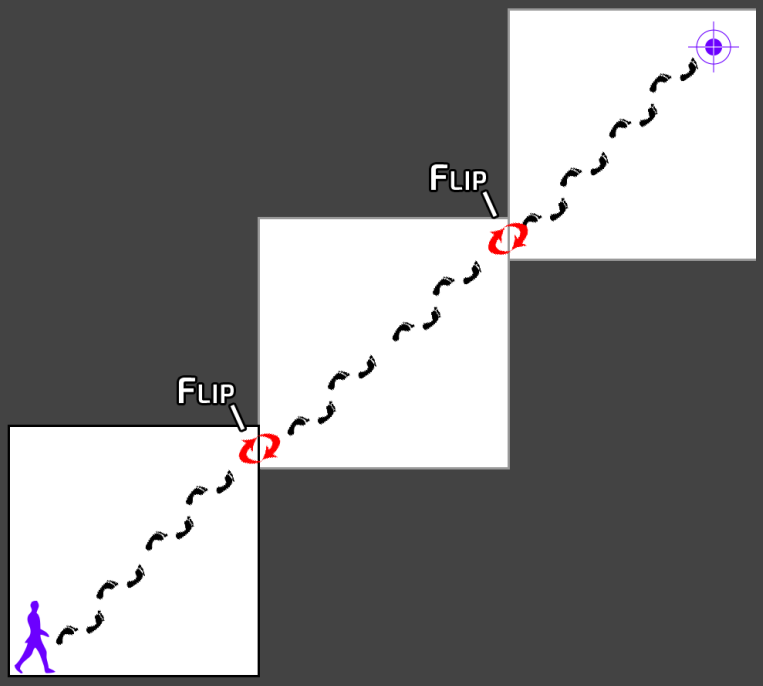
Player Rig Features:

- Device Setup Manager:
    - The Device Setup Manager component allows developers to set device-specific settings such as target framerate.
    - The component uses the player's controller to determine the current active device.
    - The Fixed Timestep is calibrated at runtime to align itself with the target framerate, ensuring that the ingame physics will behave appropriately, regardless of framerate.
    - There are also Quest-specific settings available such as Fixed Foveated Rendering (FFR).
    - This feature is present on all Player Rig prefabs.
- Body Presence:
    - Players will physically react to the world around them.
    - Players will collide with solid surfaces and fall off of ledges if walked past them.
    - Players can walk up and down sloped surfaces using the Step Up Easing Time on the "Player Body Controller" component to determine the smoothness of the raising and lowering movement.
    - Players can step over and on top of surfaces that are no greater than the Step Height on the "Player Body Controller" component.
    - Players will not be able to interact with objects, or use thumbstick movement if their head is colliding with a solid object (defined by the "Ground" layer by default).
    - Using this feature, players can be forced to physically crouch in order to traverse tight spaces for added immersion.

    - This feature is present on all Player Rig prefabs.
- Physics-based Hands:
    - Player hands will physically react to the world.
    - Objects that are subject to physics will be pushed by hands unless they are grabbing the object.
    - The “Physics Hands_Manager” component on each of the rig’s Direct Interactor hands will allow you to adjust how the hand is oriented, depending on the current device.
    - This feature is present on all Player Rig prefabs.
- Snap-Turn:
    - Pressing left, or right on the Right Thumbstick (or Touchpad on HTC Vive) will rotate the player 45 degrees (be default) in that direction.
    - The axis point of rotation is the player themselves rather than the center of the play-area.
    - The rotation amount can be tweaked in the “XR_Snap Turn” component on the “XR Rig”
    - This cannot be performed while climbing.
    - This feature is present on all Player Rig prefabs.
- Leaning:
    - Pressing the Primary Button on the Left controller (double-tap trigger on HTC Vive) will allow the player to walk through solid objects.
    - The player can only travel as far from their starting position as the “Lean Allowance” allows.
    - If the player’s head collides with an object the effect cancels, preventing walking through walls.
    - This feature is present on all Player Rig prefabs on the “PlayerBodyController.”

- Mirror-Step:
    - This movement method requires no controller input, freeing all buttons and triggers for other mechanics.
    - Mirror-Step uses a system of triggers that are calibrated to your headset's Guardian Boundary/Play Area.
    - By physically walking to the edge of your space, you will enter one of these triggers.
    - This trigger will flip your Guardian Boundary/Play Area 180° automatically, allowing the player to turn around and continue walking to their desired destination (see diagram below).



    - Due to the nature of this movement method, it is recommended only for large, roomscale experiences such as those that can be achieved using Quest, Quest 2, or using a VR backpack setup.
    - This feature is present in the "Player_Rig_MirrorStep" Prefab Variant.

