

HeadJoystick Unity API

Created by: Abraham M. Hashemian
s.e.hashemian@gmail.com

Step 1 - HTC Vive Setup

Execute the SteamVR and run the “Room Setup” from its top-left corner menu, and then choose “Standing Only”. Put the Vive HMD on the center of the swivel-chair seat while the chair and the vive HMD are facing toward the forward physical direction when start using your unity project, and then calibrate the center of the Vive room setup. In this way, you will be sure that the swivel-chair is at the center of the vive visible area, and all the users start from the same physical direction with no need to change the initial direction within the code.

Step 2 - HeadJoystick Hardware Installation

1. It's recommended to use a swivel chair, which has both yaw rotation and backrest tilt. (Backrest's tilt supports the player's neck during backward movement). Considerations:
 - a. If you can't attach a Vive tracker (or right controller) to the chair backrest, please ask the user to simply seat straight while their head would be on the center of chair yaw rotation, and then start the project and press SPACE to move.
2. Attach Vive tracker to the exact center of the chair backrest using a track belt. As the below images show, tracker's LED should be toward down (perpendicular to the floor) and the tracker should be tilted back, otherwise, it's rotational values in Unity will change dramatically when it tilts to zero degrees. So if the chair backrest is perpendicular to the floor, you can use a piece of cardboard between the track belt and the chair's backrest to ensure it's tilted.



3. Turn on both controllers and wait until see them in the SteamVR and **THEN** turn on tracker or any controller you want to attach to the chair. Then you need to run the project to calibrate the tracker and then start locomotion using the below three steps:
 - a. After running the project on Unity, Sit on the chair. If the chair backrest is not tiltable, skip this step otherwise lean back on the chair backrest until you hear a Ding sound. Possible problems:

- i. Don't rotate the chair yaw while leaning back. Don't move it on the ground either, otherwise, in the Console window, a few numbers show up, which means that something is wrong and you need to do the whole calibration process once more.
- b. Wear the Vive headset, seat comfortably, and then press space to start locomotion i.e., walking, running, driving, or flying. The position of your head will be your zero-point when pressing SPACE. (You'll stop when your head comes back to this point). Considerations:
 - i. We usually ask people to touch their back to the chair backrest when pressing SPACE, because they can find their zero-point later easily using the physical force feedback of the chair backrest. However, this zero-point requires players to lean back if they want to go backward, which might not be easy for some people - due to more weight distribution on their lower body. These people could seat straight when pressing space, so they can go backward easily, but then they need to lean forward to go forward, which might be less comfortable for their back in the long term.

Step 3 - HeadJoystick Coding

Using HeadJoystick in a project requires a script - called HeadJoystick.cs, attached to the player's game object to move it. This script provides methods to check the current status of HeadJoystick calibration and control its locomotion.

- **HeadJoystick global setting variables:** Some of the public variables in HeadJoystick.cs can be used to change its settings:
 - *Speed Sensitivity* control HeadJoystick sensitivity, where its higher values cause more velocity for the player with fewer head movements. Therefore, reducing its sensitivity reduces the accuracy of HeadJoystick, but makes it easier for shorter users.
 - *Upward Speed Scale* multiplies *speedSensitivity* for vertical locomotion. It should be generally more than one because users usually have a shorter range for their vertical head movements compared to their horizontal head movements. Similar to speed sensitivity, you could reduce this variable for shorter people, although it reduces their vertical motion accuracy.
 - **Dead Zone Radius:** This public variable create an dead zone around zero-point, which makes it easier for the user to stop their locomotion, although they still have their natural head movements due to head tracking. You could put this public variable between 10-15 cm [0.1 .. 0.15], to makes it easier for the user to stop. Larger values of this variable require lots of head movements to move in the game.
- **HeadJoystick Methods:** To use HeadJoystick.cs script to move the player, you need to call its methods in your scripts. A sample locomotion script is attached to the player game object - called Locomotion. Unlike the HeadJoystick.cs script, which you should not change, feel free to change Locomotion.cs script based on your game requirements. To help you do this easier, here I explain it's details:

- **Start() methods:** You need to call `HeadJoystick.initialize ()` method in your locomotion script to initialize the HeadJoystick including reading its configuration file. This method requires two keyboard buttons to start the calibration process. The first button defines when the “Lean back” should be started (usually using `KeyCode.RightAlt`) and the second argument defines when the locomotion should be started (usually using `KeyCode.Space`).
- **Update() Methods:** In `Update()` method, you should call `HeadJoystick.calibrate()` method to calibrate HeadJoystick, and then when the calibration process is complete i.e., `HeadJoystick.isReady() == true`, you need to call `HeadJoystick.move()` to move the player. Details for each method:
 - **HeadJoystick.calibrate()** gets one optional boolean argument if you want to see the updated zero-point in the game window, to be sure that Vive tracker is calibrated successfully and is tracking correctly. If you call `calibrate(true)`, you should see a sphere at the center of the chair backseat tilt in the game window. If this sphere is in a wrong position (e.g., behind tracker), it means that something went wrong in the calibration process, and thus you need to replay the game. If the issue insists, close and open SteamVR, and then run the game.
 - **HeadJoystick.move()** gets three arguments (two mandatories and one optional) including:
 - *Horizontal speed limit:* positive values of this argument limits the maximum horizontal locomotion speed in meters/seconds. Negative values remove the velocity limit, which allows the user to locomote as fast as they can but increase the chance of motion sickness. Putting zero in this argument disables the horizontal locomotion and can be used as a brake.
 - *Vertical speed limit:* positive values of this argument limits the maximum vertical locomotion speed in meters/seconds. Negative values remove the velocity limit, which allows the user to locomote as fast as they can but increase the chance of motion sickness. Putting zero in this argument disables the vertical locomotion and turns HeadJoystick to a ground-based locomotion interface instead of a flying interface.
 - *Show tracker zero position:* This argument is optional and set it as true if you want to see the updated zero-point in the game window, to be sure that Vive tracker is calibrated successfully and is tracking correctly. If you set this variable to true, you should see a sphere at the center of the chair backseat tilt in the game window. If this sphere is in a wrong position (e.g., behind tracker), it means that something went wrong in the calibration process, and thus you need to replay the game. If the issue insists, close and open SteamVR, and then run the game.