**Triggering System**

**Overview:**

The triggering system is part of the gizmoball game in which it connects triggers to actions. An action may only occur once a certain requirement has been satisfied – the ball colliding with another object that has been assigned a trigger. Once a collision has been detected, it will trigger an action, the actions can be simple state changing actions, for example changing the velocity of the ball. However, an option for the user would be to add more complicated triggers when in build mode, for example another gizmo in the game’s plane may have been assigned a trigger which will in turn activate another gizmo (example, flipper) to change its state.

**How it works:**

Gizmo’s would take the role as the observable objects which would allow the observers to be notified when the *collisionDetected* state becomes true. Upon a collision, the gizmo that the ball had collided with will notify the observers using the *notifyObservers()* and *setChanged()* which will then trigger some action that the user has assigned to that gizmo. The observes *update()* method would call upon methods like *activateFlipper* or *rotateGizmo.* It may also be possible for two triggers to be activated at the same time, to avoid this conflict, a possible solution would be for the first trigger to activate which will disregard ay other trigger from activating until the action that the specific trigger is assigned too has completed.

**Observable and Observer:**

To be able to tell if there has been a collision, we would need to track two things: the ball and the gizmo. Since the ball could collide with any given gizmo on the plane, each gizmo would need to have its own individual observer so that no two triggers from different gizmos are activated at the same time.