- Teleportation:
    - Pressing the designated Teleport button, designated by the XR_Teleport component on the XR Rig, will perform a basic arc teleport.
    - The properties of the Teleport arc can be modified in the “Teleport_Ray_L” and “Teleport_Ray_R” GameObjects.
    - This feature is present in the “Player_Rig_Teleport” Prefab Variant.
- Thumbstick Walking:
    - Players can use the Thumbstick (or Touchpad on HTC Vive) to walk.
    - The walking direction can be switched between 4 different options:
        - Body Forward is dictated by the “Body Alignment System” component on the “PlayerBodyController” and takes the general direction of the controllers and head as a single forward direction. This allows players to walk in one direction, but still be able to look left and right.
        - Head Forward uses only the general forward direction of the head.
        - Left and Right Hand Forward takes those respective hands’ forward directions to steer the player.
    - Walk speed can be tweaked in the “XR_Movement_Controller” component on the “XR Rig”
    - This feature is present in the “Player_Rig_SimpleWalk” and “Player_Rig_Adventure” prefab variants.
    - Only surfaces with the “Ground” layer will register as valid surfaces for movement.

- Running:
    - While Thumbstick Walking (see above) players can rhythmically pump their arms to run.
    - The player's forward direction will shift from the specified forward direction to the forward direction of the player's gaze, making minor adjustments easier at high speed.
    - Running speed can be tweaked in the "XR_Movement_Controller" component on the "XR Rig"
    - This feature is present in the "Player_Rig_Adventure" prefab.
    - Only surfaces with the "Ground" layer will register as valid surfaces for movement.
- Jumping:
    - While grounded, players can swing one arm downward and, during the downward swing, press the Trigger on the swung controller. This will perform a jump.
    - The jump's properties can be adjusted in the "XR_Jump" component on the "XR Rig"
    - This feature is present in the "Player_Rig_Adventure" prefab.
- Arc-Jump:
    - While grounded, players can hold the button marked on the "XR_Arc_Launch" component on the "XR Rig" to begin aiming a launch.
    - Upon release, the player will launch into the air, following the arc.
    - Players can perform multiple mid-air jumps dictated by the Max Jump Count.
    - The jump's properties can be adjusted in the "XR_Arc_Launch" component on the "XR Rig"
    - This feature is present in the "Player_Rig_Arc_Launch" prefab.

- Climbing:
    - Using the Grip Triggers, players can grab onto walls and pull themselves in any direction.
    - Only climbable surfaces will be registered by this component (see "Climbing Prefab" below)
    - Climbing relies on the "Climbing Handle" prefab present in the "XR Rig"
    - The properties can be adjusted in the "XR_Climb" component on the "XR Rig"
    - This feature is present in the "Player_Rig_Adventure" prefab.

- Wall Jump:
    - While clinging to a surface, the player can perform a wall jump using the hand that is currently not holding onto the surface.
    - The player is then propelled upwards and forwards according to their forward gaze.
    - Wall jumps can only be performed while climbing (see above).
    - The jump's properties can be adjusted in the "XR_Wall_Jump" component on the "XR Rig"
    - This feature is present in the "Player_Rig_Adventure" prefab.
- Swimming:
    - While inside of a "WaterVolume" prefab, or inside of any Trigger Collider using the "Water" layer, players will enter a swimming state.
    - Players can freely move using the Thumbstick (or Touchpad on HTC Vive).
    - The player's direction is dictated by their forward gaze direction.
    - While swimming, players can swing one, or both hands backward to perform a swimming stroke, propelling them forward.
    - These properties can be adjusted in the "XR_Swimming_Controller" component on the "XR Rig"

    - This feature is present in the “Player_Rig_Adventure” prefab.
- Grappling Hook:
    - Players can aim at a valid “GrappleTarget_Prefab” or any collider with the “GrappleSurface” tag (by default) to be attached to it via a wire.
    - If you would like to have a Climbable surface (see Climbing above) to be a valid grappling surface as well, you will need to have an additional collider present. One with the “Climbable” tag, the other with the “GrappleSurface” tag.
    - The “GrappleSpawner” prefab will allow you to create grapple targets between two points.
    - Players can influence their swing using the Thumbstick (or Touchpad on HTC Vive).
    - The player must be grounded in order to perform a grapple (by default).
    - If a collidable object comes between the grapple point and the player, the connection will be severed (by default).
    - If the player’s height exceeds the height of the grapple point the connection will be severed (by default).
    - These properties can be adjusted in the “XR_Grapple_Swing” component as well as the “Get Grapple Targets” component on the “GrappleController.”
    - This feature is present in the “Player_Rig_Adventure” prefab.
- Scanner:
    - Using the Menu button, players can use the Scanner to find hidden objects.
    - Objects using the “ScannerVisible” Layer will be seen in the Scanner, but not by the player camera.
    - The Scanner relies on the “RevealCam” Camera component under the “WristGizmo_Parent” and sends its feed to the “HiddenCam” Render Texture. Custom materials can be made to utilize this Render Texture.

