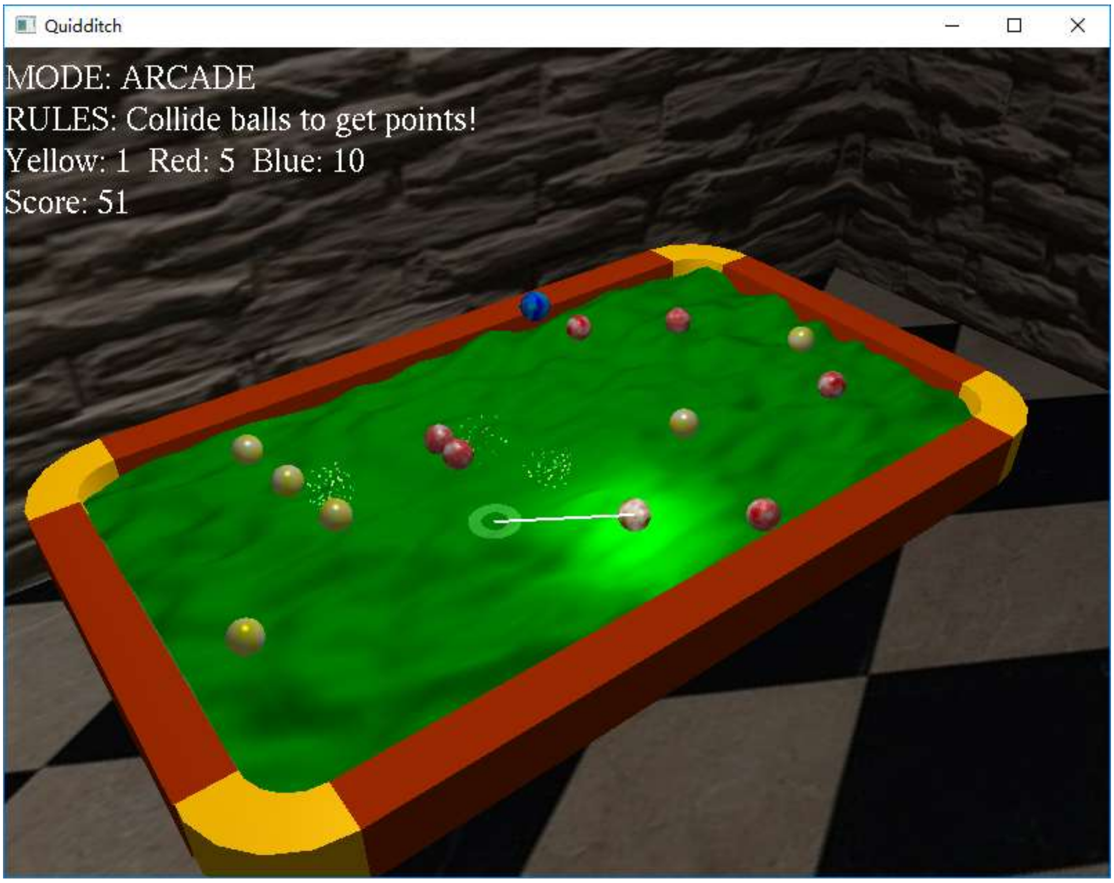
Quidditch OpenGL Game

Introduction

This is the final project of my course “Computer Graphics”. In this projects, I design a billiard game. The user can control the white ball whose name is the mother ball to collide with other balls to gain scores.

The focus of this project is not the playing method but the exploitation of OpenGL.

Instruction



MODE: ARCADE
RULES: Collide balls to get points!
Yellow: 1 Red: 5 Blue: 10
Score: 51

The player can click a place on the screen as the destination. Then there will be a white line connected the current place of the mother ball and the destination. After you press the **space** key, the ball will be issued as a specific velocity.

To make the game more interesting, the topography of the table is up and down instead of a plain. The height is generated by perlin noise to make it similar to the real world’s rise and fall.

All the picture attached to the balls are generated by perlin noise to make it realistic.

The player can change the prospective via pressing “**WASD**” keys.

Difficulties

1. It is difficult to imagine the sequence of matrix transform which can make the ball rotating like in the real world.

   Each time the ball rotating, we need to calculate the rotating matrix:

```
        glPushMatrix();
        Vec z = delta.dc(Vec(0, 1, 0));
        glLoadIdentity();
        glRotatef(-theta, z.x, z.y, z.z);
        glMultMatrixf(rM);
        glGetFloatv(GL_MODELVIEW_MATRIX, rM);
        glPopMatrix();
```

When drawing the ball, load the matrix:

```
        glMultMatrixf(rM);
```

2. After the user click on the screen, I design a ripple to show where the user has clicked as a feedback. However, to avoid it being blocked by the table or balls, I need to close the lighting and open the blend mode to make it unblocked.
3. To make the game more verisimilar, I use perlin noise.

Key Requirement and Implement

1. The up and down of topography, which is implemented by using perlin noise
2. Special Effects
   a) Collide rainbow sparks (Particle Effect)

   I use Spark class and SparkComponent class to meet this requirement, which generate a random number of particles towards random directions whose color is also random after a collide between balls.

   b) Water Ripple

   As mentioned before, I use a ripple which grows larger and more transparent in the `MouseResponse` class.

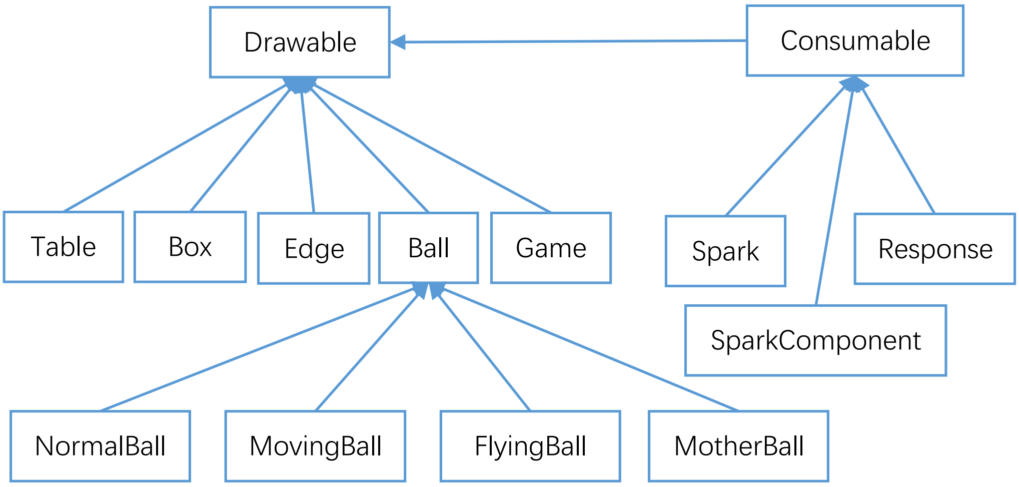
   c) Spotlight

   The player can press ‘o’ key to open or close the spotlight which is always above the mother ball.

   d) Perlin Noise

   Use perlin noise to generate the picture attached to balls and the topography of the table.

Architecture



The class **Drawable** is the father class of all elements which need to be draw on the screen.
The class **Consumable** represents the elements which may vanish as the time passes such as the spark (class **Spark**) and water ripple (class **Response**).
To organize the elements orderly, each drawable has an array storing its child **Drawable** elements and its **Consumable** children. Under this kind of organization, we can iteratively draw the whole game elements easily and deconstruct all the objects easily.