

Project 4 – Real-Time 3D Game Programming

Due: April 16th (11:59pm)

Overview:

In this assignment you will make an interactive 3D game using OpenGL with a programmable shader pipeline. The game will be a first-person game where the user explores an environment to find a goal. The environment will have impassable obstacles along with doors/gates that can only be opened with the right key.

Map Format:

The environment the user explores will be represented by a **map** that stores the world as a **rectilinear grid**. The map will be stored in a text file where the first row is the width and height of the environment. The player will start at the point marked **S** and must reach the point marked **G** to finish the level. The player's progress may be hindered by walls (marked **W**) and doors (marked with a capital letter **A-E**). Throughout the environment will be keys (marked with a lowercase **a-e**). A player may not cross through a door unless they have picked up the corresponding key. A player may never cross a wall.

No Doors

```
5 5
0000G
WWOWO
OWOWO
OWOWW
S0000
```

Door w/ Key

```
5 5
0000G
WWOWO
OWAWO
OWOWW
S000a
```

Requirements (80 points):

- *Continuous Movement:* The player must move continuously through the environment (not in discrete jumps between grids).
- *Walls & Doors:* Each element type must have a unique rendering. Walls must look different than doors, and each of the five doors must look different than each other.
- *Keys:* Each key must be rendered as a physical object (e.g., a cube, teapot, or key model). Whenever a player moves the key must be rendered in front of them moving with the character.
- *User Input:* Users must be able to move smoothly around the map with mouse or keyboard input. One possible (easy) solution is to rotate with the left and right keys, and move forward and back with the up and down keys.
- *Collision Detection:* Users must not be able to move through walls or locked doors.
- *Floors & Ceilings:* Floors must be rendered, the ceiling is optional.
- *Lighting:* Your scene must have some ambient and diffuse lighting.
- *New Maps:* You must create new maps showcasing your game engine.

The above features are 80 points, for full credit (100 points) add additional features from the list below (ask us if you have something in mind not on the list). It's okay to extend the scene format, but include a scene showing the new features!

Additional Features:

The number is how many points a feature is worth. Partial credit is possible.

Lighting & Materials:

- (5) Add multiple Point Lights to the scene file
- (5) Create a flashlight that moves with the player (must be a spot light)
- (5) Texture map the walls and floors (use different textures for different models)
- (5) Bump map or normal map the walls and floors

User Interface

- (5) Integrated keyboard and mouse control (must support strafing/sidestepping)
- (5) Jumping (must be smooth & realistic for full credit)

Dynamic Environment

- (10) Add characters to the environment which move on their own (perhaps chase the user slowly). These characters should not move through walls.
- (10) Character interaction: Allow the user to freeze or shoot the other characters at a distance (you cannot freeze or shoot through walls!).
- (5) Place items in the environment that alters a character's abilities (power ups).
- (5) Animate the process of doors opening after they are unlocked.

Environment

- (5) Procedurally generate levels (there should always be a solution)
 - (5) Create your own 3D models to represent the user, keys, doors, non-player characters etc. (e.g., with <http://www.wings3d.com/> or <https://www.blender.org/>).
 - (5) Load existing models in the OBJ 3d format: [http://enwp.org/Wavefront .obj file](http://enwp.org/Wavefront_obj_file)
- It's okay to support only some of the model features (e.g., only triangles)

Miscellaneous (highly suggested)

- (5) Make a video of yourself playing, showcasing your features
- (5) Two artistic/well-composed screenshots from your game

Submission Instructions:

Create a sample webpage with:

- A zip file of all your code
- Several output images from your game (including at least one of the sample scenes) demonstrating all the working features
- Brief description of your implementation, any issues you saw, and a list of any extra credit tasks you attempted
- Submission to Art Contest [5 points for submissions] (optional)

Hints:

- You probably want to start with one of the working OpenGL examples
- Your first task should be to analyze the file format to figure out how to generate walls.
- Only try doors and keys *after* you have walls displaying correctly, and users moving through them.
- There are several easy points (e.g., make a video), check out all the options.

Extra Credit (*Optional, up to 20 points*):

The assignment is graded out of 100 points. The required material is only 80 points, and you must complete an additional 20 points of work to receive full credit. If you have 100 points worth of working features, you may turn in up to 40 points worth of additional features to be graded as extra credit, but only at half the value (e.g., 140 points of perfect features would turn into $100 + 0.5 \cdot 40 = 120$ points). *You must explicitly identify which features you are submitting as extra credit* as these are the features that will count for half. If you don't list which features are normal and which are extra credit, we will randomly choose 100 points worth of features and grade those. As with all other features, be sure to document your extra credit working through text and illustrative images in your report.