**Player hierarchy**

The player character’s component hierarchy was inherited from UE4 VR template. The root component, as well as VR origin transform, represents center of playzone in reality. The camera attached to VR origin is mastered by VR head-mounted display, where player’s view is based. One pair of hands will be spawn at runtime and attached to VR origin, mastered by paired VR motion controllers. As is designed to imitate head stance, HUD is directly attached to camera to maintain relative transform with simplicity.

A mobile node is attached to VR origin. The node has its position and yaw imitating camera (i.e. HMD) while maintaining its relative pitch and roll. This particular design emphasizes player’s body position and orientation in reality, as reflected by the capsule collider attached to it.

**HUD**

Inspired by realistic head-up displays, HUD in the game simulates intuitive, minimalized holographic projection on the astronaut mask. As a feedback segment of human-computer interaction, the HUD provides visualized essential information on system state, including motion state, equipment status, and spatial awareness.

Similar to real-life HUD, pure green (#00BF00) was chosen as theme color of the HUD. Studies suggested that human eyes are most sensitive to green colors and are best at distinguishing among them. As possible adaptation to natural life, human eyes also appeared to be less exhausted staring at green than other colors.

content

An attempted was made in early development stage to deliver HUD as conventional widget, a summary of Unreal Engine 2D entities. The outcome appeared to be complete failure in visual deliverance, as HUD elements were rendered at zero depth of field (referred as DoF below) on solely left VR goggle. With reference to multiple sources, this phenomenon was explained due to following UE4 rendering logics:

1. Widgets in UE4 are always rendered at post-processing stage, implying they may never participate in 3D rendering.
2. The VR binoculus (double-eyed) displays were conventionally divided into 2 separate screens (one per VR lens), with screen UVs respectively mapped to U value of 0-1 on left screen and 1-2 on the right. (The similar behavior also applies to dual screen setup based on conventional displays.) The default post-processing material of widgets only renders to screen UV of 0-1, namely all contents are delivered on the leftmost display.

The principles above implies the incapability of 2D widgets on presenting HUD in VR mode. It is also suggested by the sources that all visual items must be rendered in 3D world to form focusable, stereo vision in head-mounted display.

The methodology was adopted in second attempt, where all HUD elements are projected on 3D surfaces with constant focal distance of approximately 60 meters, rendered occluding other world entities. This particular method, inspired by the game VTOL VR, eventually reached proposer’s expectation.

The rendering logics above were also utilized for monocular (single-eyed) rendering, inspired by early Apache Attack Helicopter’s monocular eyepiece. Phasmophobia, a thrill game, is also known for utilizing the mechanism for strengthening thrilling atmosphere. As a potential extension to human-computer interaction, players may choose to keep a single eye open to switch between HUD overlaid view and HUD-free view accordingly. Nonetheless, as being gradually replaced by binoculus display in real-life utility, monocular HUD is noticed for causing optical illusion and dizziness in stereo vision.

Jetpack is intended as primary approach player moves as an astronaut in the game. Similar to real-life astronaut jetpack, it provides 6 degree-of-freedom translational and rotation control by applying respectively linear and angular momentum to astronaut, with input by two hand controllers. In this game, differences are made where all controlling of the jetpack are assembled at a flight control stick located front left of player’s waist. This interaction model is supported by input set of Oculus motion controller, where sufficient buttons and input axes enabled rotation by rotating control stick, and locomotion using thumbsticks and buttons. The simplified interaction model also guarantees one spare hand for complex tasks.

Rotation control is simulated by rotation of player’s hand, with controlling logic similar to real-life aircraft control sticks. Yaw, pitch and roll are mapped to corresponding rotation actions.

Input set while holding flight control stick:

|  |  |
| --- | --- |
| Input | Action |
| Thumbstick Y | Move forward/backward |
| Thumbstick X | Move right/left |
| Y/B | Ascend |
| X/A | Descend |
| Trigger (axial) | Brake |
| Thumbstick press | Toggle rotation lock |

Grappling hook is intended as secondary approach that player maneuvers in environment. As being a populated concept in During development, the functionality was expanded to

|  |  |
| --- | --- |
| Input | Action |
| Trigger | Fire |
| Y/B | Switch fire mode |