

LITTLE HORSES

CS-100 PROGRAMMING PROJECT--- FALL 2017

Prepared by:

MUHAMMAD MATLOOB ALTAF 2021-10-0164

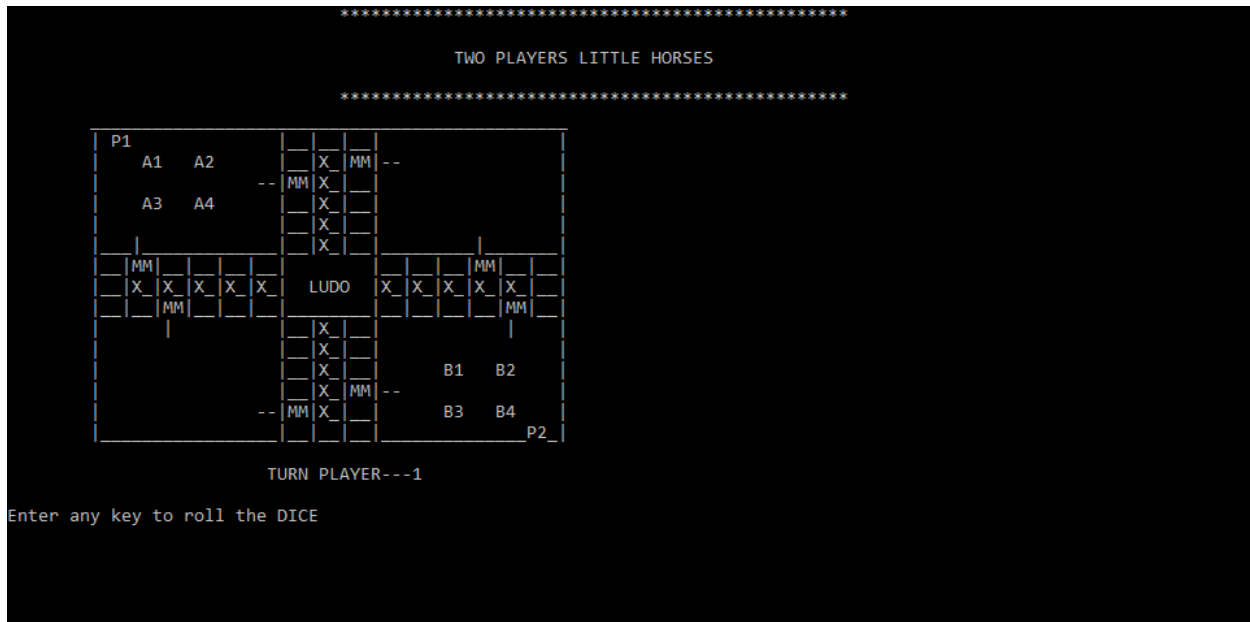
HAFIZ MUHAMMAD MOMIN IMTIAZ 2021-10-0179

INTRODUCTION

The “LITTLE HORSES”, which is the two-player implementation of The Ludo Game, is implemented as the programming project of CS-100 Computational Problem Solving, Fall 2017. The project is designed in computing language C++. This report will describe the architecture, instructions and game rule. Finally, the assurance case of the project, which shows that the project is solemn from all kind of logical errors.

ARCHITECTURE

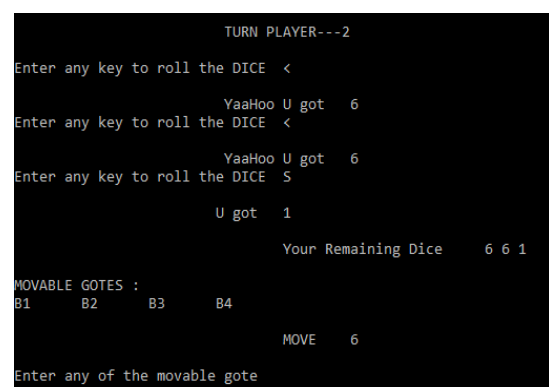
Following snap shows the output for Ludo board:



The whole output board is printed by using “cout” method and variable string is stored in empty places where the “gote” would move.

