First Assignment - 2019-2020

A

This assignment consists in the creation of playable chess board with its sets of black and white pieces, 32 in total.

A.1 Chessboard

The chessboard is composed of 8 × 8 black and white squares and each player has a set composed of 8 pawns, king, queen, 2 bishops, 2 horses and 2 towers.

In a first step you must be able to create the board plane and assign it the black and white squares texture. Note that this the board must be a single square.

The pieces are to be created according to the next table

Piece	Shape
Pawn	Cylinder
Tower	Cylinder with another cylinder on top. The second cylinder must be 25% larger than the first and its height be 25% of the first.
Horse	Parallelepiped (mostly vertical)
Bishop	Cone
Queen	Cone with a small inverted cone on top
King	Cone with a small Cylinder on top.

These pieces may be generated as a single mesh per piece. Although solutions using separate meshes will be also accepted as long as the selection and movement is properly done in terms of data structures and algorithms.

This chessboard must be playable in a sense that two people may use it to play against each other. This means that each of them may be able to select one of his pieces and move it, and eventually capture one of the opponents pieces. The selection of pieces is achieved by moving a ring or sqare along the board with the keyboard arrow keys and selecting with the space key, then moving it again with the arrow keys and releasing again with the space key. Capturing one piece is done by selecting a piece and pressing letter "c" on the keyboard.

Movements and keys: the movements are to be only using the keyboard. There are two possibilities, using the arrow keys and/or using the numeric keypad. In the latter case you may optionally add diagonal movements, otherwise there are going to be done by composition of vertical and horizontal movements.

The player's turn is selected by pressing the key "t". As a result the view must change to that of the corresponding player. Pressing "a" will toggle from the current player's view to that of a more or less distant spectator.

A.2 Environment

You should place the chessboard in an environment which can be a room where the chessboard is on top of a table. The room has a ceiling lamp, a door opening, a window and a picture on a wall. You should distribute these elements accordingly to your aesthetic preferences.

Through the door and window we can see the outside landscape composed of a green field and blue or dark sky depending of the time of the day. The ceiling lamp may be on or off, depending on user's choice, and can be acted by toggling it on key "s".

When not playing the user may move around the room, and look in any direction. Choose the appropriate way of providing this interaction. The only restriction is that the "p" key toggles between playing the game and traveling freely on the room.

A.3 Mandatory

It is mandatory to develop this implementation totally based on OpenGL Version>=3.2 with the respective vertex and fragment shaders. All the 3D primitives must be constructed by your own code, no external libraries are allowed for that, not even GLU.

The shaders must be used using the DEECshader class provided, textures from images must be loaded using CGRAimage class also provided.

In case of need you should convert files from other formats (PNG, JPG) into PPM using ImageMagick or other available tool.

A.4 Grading

What	%
Chessboard + table	10
Pieces using the proper shapes and colours	20
Selection ring or square with its movements	10
Complete keyboard interaction for game and viewing movements	10
Picture on the wall (in rectangle or frame)	10
Door and window openings	10
Outside greenfield	10
Outside blue+dark sky	10
On/Off Lamp	10
Overtall aesthetics	+10 bonus
Illumination Effects	+15 bonus

A.5 Plagiarism

If N groups present too similar code or parts of code, the classification will be divided by the number of copies.

A.6 Deliverable

The deliverable should be a ZIP or TAR.GZ file **(no other format will be accepted** containing:

- All c++ or python files
- All shaders
- All textures
- A Makefile that compiles the c/c++ source code into executable.
- A report (2 to 4 pages) describing the implementation in PDF. The report classification is between 0 and 1 and has a multiplicative effect on the overall obtained grade.

A.7 Presentation and Discussion

All groups must present their works and be able to justify and explain every part of the code delivered, otherwise the classification will be zero.