

Yakup Talha Yolcu

1801042609

CSE462 AR

HW4

Ray casting involves sending rays from the camera through each pixel of the image, and checking if any of the objects in the scene intersect with the ray. If an object intersects with the ray, the pixel is colored based on the material and lighting of the object. If no objects intersect with the ray, the pixel is colored black.

objects is an array of GameObjects, which are the objects in the scene that will be rendered in the image.

cam is a Camera object that represents the camera from which the image will be rendered.

imageWidth and **imageHeight** are integers that specify the dimensions of the image in pixels.

fov is a float that represents the field of view of the camera.

aspectRatio is a float that represents the aspect ratio of the image (the ratio of the width to the height).

lightSources is an array of Light objects, which represent the light sources in the scene.

image is a RawImage UI element, which is a display element that can display a texture.

In the **Start()** function, which is called when the script starts, the following steps are taken to render the image:

A new **Texture2D** object called texture is created with the dimensions specified by **imageWidth** and **imageHeight**.

The image object's texture property is set to the texture object.

A nested loop iterates over each pixel in the image.

For each pixel, a Ray object called ray is created by calling the **ViewportPointToRay** function on the cam object and passing it a Vector3 with the normalized x and y coordinates of the pixel and a z coordinate of 0. This creates a ray that passes through the center of the pixel. The **Physics.Raycast** function is called with the ray object and a **RaycastHit** object called hit as arguments. This function shoots the ray and checks if it intersects with any colliders in the scene. If the ray intersects with an object, the hit object is populated with information about the intersection point, the normal of the surface at the intersection point, and the collider that was hit. If the ray does not intersect with any objects, the hit object is not modified.

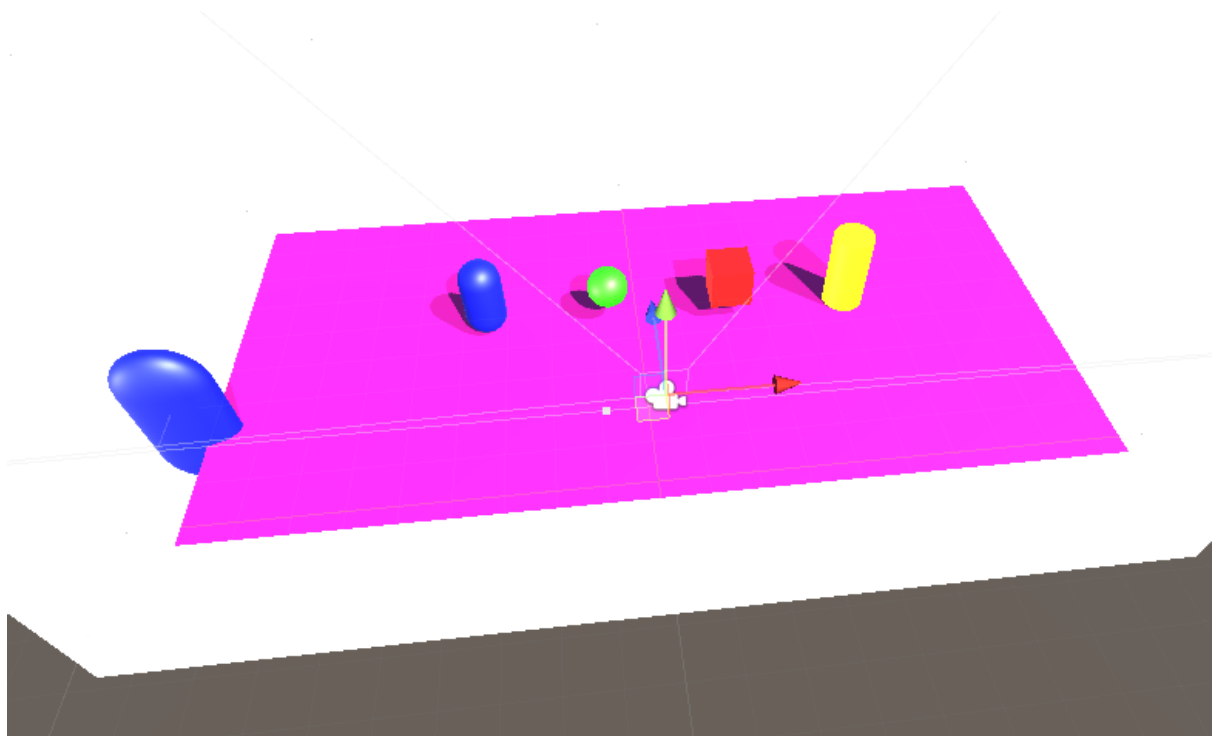
If the **Physics.Raycast** function returns true, meaning that the ray intersected with an object, the pixel is colored based on the material and lighting of the object. First, a variable called **hitCount** is initialized to 0. This variable will be used to keep track of how many light sources are hitting the intersection point. Then, a loop iterates over each Light object in the lightSources array. For each light source, the sendRay function is called with the light source

and the intersection point as arguments. If the **sendRay** function returns 1, it means that the light source is hitting the intersection point, so **hitCount** is incremented. After the loop finishes, the **newColor** variable is set to the color of the object's material. If hitCount is 0, meaning that no light

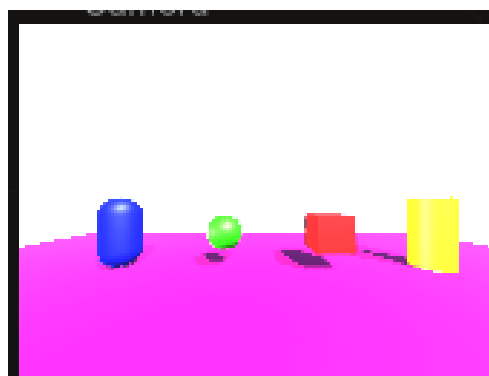
After that, we create file to save the png file to given directory

Here is an example of scene and generated image:

Scene:



That's what camera sees:



Generated image:

