rules through a DnD dice. User can use voice to move a piece (Ex. "D4 to F3") and also navigate through the application itself. User will be granted a dice roll randomly for his turn and receive a random rule from a set of special rules (**Chapter 7.3**).

The method that the application will be developed with is (MVVM) Model–View–ViewModel. this is a software architectural pattern that facilitates the separation of the development of the graphical user interface (the view).

3. Use Case Diagram



4. Mockup

The following are mockups for the application's main screens

Menu screen

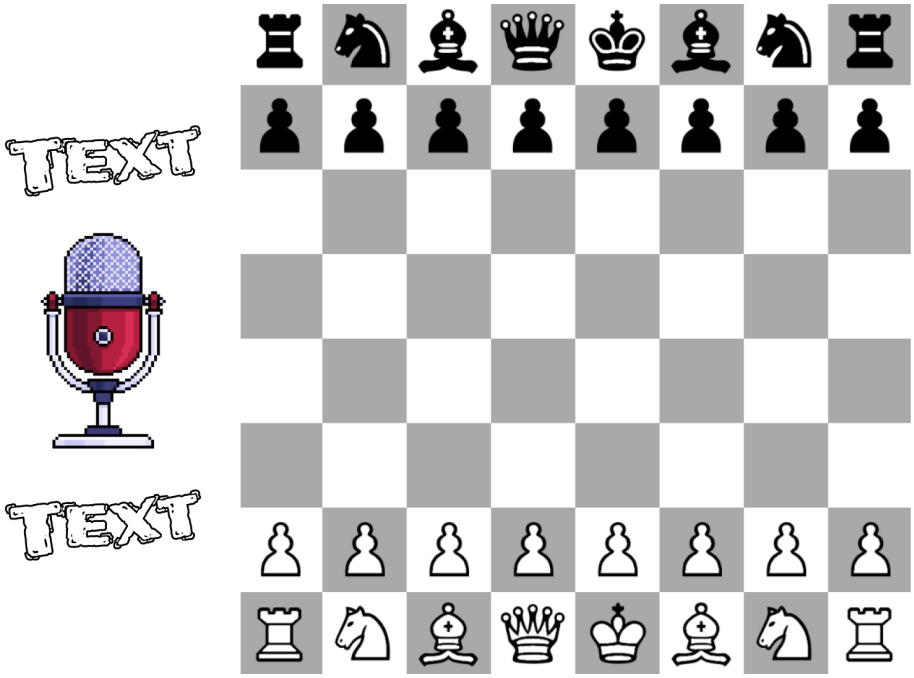
Main start screen with 3 options to proceed.

Option screen

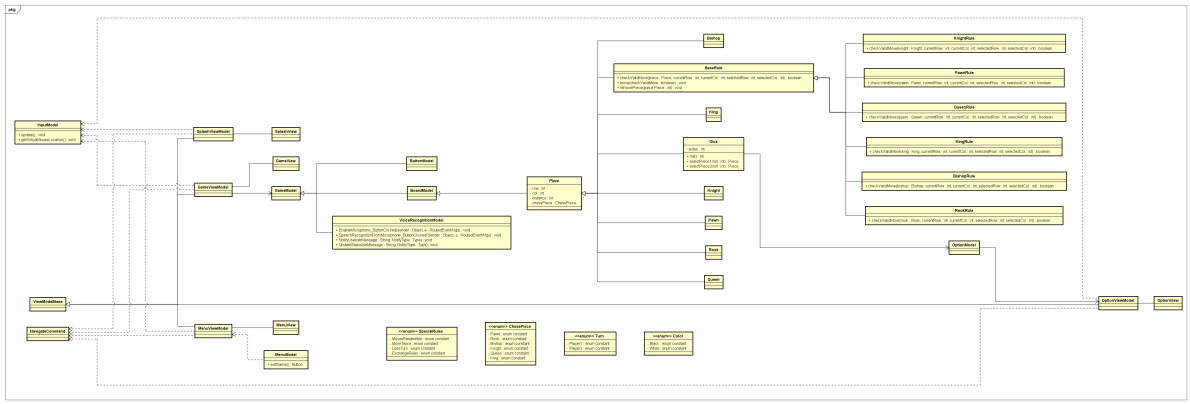
This screen would be use for changing dice and language.

