Protocol to the team's SOMAZM Semestral group project

Mikhail Lyashenko, Sergei Chirkov, Mykhailo Kuvshynov, Olga Stepanova, Zhanna Turylo, Alexander Matveenko

Analysis

Our expert system is made to determine the next move for the player currently on turn in the board game "Člověče nezlob se" depending on the current game state. System asks the player questions about where his/her figures are located, how much has fallen on the dice and uses those answers to predict the best possible move for him/her.

For realisation of such an expert system multiple sources are needed. As a software we are using CLIPS programming language. Game rules and slides from BI-ZNS lectures are used for creating rules and writing code. It is expected from the whole team to have knowledge about CLIPS programming language syntax.

Creating this project might take up to a few days for a team of students who didn't have experience creating such expert systems before. However we successfully managed to exceed this time interval up to a few months due to a tremendous number of other deadlines and subjects :)

All sources that we use for making this project could be easily found and downloaded online.

Description

We separated the process of making this expert system into multiple steps. In each step there is information about what was made, how much time did it take, who, among team members, was responsible for this and why is this important for the project overall.

1. *Analyzing game rules and building a decision tree based on them.*
   Based on original game rules we build a decision tree that based on the player's answers will determine the best possible move for him. For example, the root node of the tree refers to question about the number of figures that player has on the table and in leaves there are question like: "Move figure 3 according to number that has fallen on the dice" or "You cannot make a move, you skip your turn" and so on...
   Colleagues Olga Stepanova and Alexander Matveenko were responsible for this step.

2. *Code structure planning*
   Rules from the decision tree from the previous step were implemented into a CLIPS program. This task took us a few hours to complete. We had to analyze which variables does this problem require and how rules and their relations with each other will be implemented. It took us approximately one day to complete. Colleagues Mikhail Lyashenko and Zhanna Turylo were responsible for this step.

3. *Implementation*
   Code structure from a previous step was implemented into an empty CLIPS shell. This stem took us half a day to complete since overall structure has already been planned.
   Colleagues Sergei Chirkov and Mikhail Lyashenko were responsible for this step.

4. *Code revision, making corrections and additions*

After the main part of implementation of the expert system was complete small changes such as spelling corrections, unnecessary rules removal, missing but important rules addition and so on… It is an important step for increasing project quality and functionality. This process took us 6 hours to analyse and complete. Colleague Mykhailo Kuvshynov was responsible for it.

Implementation

1. *Conceptual design*
   We used conceptual design as was recommended in lecture 4 of BI-ZNS.

2. *Implementation structure*
   Firstly the system asks basic questions to determine the number of user's figurines on the playing field and the number on the dice and saves them into a variable. Then the system gets information about whether the playing field is empty or not. If it is empty or the user has no figures in the game and he didn't score 6 he skips his turn. If the user has figures on the playing field the expert system gets information about all figures regarding distances to the end, enemy figures. It also determines whether after making a move one of the user's figures will land on the friendly or enemy figure. After all mentioned above, steps the system recommends how the user should make his move this turn.

3. *Writing CLIPS code*
   We used if-else constructions for decision making, based on the input, obtained from the user. Lop-for-count construction is used to iterate through figurines, if there is more than one. At the stage when all the information is given, there is a fact asserted to the system with all the basic information about the figurine. Then do-for-fact is used to check some certain info about the figurine. And any-fact is used for checking, if any fact matches the condition.

4. *Testing*
   We tested on the final version of the code to identify errors that we could have missed when developing our project. During trials on different game states and corrected all errors that appeared during that. This procedure took us about 3-4 hours. At the beginning, the test plan was compiled with all the combinations of which we adhered to during testing. All test combinations were supplemented later.

Potential Use

This expert system was made as a university project and turned in for evaluation using Gitlab.

References

- BI-ZNS Lecture 4 https://courses.fit.cvut.cz/BI-ZNS/media/lectures/bi-zns-04.pdf
- Game rules https://en.wikipedia.org/wiki/Mensch_%C3%A4rgere_Dich_nicht