Open the RC\_CarController Script.

* Inside the start function, add the commands:
* GetComponent<VRTK\_ControllerEvents>().StartMenuPressed += new ControllerInteractionEvent Handler(DoStartMenuPressed);
* GetComponent<VRTK\_ControllerEvents>().StartMenuReleased += new ControllerInteractionEventHandler(DoStartMenuReleased);

These commands will mean that the program will perform actions when the StartMenu Button is pressed (when the touchpad is clicked)

* Create two new functions:

private void DoStartMenuPressed(object sender, ControllerInteractionEventArgs e)

{

rcCarScript.SetStartMenu(e.buttonPressure);

}

private void DoStartMenuReleased(object sender, ControllerInteractionEventArgs e)

{

rcCarScript.SetStartMenu(0f);

}

* These functions will call functions from the RC\_Car Script and change the state of a variable.

Open the RC\_Car Script

* Inside the RC\_Car class, add the line:
  + private float startButton;
* This creates a float that will track the status of the startButton (whether it is pressed or not)
* Also inside the class, add the functions:

public void SetStartMenu(float data)

{

startButton = data;

}

* This function changes the status of the startButton float.
* Delete the functions OnTriggerStay and OnTriggerExit
* Find the Jump function and replace the code in it with:

Vector3 movement = transform.up \* jumpPower \* Time.deltaTime \* triggerAxis;

rb.MovePosition(rb.position + movement);

* Add the function:

public void Unjump()

{

Vector3 movement = transform.up \* jumpPower \* Time.deltaTime \* (-startButton);

rb.MovePosition(rb.position + movement);

}

* This function, as the name implies does the opposite of jumping (It adds a force in the negative direction as opposed to the positive.