**COMP 3770 Assignment 1**

**Group #30**

Aisha Ali

Matthew Eppel

Jake Haas

Efe Osah

9. [30 marks total] Explain briefly, for each primitive object, why it behaved the way it did. Why did it move, why didn’t it move? What did you add or remove from the object that made it behave how it did? Put this in a document called “Explanations.docx” and put it in your project folder.

The **transparent teal plane** remained stationary throughout the simulation. The plane did not have a Rigidbody component but did have a collider component. Because of this, other objects in the scene which were physically enabled such as white sphere 2 and red sphere 1 treated the plane as a solid object. In order to achieve a transparent appearance, the material’s rendering mode was set to Transparent as opposed to all other objects in the scene which were set to Opaque, then the correct material RGBA values were applied to set the desired color and transparency.

The **orange sphere** moved back and forth along the x-axis at a constant rate of one unit per second. Because the object contained a Rigidbody component set to kinematic mode, it simply moved without calculating physics on itself and pushed all other objects with Rigidbody components out of the way. We achieved this movement by using a C# script which incrementally modified its position by 1/50th of a unit every 1/50 of a second as defined by the FixedUpdate method.

The **purple cube** has the script that makes it resize, which it does. It also collides with the green capsule and almost the orange sphere, but has no rigidbody physics, and so nothing happens, nor does gravity happen.

The **blue cube** remained stationary throughout the simulation. This object was not equipped with a Rigidbody component and as such was incapable of movement via physics and was not moved by any scripts.

When the simulation began the **first red sphere** fell onto the teal plane under the force of gravity. It remained stationary until the orange sphere pushed it outwards in the negative direction on the x-axis. When the orange sphere reversed its direction it slightly pulled the sphere back with it.

The **second red sphere** remained motionless throughout the simulation. Although it collided with the orange sphere, it was unaffected because it did not contain a Rigidbody component.

The **green capsule** lays still the entire time. It has a rigidbody with gravity, so it should be registering collisions and falling, however it is intact with the teal plane, so it does not fall any further. Also, it touches the purple cube, but the purple cube does not have a rigidbody, and so no physics collision happens.

The **yellow cylinder** remained stationary throughout the simulation. Because a Rigidbody component was not connected to this object, it was unaffected by physics.

**White sphere one** remained motionless throughout the simulation. Although it collided with the orange sphere, it was unaffected because its Rigidbody component was set to Kinematic mode.

When the simulation began, **white sphere two** fell onto the teal plane under the force of gravity. It remained stationary until the orange sphere pushed it outwards in the positive direction on the x-axis. The sphere then collided with the black cube and pushed it.

When the simulation began, the **grey sphere** fell onto the teal plane where it remained stationary for the rest of the simulation. This was a result of the Rigidbody component attached to the sphere which was set to use gravity.

The **black sphere** falls, and falls though the teal plane at that. It has a rigidbody with gravity, and so it falls down. But, instead of colliding with the teal plane, it falls through it, because it has it’s collider removed, so it can’t actually collide with anything. Though it can do physics stuff.

The **yellow cube** rotated along each of the three coordinate axes with respect to the world at a rate of 30 degrees per second along the x-axis, 60 degrees per second along the y-axis, and 90 degrees per second along the z-axis. Because each rate of rotation is a multiple of 360, the rotation was a repeating pattern. We achieved this rotation by using a C# script which incrementally modified its rotation by calling the Rotate method of Transform with arguments equaling 30/50, 60/50, and 90/50 for each respective axis within the FixedUpdate method which is called 50 times per second.

The **black cube** remained stationary until the second white sphere collided with it, at which point it followed a straight directory roughly along its local x-axis. Because the cube was rotated 45 degrees on the z-axis, it traveled up and away from the scene. This behavior occurred because the Rigidbody component of the cube was not set to use gravity.