**Rope Bridge**

**Note: This interaction is not completely implemented/has some errors, see Implementation section**

The interaction consists of a rope bridge which the player is able to walk across using thumbstick locomotion. While moving across the height of their hands will be tracked and compared in order to simulate changing balance.

**Props**

* Bridge object
  + Solid railings added in place of those integrated with the bridge to improve performance
* Thumbstick locomotion adapted VR pawn

**Interaction Expectations**

The goal of the interaction is to motivate and test use of thumbstick locomotion as well as how users react to external effects on their perception of balance. The user needs simply to walk across the bridge using thumbstick locomotion and attempt to keep their hands level while doing so.

**Implementation**

The rope bridge blueprint was acquired from here: <https://forums.unrealengine.com/community/community-content-tools-and-tutorials/37580-free-procedural-rope-bridge> and modified to include a box collider used to determine when the player is on the bridge.

Thumbstick locomotion was implemented through the MotionControllerPawn blueprint by following this tutorial with some slight modifications: <https://www.youtube.com/watch?v=kGCS4rJYzoM>. On BeginPlay, the initial rotation of the pawn is saved for reorientation purposes and a reference to self is saved in order to be ignored by the hand controller line traces for height. Custom OnBegin and OnEnd overlap functions are also added for determining when the player is on the bridge. A capsule collider has also replaced the root of the pawn to handle collision.

Each tick, the CheckBalance custom event is called. Within CheckBalance, as long as the player is on the bridge, a line trace will be run for each motion controller from their position to the ground in order to determine their height. The height of each hand is then compared and as long as one is higher than the other, a roll/X-axis rotation is applied to the capsule in the appropriate direction.

**Issue Description:** While everything would appear to be implemented correctly when looking over code, the motion controller experiences unexpected behavior when on the bridge. First, while the the pawn is able to easily reach the center of the bridge, it will be unable to ascend the other side. I believe this is due to the collision used by the planks on the bridge. Changing the plank mesh between simple and complex collision did nothing to fix this. If you click on the bridge actor in the editor while Simulating, the planks will show a red box that does not match the rotation of the meshes they are attached to, which I assume to be their colliders. For some reason, despite use of a capsule collider for the VRpawn, you are unable to “roll” up and over the edges. I also attempted changing collision presets on the collider to fix this, to no avail. Second, I was unable to determine how to rotate the view of the camera along the X axis in addition to its rotation from the HMD. I attempted several implementations, including rotating the camera component, adding a parent to the camera to rotate, and simply rotating the capsule collider, which is the version I have left in the project. In this final version, although I am applying a rotation to the X-axis, the pawn only rotates around the Z-axis. One attempt to fix this involved locking rotation of the different axes of the capsule, with various undesirable results. Third, collisions with other dynamic actors seems to add a phantom torque force/rotation to the pawn, where in it will continue to rotate in one direction indefinitely with apparent cause within the pawn BP.

**Testing Guide**

**WARNING: This interaction may make the user susceptible to dizziness and loss of balance. Make sure that the area around you is clear of physical obstacles before attempting.**

* In addition to the standard locomotion method of teleportation and walking within your headset’s boundary (if applicable), you are able to move utilizing the motion controllers’ thumbsticks
  + The left stick controls front/back and sideways movement
  + The right stick controls rotation (this is in addition to rotation based upon the headset)
* When on the bridge, the motion controllers will begin to check their height against each other
* It is recommended that you hold your hands out at your sides while walking across the bridge
  + If you are using a headset with inside-out tracking, this may not be possible
    - In this case, hold your hands in view in front of you in order to prevent loss of tracking
* If one hand is higher than the other, the camera will rotate, simulating movement of the bridge
* Attempt to keep your hands level with each other as you walk across the bridge
* WARNING: The nature of the camera’s movement may induce dizziness/motion sickness. Be aware of how you feel while using it and stop use immediately if you become uncomfortable.

**Hologram Dodgeball**

**Note: This interaction has an error which appeared at the last minute (15 minutes before due time). When grabbing the dodgeballs with either controller, a phantom rotation, like that described in the rope bridge interaction, occurs, causing the pawn to rotate out of control indefinitely. I assume this has something to do with physics and the settings I’ve adjusted for the capsule collider on the VRpawn, but I have no real way of knowing - I changed so many things attempting to fix collisons for the bridge I forgot what was changed.**

This interaction consists of an arena containing a number of holograms in the form of dodgeballs which randomly float around within it. The player can grab the holograms, transforming them into physical objects in the scene which they may then interact with as they desire.

**Props**

* Dodgeball arena
* Dodgeballs with hologram and solid material versions
* Thumbstick locomotion adapted VR pawn

**Interaction Expectations**

The movement of the dodgeballs randomly throughout the arena motivates the player to use the thumbstick locomotion to move quickly to attempt to acquire one. Use of the hologram material with the balls helps to suspend disbelief due to the fact a real dodgeball can’t randomly float through the air. The balls likely could easily be combined with another blueprint like the tennis ball launcher to create a minigame that has the player actually attempt to dodge the balls that are launched at them. If not for challenges implementing my other interaction this is something I would have attempted.

**Implementation**

The arena functions by using a box collider as the “safe area” for dodgeballs. On beginplay, the arena will spawn a number of dodgeballs, determined by the MAX\_BALLS variable and placing them in a random position within the box collider. For an unknown reason this seems to be weighted towards the center of the box. The CurrentBalls variable is incremented and a reference to the ball stored in an array. At start the box collider is also given an OnComponentEndOverlap event wherein it will check for balls leaving the box through the arena’s door and destroy them. Every five seconds, the arena will spawn new balls, up to the maximum amount.

The HologramBal blueprint is adapted from the included basketball blueprint. Using public variables for a hologram material and a solid material, the ball will change to a solid form when picked up with a motion controller and gravity will be enabled. While the ball is in hologram form, the Move event is called every tick, wherein a force determined by a random unit vector is applied to the ball, causing it to float around the arena. The hologram material was created by following this tutorial: <https://www.youtube.com/watch?v=wIz6iyIOkys> and its texture acquired from: <https://www.filterforge.com/filters/13620-v6.html>

**Testing Guide**

* In addition to the standard locomotion method of teleportation and walking within your headset’s boundary (if applicable), you are able to move utilizing the motion controllers’ thumbsticks
  + The left stick controls front/back and sideways movement
  + The right stick controls rotation (this is in addition to rotation based upon the headset)
* You may walk (via thumbstick locomotion) or teleport into the arena
* Using the trigger on either controller, you may grab the dodgeballs floating around you
  + This will transform the dodgeball from a hologram to a solid, allowing you to throw, bounce, drop, etc. it however you wish
* You will not be able to remove the dodgeballs from the arena