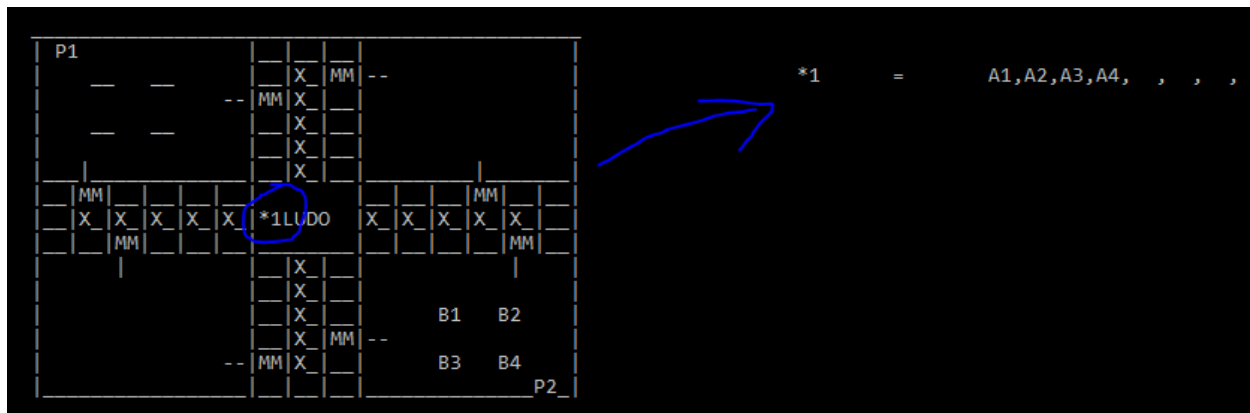


TURN PLAYER 1



TURN PLAYER 2

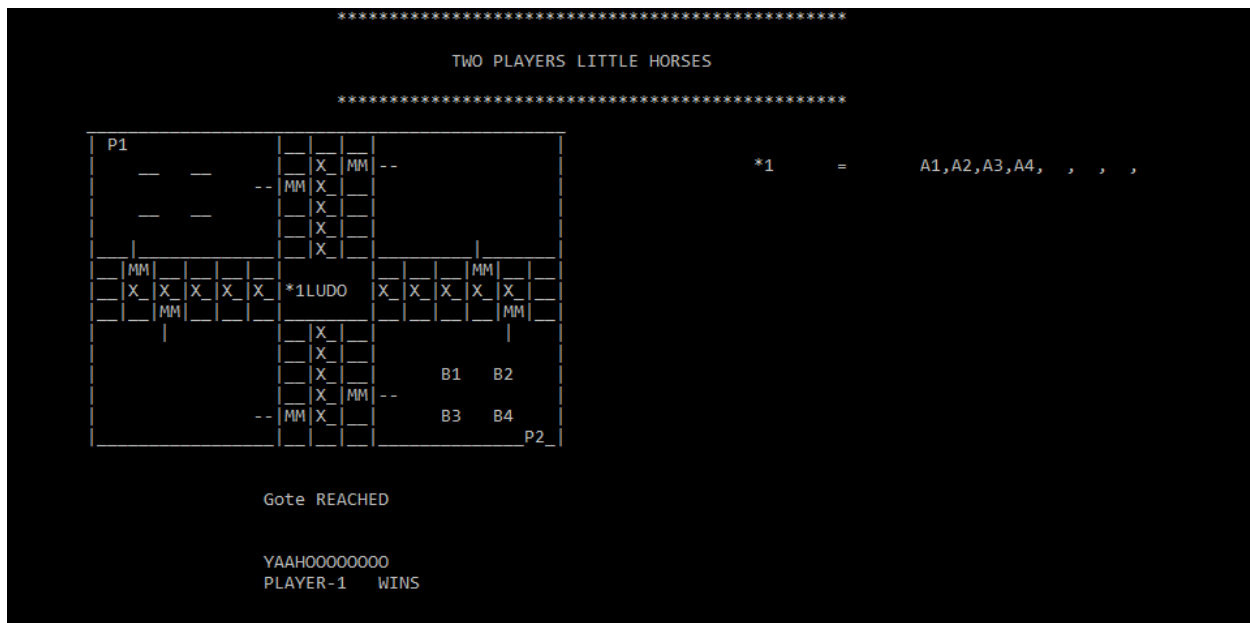
This is the output for each player which is displayed along with the Ludo board. It shows what you have got after dice roll and moveable “gotes”. And which ever “gote” is selected it moves on board accordingly.



If two similar “gote” comes on same place they two move in a new pair array (named S1,S2,S3 and so) and displayed on the screen as in above snap.

OPERATING INSTRUCTIONS

This game does not require any kind of additional information from the user. The game is not resume able as it does not store data. Also, you cannot quit game. At the end it displays the winner.



Ending Snap

GAME RULES

Game follows all the standard Ludo rules:

“The game can be played by 2 players. Each player has 4 game pieces, which are in the "out" area when the game starts, and which must be brought into the player's "home" row.

The rows are arranged in a cross position. They are surrounded and connected with a circle of

fields, over which the game pieces move in clockwise direction. There are 3 fields nearest to each

side of the board; the left one is the player's "start" field and the

middle one leads to the "home" row. This means that each game piece enters the circle at the

"start" field, moves (clockwise) over the board and finally enters the "home" row. The first player

with all of their pieces in their "home" row wins the game.

The players throw a dice in turn and can advance any of their pieces in the game by the thrown

number of dots on the dice. Throwing a six means bringing a piece into the game (by placing one from the "out" area onto the "start" field) and throwing the dice again. If a piece is on the "start"

field and there are still pieces in the "out" area, it must be moved as soon as possible. If a piece

cannot be brought into the game then any other piece in the game must be moved by the thrown

number, if that is possible.

Pieces can jump over other pieces, and throw out pieces from other players (into that player's "out"

area) if they land on them. A player cannot throw out his own pieces though and make pair

further than the last field in the "home" row."

TESTING

A through testing is done by playing the whole game and there is neither a logical error nor a run time error. Every specific case, which could be possible, is checked out to eliminate errors.

- Remaining numbers of dice are shown on screen for user convenience
- "Gote" of second player was not going to 0 after value 51, the issue was resolved
- Remaining "gote" in pair array was not being killed, the issue resolved

CONCLUSION

As the 99% of the game is coded in the concepts taught in class, we hope that you will like it.