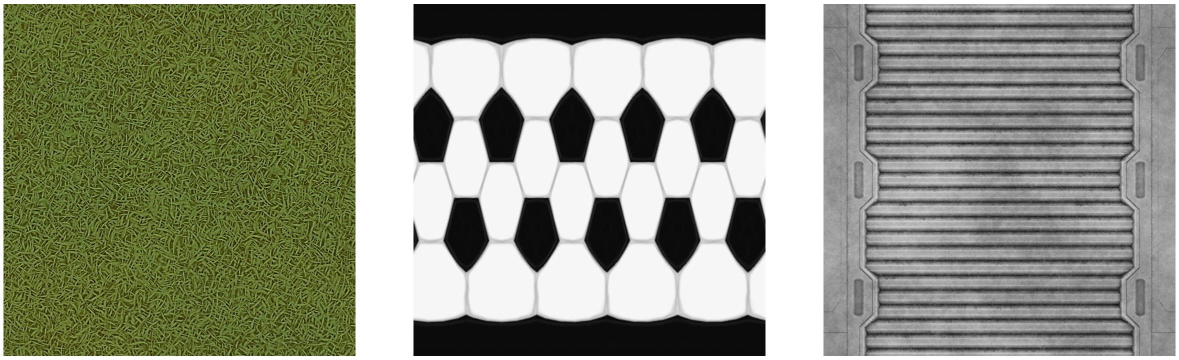
Computer Graphics
Assignment 4

Lovejeet Singh Parihar (IMT2019048)
Avantika Mudumbai (IMT2019013)
Utkarsh Padariya (IMT2019089)

May 2022

1 Description

- In this assignment, we had to create a VR simulation of simple 2 player football-like game.
- In this game, the aim of a player is make a goal, while avoiding the opponent as well as the obstacles on the field.
- **Game Play:** A player can do the following:
  - Carry the ball
  - Dribble the ball
  - Kick the ball towards the goal
  - Push the ball forward
- **Lighting:** There are a total of 7 lights:
  - 4 fixed lights for each corner of the field.
  - 2 lights fixed at the head of each player.
  - 1 search light for the football
- **Textures:** Images were used as texture maps along with normal, roughness and displacement maps.



(a) Ground

(b) Football

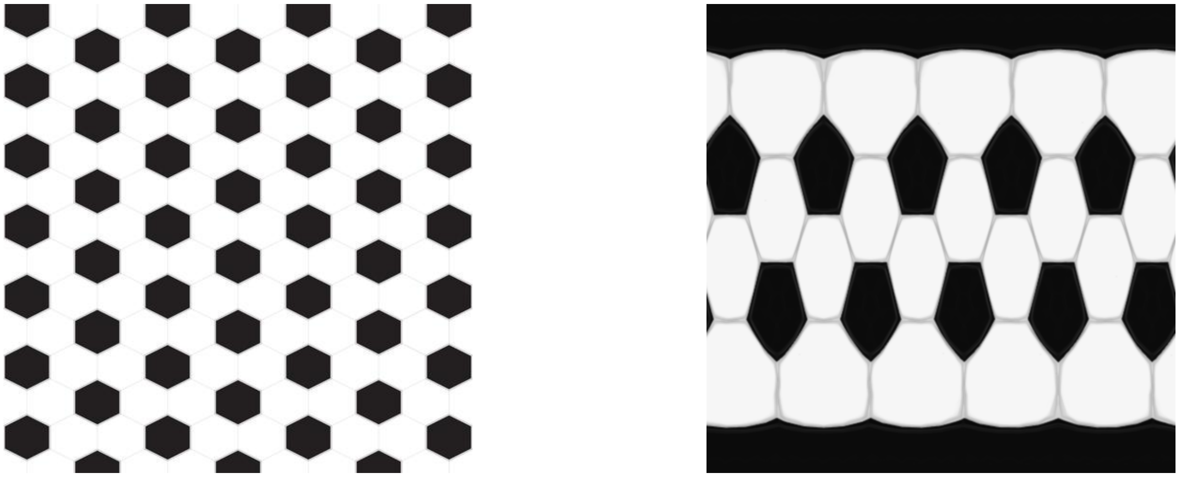
(c) Player

Figure 1: Image maps used

- **Views:** There are 3 modes of viewing the scene:
  - Box view: See the entire scene from a height
  - First Person Perspective: See the scene from the perspective of the player
  - Third person Perspective: Look at a player in the scene as an audience member

2 Question

What kind of distortions do you notice with the texture maps you have used? What would be your approach to correcting them?
Answer: Initially, we used an image with regular hexagons as shown in figure 2a. But since this was to be used for a spherical object, the hexagons were distorted in the rendering. To correct this distortion, we used an image map as shown in figure 2b. The pentagons in this figure are distorted such that when the image is used as a texture map for a sphere, the pentagons appear regular.



