



GAME2012 – 3D Graphics Programming

Midterm

Instructors: Alex Richard (arichard6@georgebrown.ca)
Hooman Salamat (Hooman.Salamat@georgebrown.ca)

This is an individual practical test

You will have 48 hours to complete this midterm, though it was designed to be completed in 2. The midterm link will disappear after a window and no submissions will be accepted. Make the best use of the time. For this midterm, you'll be rendering 2D and 3D objects in a small area as well as manipulating the camera and using shaders.

Overview

You can find a Zoom video link to a video walkthrough of a solution to get an idea of what the result of the midterm can look like. To summarize, it will be a 10x10 mini maze with 2 unit-wide halls and a 4x4 final "room" with a sliding door and three cubes inside. Details follow in this document. You cannot use the layout in the solution nor the exact colors aside from the cubes, and you will get enough freedom so that plagiarism will immediately be detected. **It is NOT worth risking 0% (or worse depending) for a 30% midterm.**

Start Project

You **MUST** use any starting project that we have provided for you in this course, so that means **MUST** use Hooman's template, either a lab or lecture example. You will be drawing shapes from scratch so there should be nothing else besides these requirements in the window. It helps to choose one that has:

- Indexed draws with cloned vertices
- Drawing multiple shapes in one render
- Keyboard and mouse input
- Manipulating the camera

I did mention which one would be best to work with in a recent lecture, and yes, this means you can use any code Alex or Hooman has provided.

Project Customization (2 marks)

Add your name and student number to the window title. **(1 mark)**

Make the window background black – usually the default we have. **(1 mark)**

Perspective Configuration

Use a perspective view with the following attributes: **(1 mark)**

- Any degree amount that is >45 and ≤ 60 .
- 1:1 aspect ratio.
- a display range of 0.1 to 100 units.

Camera Config

Set your camera with the following attributes:

- A starting position that is slightly above your ground level (0), with your x and z position outside the maze but in front of the entrance to your hallway. Put z several units back, e.g. 5-10. **(1 mark)**
- Set the following directional movement: **(3 marks)**
 - W and S for forward/back.
 - A and D for left/right.
 - R and F for up/down.
 - Q and E rotate the camera just on its Y axis, left/right
- You MUST use bitflags to do this, and the example I hinted at to start with has a lot of this smooth camera movement done. You just add more.

Do not set the speed too fast or too slow or you will lose the 3 marks if it is either.

Grid Creation (3 marks)

For the floor of the maze, you must create a 10x10 (no less but you can go bigger) flat grid where it's a checkerboard pattern and the details are thus:

- Each tile of the grid should be 1x1 units.
- Each tile should also have another rendered square that is just wireframe.
- There should be an alternating color pattern like a checkboard with any two colors.
- Begin the grid with the first tile being flat on the ground at 0,0,0.

Maze Design (1 mark)

You have a lot of freedom to create the maze and I have provided a grid jpg to help you. You MUST draw out your maze, even in paint and it must match your implemented maze in OpenGL. It must be submitted too.

Wall Creation (12 marks)

For the walls of the maze and a bit more of the design, here are the details:

- You must turn off culling – which will also make this easier.
- The walls should be at least 4 units high but no more than 6. They must all reach the same height.
- The walls must be of a different color than anything else.
- The walls, like each grid square must have a white wireframe too.
- You must have a hallway with a bend or few, which leads to a 4x4 room, and your hallway must be 2 units wide. No more no less.
- Your room must be 4x4 units and must have a wall that has room for a 1x1 doorway.
- You have freedom to design other features like dead ends. You must also have walls AROUND the maze, except for where your hallway starts.

Door Creation (2 marks)

In the doorway, there will be a sliding square of a different color than anything else that moves back and forth in the doorway. There's no collision so just have the door move back and forth for the length of the entire 1 unit (like my solution) but not too fast.

Cubes Creation (5 marks)

In the "room" you must have three cubes created with indexed draws and the details are as follows:

- The cubes, of course must fit in the room so scale them small enough so all three can fit comfortably in a line.
- Turn culling back on for the cube rendering.
- The three scales for the cubes MUST be different.
- The cubes all must have a set of opposite sides that are white on one side and black on the other. Could be top/bottom, front/back or left/right.
- For the colors of the other sides, one cube must be red, one cube must be green, and one cube must be blue.
- The colors of the sides cannot be interpolated, and if they are, you don't get the marks. I had examples for this.
- Rotate the different cubes thus:
 - Red cube must rotate on one axis only.
 - Green cube must rotate on two axes only.
 - Blue cube must rotate on all three axes.
 - Rotation speeds must be different and both directions need to be used for any combination of two cubes.

Coding and Misc

Aside from the mandatory use of Hooman's template with shaders, you can create any code, including functions that will help you. Also, in the interest of time, you can copy/paste any of your own code in the .cpp for repeated creations of wall segments or other things. Transforms and rebuffering of colors are going to be a must. I am not going to deduct marks for messy code.

- Everything you need can be found in Alex's lecture examples.
- It took him 1.5 hours, and most of it was creating the walls.
- It will take some trial and error to properly scale and rotate the walls in place, but that's why it has the most marks.

Midterm Weight: 30% of course grade

Task	Possible Marks	Description (see mark breakdowns in the outline)
Customization	2	You customized the project properly
Perspective Config	1	You configured the perspective properly
Camera Config	4	Camera placed outside the maze in front of your hallway - 1 You move the camera properly in all 6 directions - 2 You can also rotate the camera with Q and E - 1
Grid Creation	3	You created the base grid for the maze properly - 2 Your grid tiles have alternating colors - 1
Maze Design	1	You have a thorough design for the maze
Wall Creation	12	Your walls all use one set of data (a square) - 1 There is a wireframe included - 1 The walls are transformed and placed properly and look like some effort was made (not just a few walls) - 7 There is a path that just doesn't go straight to the "room" - 1 The walls are all of the same height - 1 You turned culling off for the walls - 1
Door Creation	2	The sliding door is of a proper size and place - 1 The door slides back and forth in the doorway repeatedly - 1
Cubes Creation	5	The cubes are of different sizes - 1 The cubes are different colors - 1 The cubes have opposite white/black sides - 1 The cubes rotate in different directions and speeds - 1 You turned culling on for the cubes - 1
Total:	30	

Submission Details:

- Delete the hidden .vs directory .
- **Submit the ENTIRE project, NOT just the .cpp and shader files.**
- Add your design image or you will not get the 1 mark.
- No video or GitHub required this time.
- Submit through Midterm link in D2L ONLY.
- Naming convention:

GAME2012_Midterm_LastnameFirstmane.zip (or .7z or .rar)

Penalties:

- You submit only a .sln file: **0% for submission**
- You didn't use Hooman's template: **0%**
- You only have an example that we provided **and no extra code**: **0% for submission and plagiarism report**
- You coded it in any language other than C++ with OpenGL: **0% for submission**
- You still have a hidden .vs path in your submission: **1 mark off**
- Wrong naming convention: **1 mark off**
 - So make sure you take 10 seconds and rename it!