# Royal game of Ur

Barbora Dohnalová



### *Revisions*

Creation date: 05. 06. 2020
Document version: 01
Last updated: 10. 06. 2020

**Notes:**

DATE - NAME: note
05.06.2020 - Barbora Dohnalová: Document created
07.06.2020 - Barbora Dohnalová: Describe the rules
10.06.2020 - Barbora Dohnalová: Rest of the specification

## Product Goal

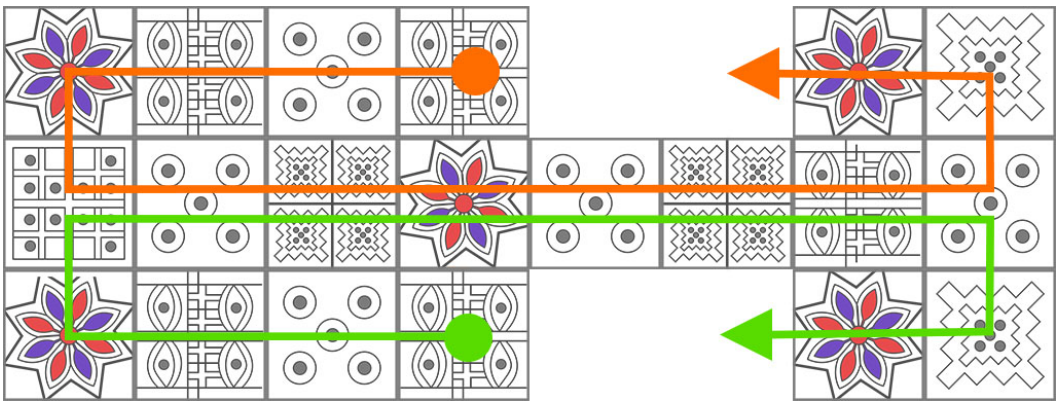
Implementation of the Royal game of Ur (using WinForms).

## Functional description

### Rules of the game

The Royal game of Ur is a race board game for two players. It is played on a board consisting of 20 squares, four four-sided dice and seven tokens for each player. The goal of the game is to get your own tokens to the end of the board sooner than your opponent.

The players take turns throwing the dice. Each dice has two of its vertices marked with a dot. The sum of the dots on all the dice thrown is the number of squares the player has to move along the given path (a number between 0 and 4; 0 means the player loses their turn). However, the player can choose which of their tokens to move, which adds a strategic aspect to the game.



(Path through the board[1], credits: https://gameofuronline.com/)

If the player moves their token on a space with their opponent's token, they capture it and this token has to make its journey again from the start. When the token lands on a square marked with a rosette, it is safe from capture (that means the opponent cannot land on this space even if he throws the right number). Also, the player who landed on it gets one extra turn.

To move your piece out of the board, you have to roll the exact number (e.g. if your token stands right next to the end and you throw anything other than 1, you cannot move this token). Whoever gets all 7 of their tokens out of the board first, wins.

## User interface

*Describe what the user interface will look like. What inputs it will accept. If there are multiple states of interactions, describe them all.*

- Main menu with possibility to view the rules and choose an opponent (buttons)
- Gameplay - point and click - player clicks to throw the dice and then chooses which of their tokens to move (if there are multiple possibilities)

## Functional requirements

*See: https://en.wikipedia.org/wiki/Functional_requirement*
*In a bullet list, describe what features your application will have with respect to the user.*

- Artificial player - multiple difficulties:
  - 0 - chooses tokens to move randomly
  - 1 - chooses token with the best outcome (good outcomes are getting a token to safety, throwing out opponent's token or landing on a rosette), if there is none, chooses randomly
  - 2 - chooses token to move with expectiminimax (version of minimax that takes random events into account) - similar to 1 but looks more moves into the future

[1] There are multiple possible paths through the board, it is not known which one used to be played

- Game implementation
- Main menu

## Data inputs

*Describe what data inputs your program will make use of.*

## Deadline

August 31th?