SE 3860
Software Maintenance and Reengineering
Spring 2020

Assignment 1

Tasks

There exist many classic video games, which have unfortunately become outdated to the point where they can no longer easily run on current operating systems. This include several old games that ran on MS-DOS or other older systems. A list of games with available source code can be found at

https://en.wikipedia.org/wiki/List_of_commercial_video_games_with_available_source_code.
http://www.fraserking.co.uk/turbo-pascal-games-download.asp?MainCategory=turbo-pascal&SubCategory=turbo-pascal-games
https://www.classicdosgames.com/misc/source.html
http://www.wieringsoftware.nl/mario/index.html

Other games that you feel our also potential project choices can also be used. For your reengineering project you will choose an old game to reengineer such that it can run on current operating systems. The reengineered game must use a current Object-Oriented programming language (Java, Python, or C#).

In order to run the original game, you will likely have to replicate the execution environment. An emulator may be useful for this task. DOSBox is a popular emulator to run DOS games https://www.dosbox.com/.

You must be able to provide the original source code for the game and be able to run the game. You must also choose a game that you can reengineer given the constraints of the course (time, ability, knowledge). A game that would require knowledge and implementation using a custom 3D game engine may not be a good project choice. Remember that you will be reengineering the original source code and that you will likely be producing more lines of code that the original version.

Several DOS games have already been rewritten to run on current systems. If the game has already been rewritten in the past, you may use non-original versions of the game only if those versions are also outdated and cannot run on a current operating system. Some games have newer object-oriented versions which can run on current operating systems. You can still choose to reengineer the earlier versions of these games; however you cannot use the newer source code or copy from new existing implementation.

1. **Project Proposal**

   Provide the following information.

   a. Include the name of the game
   b. Include where you can obtain the source code for the original game
   c. Include the language the game was originally written in
   d. Include an estimate of the number of lines of code in the original game
   e. Include what language you intend to reengineer the game into
   f. Describe the scope of the project. The original game may have too much functionality to reengineer all features. State what functionality your new game will have in comparison to the old game.
   g. Include any software, libraries or tools you think you may need/use for the project.
   h. List all team members
   i. Include your team name

2. **Video of Game**

   Upload a video showing the original game executing. It may be difficult to get an old game running. This step is to ensure you can get the game running such that you can reengineer the requirements of the game.

3. **Team Report**

   Each team will submit a document stating how the team completed the work for the assignment (team dynamics). This should include a high-level description of what work each team member did (who did what). It should also include an estimated amount of time each team member spent working on their task. Explicitly state the time each team member worked on the project for each task they have worked on. I recommend using a table for this. Any issues of concern can be addressed in the team report (what were struggling points with the assignment). Include the team meeting times.

4. **Individual Report**
   Each student will submit an individual report in the separate corresponding assignment on canvas (the individual one). The individual report will include more specific details on what you have worked on. This differs from the team report, which will include only higher-level task information, where lower-level details can be discussed in this report. If the high-level details are sufficient, they can be restated. In addition to a more detailed description, any discrepancies in the team report can be included as well as any issues that you feel are important but did not feel comfortable including in the team report. The individual report should not be longer than a length of one page, however there is no strict page limit.

Submission

1. **Project Proposal**: Submit the project proposal. Include a link to the Google Doc in the submission that has edit privileges. Please name this document in the following form "<Team Number>_Assignment01".
2. **Video**: Submit an mp4 file of the running game.
3. **Team Reports**: Include a link to the Google Doc that has edit privileges for you team report. Name this doc "Assignment01_TeamReport_<team number>".
4. **Individual Reports**: Upload a word document for your individual report in the corresponding individual assignment on Canvas. Name this doc "Assignment01_IndividualReport_<your name>".

Grading

Criteria	**Possible Points**
Proposal	80
Video	10
Team Report	5
Individual report	5
Total	100

List of past projects

1. Tetris
2. Bloxinies
3. Mines
4. Future Blocks
5. Breakout
6. Oregon Trail
7. Battleship