# Quidditch World Cup

Yuqing Guan (yg2392), Jihan Li (jl4346), Yilin Xiong (yx2274)

## Main goal:

Inspired by the world-famous fiction, Harry Potter, we use OpenGL to implement a Quidditch game. A user can select one among the four houses is Hogwarts: Gryffindor, Ravenclaw, Hufflepuff and Slytherin. After selecting one team, our program will randomly assign a rival team controlled by an artificial intelligence.

We simplified the Quidditch game to improve the maneuverability of our game. When playing the game, the user can control a wizard/witch and let it fly by a broom. The players will compete to get a ‘Quaffle’ ball. When one player of the user’s team catches the ball, the user can fly with this ball, pass it to teammates or shoot one of the rival team’s holes to score. One team will get 10 points for each goal, and the first team with 150 points wins.

## Algorithm:

Basically, we use algorithms to detect and resolve collisions, and do with artificial intelligence.

* Collision Solution: We define a class named CollisionObject which has child classes of both player and ball. When player or ball collides with the boundary of the stadium, it will bounce back in the direction of reflection, and at the same time, gravity will make it fall. When a ball without a holder collides with a player, the player will immediately take the ball. When a player collides with other players, if one of the two players has the ball, the ball will be taken by the other one; if neither of the players has the ball, both of them will bounce back and fall to the ground.
* Artificial Intelligence:

Our artificial intelligence controls three types of roles: the ball holder of the computer team, the attacker of the computer team, and all the other players not controlled by the user.

When the ball is taken by a player in the computer team, he will fix the goal and try to move towards it. When he is close enough to the goal, he will automatically score the goal. If a player of the user team tries to catch him and take the ball, he will either move away or pass the ball to the nearest teammate.

When the ball is taken by the user, one of the nearest player of the computer team will become the attacker. He will fix the ball and try to catch the user player. If he collides with the user player, he will take the ball away and become the ball holder.

Each of the other players in both teams has a random timer embedded. When the global timer coincides with the player timer, this player will change his direction towards the ball, and try to keep a distance from the ball. This avoids more than two attackers appear and collide.

Meanwhile, all the players controlled by AI except for the attacker are set to avoid collisions with the other players and the boundary of the stadium. They will automatically change their directions by a certain angle.

## Results and achievements:

### Rendering:

* Background: We use a skybox as the background. If there does not exist a terrain map, our program will randomly generate one terrain and smoothen it by Gaussian and mean filters, otherwise our program will load the existing terrain.
* Stadium: We import a model of a Quidditch stadium, locate it in the center of scene
* Player: We use a model of a wizard with joints and links. The textures of the coat, shirt and emblem of this model are replaced according to the player’s team. We use forward kinematics to change the pose of players during the whole game. Also, each player is combined with a broom.
* Collision object: The players and the ball all have a collision radius within which two objects will be detected as collided and then trigger control and AI functions
* Minimap: In order to help users find the positions of players and the ball, we use a minimap to indicate all collision object in this field.
* Models, textures and shaders: We develop loaders to import OBJ and IQE models, including the vertices, UV coordinates, normals, meshes and textured materials. For IQE models, we also import joint and link information. All models are rendered by shaders based on Bling-Phong reflection model.
* 2D Rendering:
  + - TODO render

### Interaction and control:

* On-screen control: We use ‘screen’ objects to manage 3D models and interactive objects.
  + The loading screen will use the progress and logs of the loading process by multi-threading.
  + After loading or exiting from the playing screen, you can see a start screen with two buttons. You can enter a new game or just exit our program.
  + Team selection screen and play screen: In team selection screen, you can choose one of Hogwarts’s four houses as your team and our program will randomly set another house to be the rival team. After selecting your team, you can enter the play screen to play the game. When the game starts or one of the two teams wins, a CG animation will be played.
  + Modal dialogs: When the user requests to exit the program or the current game, a modal dialog will be displayed to ask for the user’s confirmation.
* Mouse control: We provide buttons in the start screen, team selection screen and modal dialogs. Buttons are bind with listeners to perform different actions.
* Keyboard control: A keyboard is the main controller of our game:
  + All-time control:

|  |  |
| --- | --- |
| Key | Effect |
| Q/ESC | Exit game in play screen  Cancel operation in modal dialogs  Exit program in other scenarios |
| F2/F12/P | Screenshot |
| F11 | Toggle full screen |
| Enter | Start new game in start screen  Confirm operation in modal dialogs |

* + In-game control:
    - TODO How to play our game
* Animation: Generally, we provide camera animation and model animation for starting the game, ending the game and shooting a goal. We write key frames of our animation as strings, and store them in an array. Then we parse the string array into actual behaviors of camera or models. For translation, we record the starting point and ending point, and do linear interpolation frame by frame; for rotation, we record the initial direction and the rotation angle, and do angular interpolation based on Euler angles model.

### Artificial Intelligence:

TODO AI, Please introduce it in detail in the ‘Algorithm’ section and mention it briefly here

## Contributions:

* Jihan Li: Team leader, starting menus, animation, game interaction and control, AI design.
* Yuqing Guan: Basic framework, scene, models, collision objects and forward kinematics.
* Yilin Xiong: Loading menu, modal menus, collision solution, game control, AI design.

## Notes:

* In order to test our program, please extract ‘Final\_ yg2392\_ jl4346\_ yx2274.zip’, change directory to ‘Quidditch’ and then execute ‘ant’.
* Because we use multiple threads in the loading process of our program, this program cannot be X11 forwarded if it is tested on a remote machine. Therefore, please test our program in local machines.
* Since our program involves the rendering of many complex shapes and textures, it is better to run our program with a high-performance graphics card.

## References:

* Inspiration: Harry Potter Series
* Sky: <http://www.braynzarsoft.net/vision/index.php?p=VT&texture=25>
* Terrain: Yuqing Guan’s first programming assignment
* Backgrounds:
  + Loading screen: [http://www.texturex.com/Wood-Textures/red+wood+texture+grain+ natural+wooden+paneling+surface+photo+wallpaper.jpg.php](http://www.texturex.com/Wood-Textures/red+wood+texture+grain+%20natural+wooden+paneling+surface+photo+wallpaper.jpg.php)
  + Start screen: <http://harrypotter.wikia.com/wiki/1994_Quidditch_World_Cup>
  + Green buttons: <http://pixshark.com/subscribe-button-green-png.htm>
  + Wooden buttons:
  + Modal dialogs:
    - [http://www.texturex.com/Wood-Textures/red+wood+texture+grain+natural+ wooden+paneling+surface+photo+wallpaper.jpg.php](http://www.texturex.com/Wood-Textures/red+wood+texture+grain+natural+%20wooden+paneling+surface+photo+wallpaper.jpg.php)
    - <http://guide-to-pottermore-items.blogspot.com/2013/01/ps-chp-11-golden-snitch.html>
* Emblems of Hogwarts, Gryffindor, Slytherin, Ravenclaw, Hufflepuff:
  + <http://rapunzelalltangledup.blogspot.com/2012/03/harry-potter-documents-gift.html>
* Progress bar:
  + <http://www.mundusbellicus.fr/forum/archive/index.php?t-8676.html>
  + <http://www.wallpaperseries.com/textures/blue-diagonal-line-pattern-background-wallpaper.html>
* Quidditch stadium: <http://tf3dm.com/3d-model/quidditch-stadium-65161.html>
* Player: <http://blackrosebunny.deviantart.com/art/MMD-WHAT-294589531>
* Broom: <http://www.turbosquid.com/FullPreview/Index.cfm/ID/681862> (Purchased $12 by Yuqing Guan)