Task

**Chess Game**

1. **Questions**
   1. Who will play the game? - **Two players**
   2. It is online game or in a single PC? **– Single**
   3. How much time do players have for a round? – **Players can decide the timing (from 5 mins to 30 mins)**
   4. What to do if a player does check to opponent’s king? – **Highlight the king’s place with red color.**
   5. How to show the movement of each figure? – **Highlight with some yellowish color, and red color, if a figure can capture another one.**
   6. What do we do if player do not drop the figure on the correct place? – **Place the figure where it was before**
   7. Do we have to save players’ games? – **Yes**
   8. Do we have to save current game? – **Yes**
   9. What to do if players want to continue previous game? – **Add “Continue” button in the menu.**
   10. Where will the data be stored? – **xml-format files**.
   11. Do we need an account system – **No**
   12. Do we need to add sound-effects? – **No**
   13. Should we add a “Back” button if one of the players made a mistake? – **No**
   14. Can player play against computer? – **No**
   15. Should rules be online or embedded - **Embedded**
2. **Big Picture Design**

**Structure:** WPF Application, BLL, DLL.

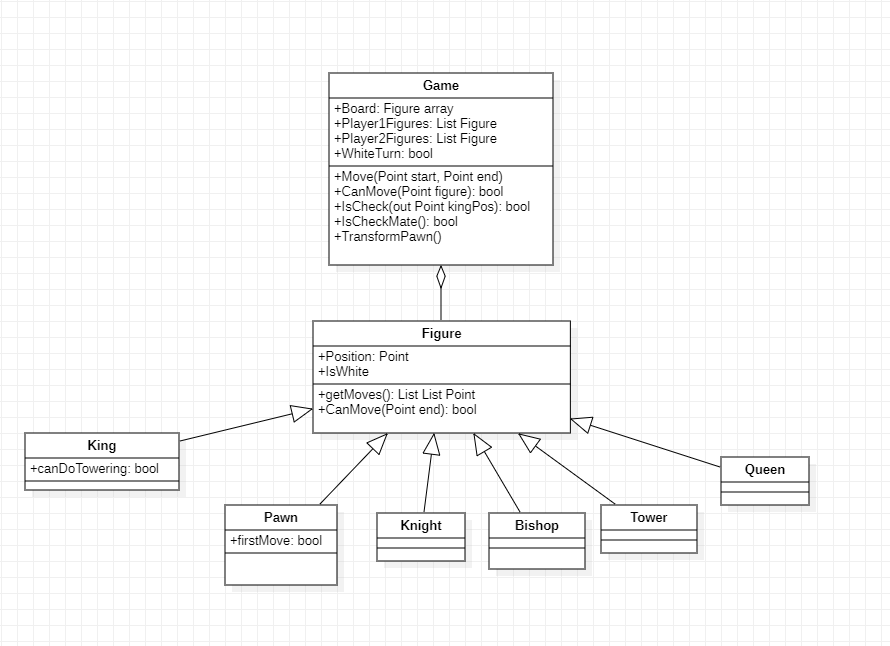
**Data:** XML-format files.

**Users:** Two players

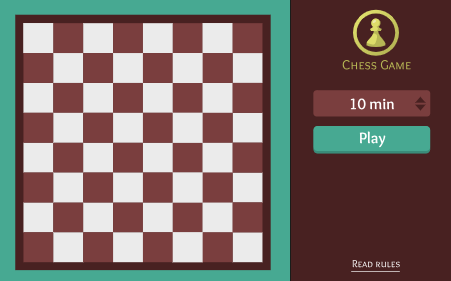
**Key Concepts:**

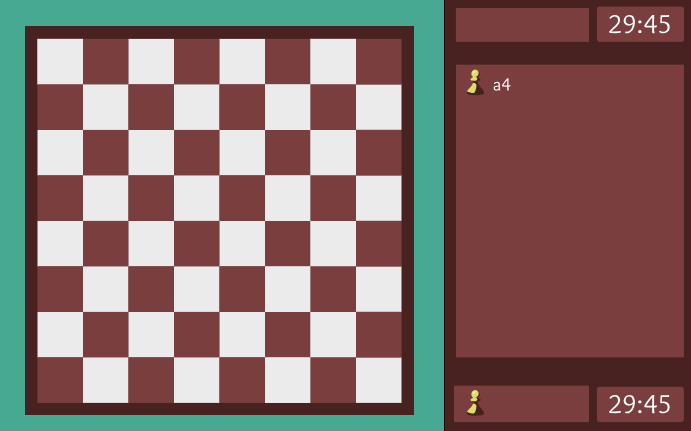
* Showing movements
* Saving games and current game
* Timer system
* Figure models
* Move validation

