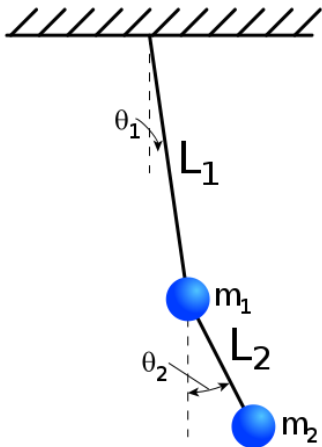


Computer Graphics 2018 - Homework 1

In this homework, you're required to write a program that simulates several double pendulum systems and render it using OpenGL.

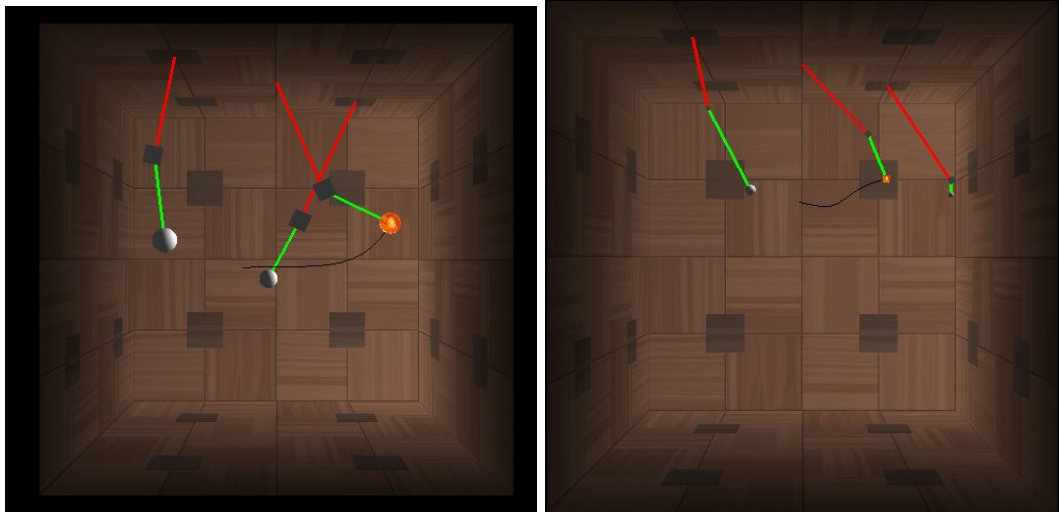
Terminology :

- Double pendulum system : (from wiki) a pendulum with another pendulum attached to its end, and is a simple physical system that exhibits rich dynamic behavior with a strong sensitivity to initial conditions.



- Rod : The line that connects the ceiling and the bob, or between two bobs.
- Bob : The sphere in the image that provides weight to the system.

Result :



Objective overview :

1. Render three double pendulum systems and five planes that form a box around the pendulum systems.
2. Apply textures on the planes that form the box and the lower bob of the center double pendulum system.
3. Set up lights so that the lower bob of the center double pendulum system is the light source.
4. Keyboard callback function that controls the position and orientation (look-at position) of the camera.
5. (Bonus) Draw a trajectory following the bottom bob of the center pendulum system.

Details in each part :

- Double Pendulum System (55%) :
 - Render 3 pendulum systems with motion(that includes 6 rods and 6 bobs). (40%)
 - The top bob of the system should be a cube, and should be rotating so that the top face of the cube is always perpendicular to the rod. (5%)
 - The scale (or radius) of the bob should be directly proportional to the cube root of its weight(The bob should be larger if its heavier). In short, your code should be something like :

```
float scale = powf(topBobWeight, 0.33) * someConstant;  
// ... some other code  
glScalef(scale, scale, scale);  
// ... draw the bob
```

- The rod and the light source bob should be rendered with light disabled. (5%)
 - The texture of the light source bob is provided, which is "Resource/sun.bmp". (5%)
- Camera Controller (20%) :
 - Camera shall be able to rotate along three axis (X, Y and Z axis) with respect to the world position of camera. (10%)
 - Camera shall be able move along three axis in world space. (10%)
 - Input (Please use these keys, actual functions of each key is not restricted) :
 - A, D: move camera on X axis
 - W, S: move camera on Y axis
 - Q, E: move camera on Z axis
 - J, L: rotate camera along X axis
 - I, K: rotate camera along Y axis
 - U, O: rotate camera along Z axis

- Planes (15%) :
 - The planes shall form a box but excluded the front face. (10%)
 - The texture of the plane is provided, which is “Resource/ceiling.bmp”. (5%)
- Bonus (10%)
 - The length of trajectory should be a fixed number specified by user. (5%)
 - Other bonus is welcomed. (5%)

Restrictions :

- When rendering double pendulum systems, please use `glPushMatrix()` and `glPopMatrix()` to enter local space. If you render everything in world space, there will be penalty.

Scoring Criteria :

- Double Pendulum System (55%)
 - Render 3 pendulum systems with motion (40%)
 - The top bob of the system should be a cube, and should be always perpendicular to the rod. (5%)
 - The rod and the light source bob should be rendered with light disabled. (5%)
 - Texture (5%)
- Camera Controller (20%)
 - Rotate (10%)
 - Translate (10%)
- Planes (15%)
 - Render 5 planes (10%)
 - Texture (5%)
- Bonus (10%)
 - Trajectory (5%)
 - Other Bonus (5%)
- Report (10%)
 - Please hand in a report to briefly explain how you finish the assignment.

Deadline:

Please hand in your code and report using E3, deadline will be 2018/10/29 23:59:59.
 Penalty of 10 points of the value of the assignment per late day

Running Environment:

Visual Studio 2017 is recommended.

You may specify your target environment by providing a README file, but it's better to inform the TAs beforehand, just in case we cannot execute your program.

Upload Format:

Please hand in your project file and report as STUDENT_ID.zip to e3 platform.(ex : 0656602.zip).

Hint:

1. You may use these functions below:

```
gluPerspective, glColorMaterial, glLightfv  
glGenTextures, glBindTexture, glTexParameterf, glTexImage2D, glTexEnvf,  
gluNewQuadric, glutSolidCube, gluSphere, ...  
and some other functions mentioned in the class before.
```

2. The side length of the texture image should be 2^N bit.
3. You can try the difference between GL_REPLACE and GL_MODULATE in the function `glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);`
4. You should bind a texture with its texture id, and then start to use it to draw (remember to enable), or adjust its parameter.

example:

```
glGenTextures(number of textures, array to store texture ids);  
glBindTexture(GL_TEXTURE_2D, a texture id);  
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);  
//scale linearly when image bigger than texture  
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);  
glTexImage2D(GL_TEXTURE_2D, 0, 3, image_width, image_height,  
0, GL_RGB, GL_UNSIGNED_BYTE, image_data);  
glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
```

5. You may need to disable LIGHTING or TEXTURE2D when drawing some objects.

References (Double Pendulum):

- https://en.wikipedia.org/wiki/Double_pendulum
- https://www.youtube.com/watch?v=uWzPe_S-RVE
- <https://www.myphysicslab.com/pendulum/double-pendulum-en.html>

Source code reference (Texture mapping):

<http://www.programming-techniques.com/2012/01/how-to-do-texture-mapping-in-opengl.html>