    - This feature is present in the “Player_Rig_Adventure” prefab as well as a modified version of the “Player_Rig_Standing” prefab variant present in the “XR_Scanner” example scene.
- Mini-Map:
    - Using the Menu button, players can use the Mini-Map to find important objects in an overhead view.
    - A “Minimap_Manager” prefab should be placed into the Scene, allowing developers to create Mini-Map targets with a custom icon and color.
    - The Mini-Map relies on the “MiniMap_Camera” Camera component and sends its feed to the “MiniMap” Render Texture. Custom materials can be made to utilize this Render Texture.
    - This feature is present in the “Player_Rig_Adventure” prefab as well as a modified version of the “Player_Rig_Standing” prefab variant present in the “XR_Minimap” example scene.

Trigger Volumes:

- Player Trigger:
    - This simple trigger will perform a Unity Event upon entering, or exiting the trigger.
    - This can be used to trigger environmental set-pieces, hazards, or anything that does not require conscious player interaction.
- Color Fade Trigger:
    - The player’s view will fade to a color at a speed dictated by the trigger.
    - The view will return to normal upon exiting the trigger.

- Checkpoint:
    - The default spawn location of the player will change upon entering this trigger.
    - Spawn locations are managed by the “Player_Spawn_Manager” component on the “XR Rig”
- “Death” (Respawn) Trigger:
    - Players that touch this trigger will respawn at the last Checkpoint (see above).
- Climb Up Trigger:
    - Players that grab this trigger will be automatically placed upright at the location that was grabbed within the trigger.
    - This can be used in instances where it may be otherwise difficult for a player to hoist themselves over a ledge to complete a vertical climb.
    - The prefab itself is only represented as a trigger with a simple highlight material. However, in practice, it might be appropriate to align this trigger to a more obvious, physical object within your scene.
- Climb Up Trigger (Manual Placement):
    - This variant of the Climb Up Trigger (see above) allows you to specify the location that the player arrives at when the trigger is grabbed.
    - This type of manual teleportation can also be applied to other use-cases, not just climbing to the top of a structure.
- Water Volume:
    - This prefab contains multiple trigger volumes and particle effects that, together, create a rectangular block of water.
    - Players that enter this area will enter the Swimming state.
    - Physics-based objects that enter the area will be subject to (somewhat) realistic water physics.

- Teleport Surface:
    - This surface can be teleported onto using the Teleport movement method (see Teleportation above).
- Spring:
    - This simple object will bounce the player into the air, according to the component's Aim Direction.
    - The Spring can either maintain, or negate the player's incoming momentum.

Interactable Objects:

- Offset Grabbable Objects:
    - These are grabbable objects that use an extension of the base "XR Grab Interactable" component.
    - These allow players to grab onto them at the point of contact rather than snapping to a specified point.
    - This is recommended for most basic grab interactions, as this method works well with the physics-based hands of the player (see above).
    - In addition, there is also a prefab set to utilize the Velocity Tracked movement method. This allows the object to collide with the environment while grabbed.
- Multi-Hand Grabbable Objects:
    - These are grabbable objects that use an extension of the base "XR Grab Interactable" component.
    - Like the Offset Grabbable (see above), this grabbing method allows the object to be grabbed at the point of contact. Unlike the original, however, it can be grabbed with two hands at once and manipulated appropriately with both hands.

    - This grab method is ideal for larger objects such as large crates, or long objects such as a broom handle.
- Snap Drop Zone:
    - Similar in function to the *XR Interaction Toolkit's* Socket Interactor, this will hold onto a single object that is manually placed within it. However, this version offers new features and added stability.
    - Snap Drop Zones can be attached to other objects, allowing for nested inventory spaces such as, for example, a briefcase containing multiple Zones being placed into a single Zone itself.
    - Only grabbable objects with the "StorableItems" layer are accepted by default.
    - Other, nearby Zones' colliders can be placed within the "List Of Other Snap Drop Zones To Disable" list. This can be used to manage overlapping Zones if they are close together, creating a "first come, first served" relationship.
    - Additionally, Snap Drop Zones can utilize the "Look For Matching Name" check to look for objects that contain a particular string in its name.
    - The "Look For Matching Name" check can be used to solve puzzles as demonstrated in the "XR_Interactables" example scene.
- Puzzle Manager Component:
    - The "Puzzle Manager" component can be attached to a GameObject and then given a list of "XR_Interactable_Lock" components. When all of these locks are unlocked, the Puzzle Complete Unity Event will fire.
    - A previously complete Puzzle can be undone if one, or more of the locks are locked again.
    - The "XR_Interactable_Lock" component is attached to several interactive prefabs already.

