

1. MovementControlEventChannel.cs

- Type:** ScriptableObject Event Channel.
- Namespace:** `ProjectDoors`
- Serialized Fields:** None.
- Public Properties:** `Instance` : Singleton instance for global access.
- Public Events:** `OnStopMovement` : Event triggered to stop movement.
`OnResumeMovement` : Event triggered to resume movement.
- Public Methods:** `RaiseStopMovement()` : Invokes `OnStopMovement` event for all subscribed listeners.
`RaiseResumeMovement()` : Invokes `OnResumeMovement` event for all subscribed listeners.
- Purpose:** Acts as a global event channel for stopping and resuming movement in the game, ensuring consistent movement control across different components.

2. DoorEventChannel.cs

- Type:** ScriptableObject Event Channel.
- Namespace:** `ProjectDoors`
- Serialized Fields:** None.
- Public Properties:** `IsDoorOpen` : Tracks the current state of the door (open or closed).
- Public Events:** `OnOpenRequest` : Event triggered when a door open request is made, passing `userPosition` and `force` flag.
`OnCloseRequest` : Event triggered when a door close request is made, passing the `force` flag.
- Public Methods:** `RaiseOpen(Vector3 userPosition, bool force)` : Sets `IsDoorOpen` to true and invokes `OnOpenRequest` .
`RaiseClose(bool force)` : Sets `IsDoorOpen` to false and invokes `OnCloseRequest` .
- Purpose:** Manages door interaction events, providing a centralized system for triggering door operations and tracking door states.

3. PlayerActionsEditor.cs

- Type:** Custom Unity Editor Script.
- Namespace:** `ProjectDoors`
- Serialized Fields:** None.
- Overrides:** `OnInspectorGUI()` : Draws the inspector interface for `PlayerActions`, displaying input bindings and providing buttons to rebind or reset them.
- Private Methods:** `StartRebinding(InputAction action, int bindingIndex)` : Initiates an interactive rebinding process, excluding mouse controls.
`SaveBindingOverride(InputAction action)` : Saves current input bindings to `PlayerPrefs` .
`ResetBindings(InputAction action)` : Deletes saved bindings from `PlayerPrefs` and resets them to defaults.
- Purpose:** Enhances the Unity Editor by allowing developers to view, modify, and reset input bindings for `PlayerActions` within the inspector interface.

4. InteractionEventChannel.cs

- Type:** ScriptableObject Event Channel.
- Namespace:** `ProjectDoors`
- Serialized Fields:** None.
- Public Properties:** `Instance` : Singleton instance for global access.
- Public Events:** `OnInteractRequest` : Event triggered for interaction requests, passing the `position` .
- Public Methods:** `RaiseInteractRequest(Vector3 position, bool isOpen, Door door)` : Invokes `OnInteractRequest` event for all subscribed listeners.
- Purpose:** Serves as a global event channel for player interactions, facilitating interaction logic between the player and objects like doors.

5. RandomWanderAI.cs

- Type:** MonoBehaviour Script for AI.
- Namespace:** `ProjectDoors`
- Serialized Fields:** `wanderRadius`, `gizmoColor`, `waitTime`, `walkSpeed`, `runSpeed`, `stoppingDistance`, `AnimatorParameter`, `pathRecheckInterval`, `stuckCheckDuration`.
- Private Fields:** `NavMeshAgent agent`, `Animator animator`, `Vector3 wanderAreaCenter`, `bool isMoving`, `float timer`, `Vector3 lastPosition`, `float pathRecheckTimer`.
- Lifecycle Methods:** `Start()` : Initializes the AI's NavMeshAgent, Animator, and wander area.
`Update()` : Handles AI movement, stuck detection, path recalculations, and animation updates.
- Public Methods:** `RandomNavSphere(Vector3 origin, float dist, int layermask)` : Generates a random valid NavMesh position.
- Private Methods:** `CheckAndRecalculatePath()`, `FindBetterPath()`, `ChooseNewPath()`.
- Purpose:** Implements a wandering AI agent with dynamic path recalculations, stuck detection, and animation management.

6. DoorHierarchyIcons.cs

- Type:** Editor Utility Script.
- Namespace:** `ProjectDoors`
- Serialized Fields:** None.
- Static Fields:** `pivotIcon`, `knobIcon` : Loaded textures for pivot and knob icons.
- Static Methods:** `OnHierarchyGUI(int instanceID, Rect selectionRect)` : Draws icons in the Unity Hierarchy for doors with active pivot adjustments or unlocked knobs.
- Private Methods:** `DrawIconAndText(Rect pairRect, Texture2D icon, string label, GUIStyle style)` : Renders an icon with text in the Hierarchy window.
- Purpose:** Adds visual indicators in the Unity Hierarchy for doors, showing pivot adjustment mode or unlocked knob states, enhancing editor usability.

7. DoorLever.cs

- Type:** MonoBehaviour Script for Door Lever Interaction.
- Namespace:** `ProjectDoors`
- Serialized Fields:** `controlledDoor`, `doorEventChannel`, `activationDistance`, `playerTag`, `raycastIgnoreLayers`, `textYOffset`, `objectToRotate`, `leverRotationAngle`, `leverRotationSpeed`, `matchRotationSpeedToSound`, `leverSounds`.
- Private Fields:** `AudioSource audioSource`, `Camera mainCamera`, `TextMeshPro interactionText`, `bool isPlayerWithinRange`, `Quaternion initialRotation`, `Quaternion targetRotation`, `Coroutine rotationCoroutine`, `bool isLeverToggled`, `bool isInitialized`.
- Public Methods:** `SetControlledDoor(Door door)`, `AssignDoor(Door door)`.
- Private Methods:** `Awake()`, `Start()`, `Update()`, `LateUpdate()`, `OnValidate()`, `HandleProximityInteraction()`, `HandleMouseClickedInteraction()`, `ToggleLever()`, `RotateLever(System.Action onComplete)`.
- Purpose:** Manages player interactions with a door lever, handling proximity detection, mouse clicks, lever rotation, and sound playback, while automatically assigning the controlled door and event channel.

