

Assignment 4 – 3D Games

Due: November 20th (start of class)

Getting Started:

The goal of this assignment is to make an interactive 3D game using OpenGL with the programmable pipeline.

Map Format:

The map stores the world as a rectilinear map. The first row is width and height. 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):

- *Walls & Doors:* Each map 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 around the map with mouse or keyboard input. One 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.

The above features are 80 points, for full credit (100 points) add additional features from the list below (ask me 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 (spot light)
- (5) Texture map the walls and floors
- (5) Bump 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.
- (5) Animate the process of doors opening after they are unlocked.

Environment

- (5) Procedurally generate levels (there should always be a solution)
- (5) Use models to represent the user, keys, doors, non-player characters etc. Extra points if you create these yourself (e.g., with <http://www.wings3d.com/>).
- (5) Load models in the OBJ 3d models 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

- (5) Make a video of yourself playing, showcasing your features
- (5) An artistic screenshot from your game (in a working or broken state ☺)

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 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 (optional)

Hints:

- You should start with one of the recent OpenGL examples on the webpage
- 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.