(a) Initial Map

(b) New map

Figure 2: Football image maps

For both the maps, the upper region is squished when its spherical map is applied onto the square. For the first map, as it is uniform, this results in decrease in the size of upper hexagons. Whereas for the second map, when the upper part got squished, it got converted into a single black spot that is similar to what we really see in a football.

It is similar to how a map is pasted on the globe or a horizontal map is obtained from a spherical globe. The upper and lower regions look wider when they are shown in the maps.

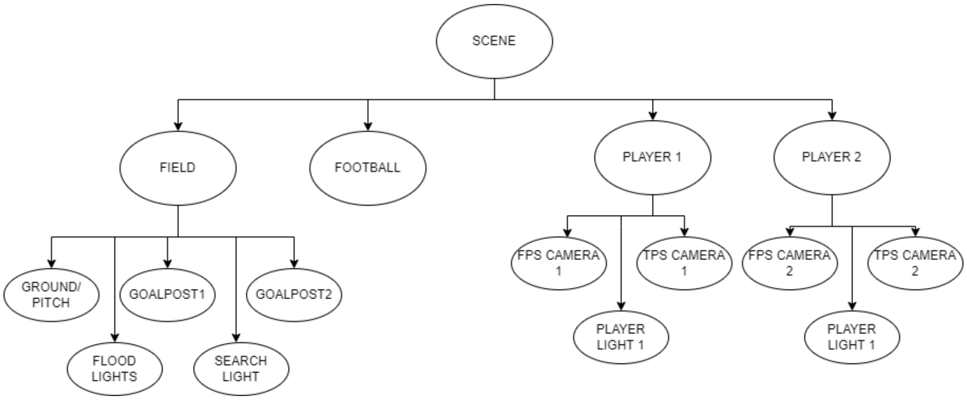
3 Design Description

3.1 Scene Graphs

We used a scene graph while rendering the the entire scene. This scene graph is used to ensure that the position of the objects in the scene maintain certain constraints.

As seen in the figure, the scene has 4 components: the field, two players, the football. The field is the static part of the scene, ie, the ground, goalposts, the flood lights and the search light of the ball.

Since there are two views related to the player, first person perspective and third person perspective, there are 2 cameras dependent on each of the player. The relative position of these cameras do not change with respect to their player.



3.2 Lights, Cameras and Viewing

There are three views available: box view, first person perspective and third person perspective. In the box view, the entire scene(within the viewing volume of the camera) is rendered. For this view, there is a perspective camera placed at the point (0, 160, 400). In this view, the user can zoom in, zoom out and rotate the camera using the mouse movements (virtual trackball implementation).

In the first person perspective, the scene is rendered from the point of view of the player. This is done using a camera placed on the head of the player. There is also a light placed at the head of the player to support the player's camera. As the relative positions of the camera and light must not change with respect to the player, the camera and the light are maintained as a "child" of the player in the scene graph. We can also change the direction in which the camera points at using the corresponding key bindings. Note that changing the direction of the camera will not change the direction in which the player moves.

In the third person perspective, the scene is rendered from the point of view of a third person, "spectator", looking at the player. This view is rendered using a camera placed at a height and distance from the player. This camera can move only along a sphere centered at the player (used mouse movements and trackball to implement the rotation).

Apart from the lights on the head of the players, there are 4 stationary lights that are placed at each corner of the field. These lights illuminate certain portion of field like flood lights in a stadium. There is also a search light which is placed at a height from the center of the field. The position of the light is is constant but its direction changes and it always points towards the ball.

3.3 Collision Detection

The collision between two objects is detected by checking for the intersection of their bounding boxes. For the collision of the ball with the obstacles, the ball is bounced back from the surface in a random direction. The speed of the ball is reduced due to a loss of energy at impact. This factor by which we have reduced the speed of the object is the coefficient of restitution.

For the collision of the player with the obstacles, if there is an obstacle in the vicinity(within a step) of the player, the player cannot move in that direction. Even if the player attempts to move in that direction, there will be no change in the position of the player, ie, the player will remain stationary. Also, if the player in motion comes in contact with the opponent, then the opponent "steals" the ball.

Resources used

1. WebGL fundamentals
2. ThreeJS documentation
3. Texture maps
4. Object Geometries
5. Stack Overflow