Game screen

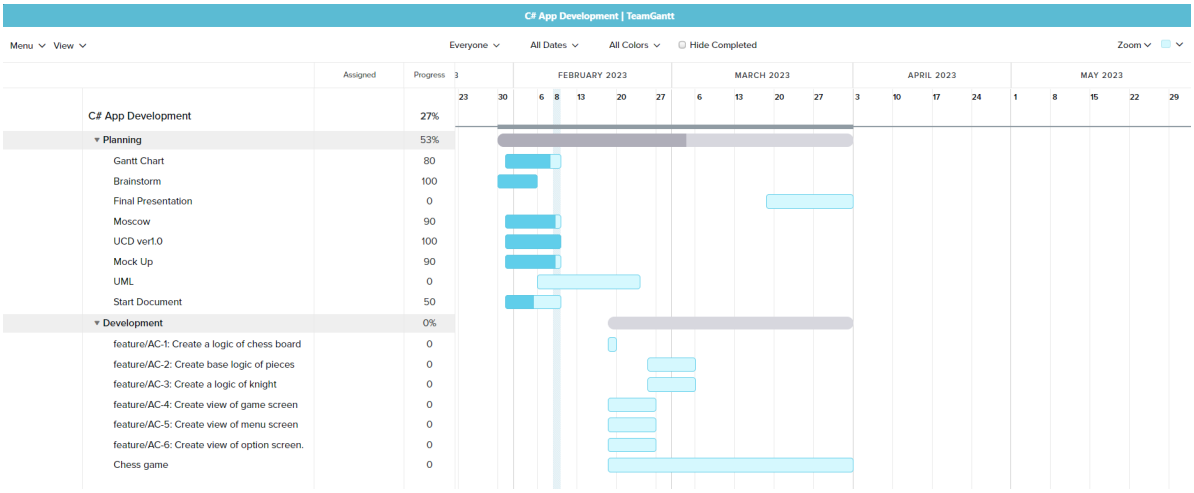
The game screen with chess pieces and microphone.



5. UML



6. Planning



1st week

```
Creation of start document and edition of start document, Seperating group roles and tasks,
Pitch idea to lecturers, Mock ups.
```

2nd week

```
Technical specification of project, start realization of project.
Send start document to lecturers.
```

3rd week

```
Designing class diagram.
Front-end: Splash Screen, Menu Screen, Option Screen, Game Screen.
Back-end: Chess pieces logic, voice command logic, dice logic.
```

4th week

```
Class diagram to be sent to the lecturers
Front-end: Connecting Views classes to ViewModel classes.
Back-end: Connecting ViewModel classes to Model classes.
```

5th week

```
Front end: Loading bar Splash Screen, connecting Splash Screen to Menu Screen,
Animation chess movement
Back end: Dice rules for chess pieces
```

6th week

```
Front end: to be updated
Back end: to be updated
```

7th week

```
Finalization stage:  Completing last steps of the application.
```

8th week

```
Testing phase, Preparing presentation
```

9th week

```
Presentation
```

7. Features

The Moscow method was used for this project to separate the various functionalities and determine which were more important than others. We used the Moscow method to separate the various functionalities within the various priorities.
This is what we came up with after conducting research and considering various functionalities:

7.1 Specifications

1. C# .NET 6.
2. Framework Monogame.
3. MVVM pattern.

7.2 Moscow

Must Have:

- Receiving input from user and parsing to the application.(Mouse input)
- Normal chess board with working chess pieces.
- Voice commands to move chess pieces. (Ex: "D4 move to F2")
- Animation for chess piece movement.
- 1 vs 1 locally from one computer.
- DnD dice to change the rule from classic one (Ex: D20 ICOSAHEDRON dice lands on 18, the game will loop 18 alive pieces from the user thus choosing the piece that would have special ability, that piece is now granted a special move to a random empty tile)
- Menu screen
- Option screen
- Game screen
- Main special rules (**Chapter 7.3.1**)

Should Have:

- Voice command to navigate through game.
- Quit the game.

Could Have:

- Online multiplayer through different computer.
- Scoreboard records win/loss from different user.
- Playing against computer.
- Different chess piece capture animation.
- Voice command: Rename pieces.
- Different DnD dice options.
- Different language option (E.g. Dutch, English)
- Extra special rules (**Chapter 7.3.2**)

Will Not Have:

- Tournaments
- Web hosting

7.3 Special moves from dice

7.3.1 Main special rules:

- Chess piece moves to a random empty tile.
- Chess piece moves twice in one turn.
- Chess pieces exchanges base case rules. (E.g. Queen gets Knight's move)
- Player loses turn.

7.3.2 Extra special rules:

- Chess piece moves to an empty tile the player chooses.
- Player can't move but instead can eliminate an opponent's piece by flipping a coin. (50-50%)
- A pawn on the board is converted to another piece.
- The player must move a specific piece (chosen randomly) on their turn, regardless of whether it's the best move to make.
- The player can move their opponent's pieces (as if they were their own) for one turn.

References

This APA references were used in the research and planning phase of the application.

(1): *Chessn't*. (2023.February 8). Figma.https://www.figma.com/file/d8h389V2wOFLojXXa27HPr/Chessn't?node-id=0:1

(2): *D&D Dice Explained*. (2023.February 7). Die Hard Dice Store. https://www.dieharddice.com/pages/dnd-dice-explained

(3): Wikipedia contributors. (2023, February 7). *Model–view–viewmodel*. Wikipedia. https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel

(4): Chessn't UML. (n.d.). Class Diagram. https://cdn.discordapp.com/attachments/1069641268956766292/1077901176663707680/UMLChessnt.png