CS7.302: Computer Graphics Assignment #1

Weightage: 15%

Due: March 25, 2025

Learning objectives

- Learn OpenGL graphics pipeline
- Learn 2D transformations
- Learn utility of callback functions

Tasks

The general aim of this assignment is to develop a simple game known as *marble solitaire*. Here are the rules and here is a working version. Extend the shared sample program to implement the following tasks. **[N]** denotes the weight of each task out of **15**.

1. **[2]** Use standard primitives available in OpenGL to draw multiple instances of squares and circles to represent houses and marbles, respectively. The board should be placed in the center of the scene. The time taken and the remaining number of marbles should be displayed in the top left and top right corners of the screen.
2. **[3]** Click, select, and drop mechanism to move the marbles from one square to another. Highlight current selection. Facility to undo/redo a finite number of moves.
3. **[6]** Game-play: Methods for checking valid and invalid moves. Methods for checking the end of the game, whether any possible moves remain, followed by the win/loss.
4. **[2]** Use *imgui* to create the interface.
5. **[2]** Report consisting of instructions to run the code, brief description of how *imgui* is incorporated, and any observations about the effort required for specific parts of the assignment.

Submission

1. Submissions should be made through github.
2. A valid submission would comprise of all the code (with makefile, readme, etc..) with adequate instructions to compile and run the code, the code **should** compile without errors.
3. Please do not upload executables.
4. Deadline: **23:59 hrs, March 25th**.
5. Any submission beyond the deadline will be considered as **late submission**. Late submissions will be evaluated out of **10** instead of **15** if submitted by **23:59 hrs, March 28th**.
6. Submissions past March 28th will not be evaluated.

Notes

1. All the rendering should happen on the GPU, that is, squares, circles, highlights, etc.. should all be drawn using the vertex shaders.
2. Avoid hard-coding values. Define constants/values in the global section so that changing and testing the code is easier. You may be asked to do so during the interactive demo.