- Item Checker:
    - The “ItemChecker_Group” will look for any grabbable object that is manually placed within it and check the name of the GameObject. If the name contains the desired string of characters, then the appropriate Unity Event will fire.
    - The “Require Manual Placement” check can be used to force players to manually place objects into the trigger. Objects that are thrown in from afar will not be registered.
    - The included prefab contains an “XR_Interactable_Lock” component.
- Button:
    - The “XR_Button” will react to the player’s hand, pressing after the threshold has been met.
    - The included prefab contains an “XR_Interactable_Lock” component.
- Dial:
    - The “XR_Dial” must be grabbed and twisted. The “XR_Dial” component can be tweaked to create different types of dials.
    - The “Dial_Value_Manager” component will dictate which values can be used to fire the Desired Angle Reached Unity Event using the Desired Values list.
    - Multiple values can be placed into this list.
    - The included prefab contains an “XR_Interactable_Lock” component.
- Lever:
    - The “XR_Lever” must be grabbed and pulled in either direction.
    - Max and Min Unity Events can be fired.
    - The included prefab contains an “XR_Interactable_Lock” component.
- Slider:
    - The “XR_Slider” relies on a Configurable Joint component. It is recommended that the mesh representing the slider’s length and the Joint’s Linear Limit be identical.

    - The “XR_Slider_Manager” component will dictate which values can be used to fire the On Secret Reached Unity Event using the Desired Values list.
    - The included prefab contains an “XR_Interactable_Lock” component.
- Drawer:
    - The “XR_Drawer” prefab uses the “XR_Joint_Grabber” component. This component creates an intentional disconnect between the grabbed handle and the physics-based object itself.
    - This allows the object to feel heavier, or lighter, depending on the RigidBody properties of the object.
    - Usage Example: The developer wishes to have a drawer that is rusted and difficult to open. The player’s physical pull on the object is influenced by the physical properties of the drawer.
- Door:
    - The “XR_Door” prefab uses the “XR_Joint_Grabber” component. This component creates an intentional disconnect between the grabbed handle and the physics-based object itself.
    - This allows the object to feel heavier, or lighter, depending on the RigidBody properties of the object.
    - Usage Example: The developer wishes to have a door that is rusted and difficult to open. The player’s physical pull on the object is influenced by the physical properties of the door.
- Climbing Prefab:
    - Adding the “Climbing_Prefab” to a scene and then populating its list of colliders will make those colliders climbable (see Climbing above), indicated by a cyan-colored outline.

- Additional Climbable Prefabs:
    - The “XR_Zipline” is a unique prefab that combines a climbable sphere with a Configurable Joint. Similar to the Slider (see above), it should have its mesh the same length as the Joint’s Linear Limit.
    - The “XR_Climbing_Moving_Block” and “XR_Climbing_Pendulum” will loop between two Transforms.
- UI Example:
    - The “XR_UI_Example” can be interacted with using the controller’s Trigger.
    - Each Player Rig has a “Ray Interactor_L” and “Ray Interactor_R” GameObject that, by default, is short in length.
    - The short length of these rays makes it so the player must interact with the UI elements closely with their hands.

Multi-Scene Interactions:

- Key Items:
    - The “XR_KeyItem” is similar to the “XR_OffsetGrabbable” (see Offset Grabbable Objects above) but it contains the “Key Item_Object” component.
    - Key Item Objects that have been previously selected by the player are marked as “found.”
    - “Found” objects will automatically destroy duplicate items upon entering a scene, ensuring that only one of itself can exist, even if the same scene is left and then revisited.
    - The “KeyItem_Manager” prefab should be placed in the same scene as the Player Rig.

    - “Found” objects will attach themselves to an available slot in the “Lost and Found” (see below) if they are a certain distance from the player, dictated by the Respawn Distance.
- Lost and Found:
    - The “Lost_And_Found” prefab should be placed in the same scene as the Player Rig, ensuring its persistence in each scene.
    - The Lost and Found will retrieve lost Key Items and place them in an available Snap Drop Zone (see Snap Drop Zone above).
- The Additive Scene Manager:
    - The Additive Scene Manager is located under Scene Manager > Additive Scene Manager.
    - This tool allows developers to easily load scenes additively to create a seamless loading environment between scenes.
    - Follow the directions in the “XR_StartingScene” scene for more information.