1. **Mapping the Data**

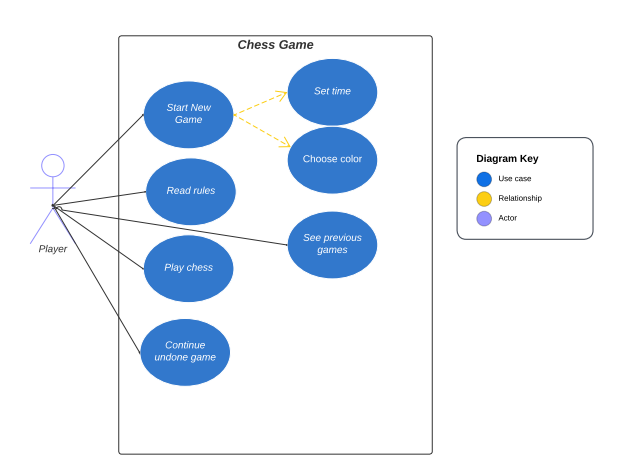
****

1. **UI of the Project**

****

****

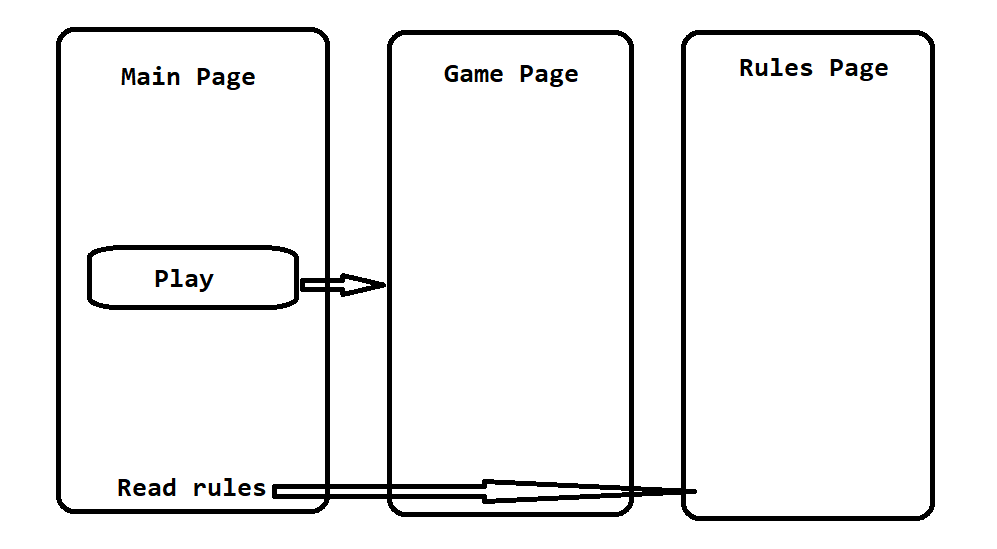
1. **Use-case diagram**

****

1. **Architecture of a program**

**6.1 MVVM**

We will use MVVM pattern to create the system of changing pages in our application. It is decided that there will be 3 pages: main page with the game settings, rules pages with the information about figures and game page with taken figures, timers and log to display the moves of the players.

****

**6.2 Adapter pattern**

This pattern is used to convert data to format that reduces the storage of memory to save data.

1. **Plan**
2. ~~UI of the MainWindow and main page~~
3. BLL

2.1) ~~Models of figures~~

2.2) ~~ChessGame service class~~

2.3) ~~Movement logic~~

2.4) ~~Attack logic~~

2.5) ~~Check and check-mate logic~~

2.6) ~~Towering~~

2.7) Test functions

2.8) Fixing logic errors

1. ~~Game Page~~
2. Timers logic
3. Loging moves
4. ~~MVVM for pages~~
5. Win window
6. Rules
7. DLL
8. Save user settings
9. Localization (not important)
10. Final tests
11. Fixing
12. Release