8. CharacterReplacer.cs

- Type:** MonoBehaviour Script for Player Character Replacement.
- Namespace:** `ProjectDoors`
- Serialized Fields:** `newCharacterPrefab`, `characterReplaced`.
- Private Fields:** `GameObject currentCharacter`, `GameObject newCharacter`.
- Public Methods:** `ReplaceCharacter()` : Replaces the player character with a new prefab, ensuring all necessary components are added.
- Private Methods:** `EnsureComponent<T>`, `EnsureRigidbodyWithConstraints(GameObject obj)`, `AddPlayerInputWithDelay()`.
- Purpose:** Provides an editor utility to replace the player character, ensuring that essential components like `CapsuleCollider`, `PlayerActions`, `ClickAgentController`, and `PlayerInput` are added.

9. PlayerInputDefaultInitializer.cs

- Type:** Editor Utility Script.
- Namespace:** `ProjectDoors`
- Serialized Fields:** None.
- Static Methods:** `OnComponentAdded(Component component)`, `CheckForNewPlayerInputs()`, `ConfigurePlayerInput(PlayerInput playerInput)`, `FindInputActionAsset(string assetName)`, `LogAvailableInputAssets()`, `IsPlayerInputConfigured(PlayerInput playerInput)`.
- Purpose:** Automatically configures `PlayerInput` components added via the Unity Editor, ensuring that input actions, UI input modules, and cameras are correctly assigned, and logs available input assets for debugging.

10. ClickAgentController.cs

- Type:** MonoBehaviour Script for Player Movement Control.
- Namespace:** `ProjectDoors`
- Serialized Fields:** `MovementControlEventChannel movementControlChannel`.
- Private Fields:** `NavMeshAgent agent`, `CharacterController characterController`, `bool isCharacterControllerStopped`.
- Public Methods:** `StopAgentMovement()`, `ResumeAgentMovement()`, `IsMovementAllowed()`.
- Purpose:** Controls player movement using either `NavMeshAgent` or `CharacterController`, allowing for stopping and resuming movement through `MovementControlEventChannel`.

11. ProximityStopper.cs

- Type:** MonoBehaviour Script for Proximity-Based Player Movement Control.
- Namespace:** `ProjectDoors`
- Serialized Fields:** `ProximityEventChannel proximityEventChannel`, `float stopRange`, `float releaseDelay`.
- Private Fields:** `Transform playerTransform`, `ClickAgentController clickAgentController`, `PlayerController playerController`, `bool isPlayerInRange`, `bool isPlayerStoppedCompletely`, `bool playerHasClickedOnObject`, `Vector3 storedPlayerPosition`, `GameObject player`, `Camera mainCamera`.
- Lifecycle Methods:** `OnValidate()`, `Start()`, `Update()`.
- Private Methods:** `HandleProximity()`, `HandlePlayerInput()`, `StopPlayerMovement()`, `ResumePlayerMovement()`, `ReleasePlayerAfterDelay()`, `IsClickOnThisObject()`, `GetDistanceToPlayer()`, `ArePlayerComponentsInitialized()`, `InitializePlayerReferences()`, `OnDrawGizmosSelected()`.
- Purpose:** Controls player movement based on proximity, ensuring that the player stops within a specified range and resumes after exiting the range or releasing a click. Integrates with `ClickAgentController` and `ProximityEventChannel` for seamless movement control.

12. DoorCreationWizard.cs

- Type:** Editor Utility Window Script.
- Namespace:** `ProjectDoors`
- Serialized Fields:** `wizardStep`, `currentStep`, `Editor characterReplacerEditor`, `GameObject tempCharacterGO`, `bool characterReplacerExists`, `Editor doorReplacerEditor`, `GameObject tempDoorGO`, `bool doorReplacerExists`, `Vector2 replacerScrollPos`.
- Public Methods:** `OpenWindow()`, `OnEnable()`, `OnDisable()`, `OnGUI()`.
- Private Methods:** `DrawWelcomeScreen()`, `DrawPlayerQuestionScreen()`, `DrawCharacterReplacerScreen()`, `DrawDoorReplacerScreen()`, `PrepareDoorReplacer()`.
- Purpose:** Provides an interactive wizard interface for setting up player characters and doors in Unity, simplifying the setup process by automating component assignments and configurations.

13. ProximityEventChannel.cs

- Type:** ScriptableObject Event Channel.
- Namespace:** `ProjectDoors`
- Serialized Fields:** None.
- Public Events:** `OnEnterProximity`, `OnExitProximity`.
- Public Methods:** `RaiseEnterProximity(Vector3 position)`, `RaiseExitProximity(Vector3 position)`.
- Additional Methods:** `HasEnterProximityListeners()`, `HasExitProximityListeners()`.
- Purpose:** Manages proximity-based events, allowing objects to trigger actions when entering or exiting a specified range.

14. DoorTrigger.cs

- Type:** MonoBehaviour Script for Door Trigger Interaction.
- Namespace:** `ProjectDoors`
- Serialized Fields:** `Door controlledDoor`, `DoorEventChannel doorEventChannel`, `string[] allowedTags`, `bool autoOpen`, `bool autoClose`, `float closeDelay`, `UnityEvent onTriggerEnterEvent`, `UnityEvent onTriggerExitEvent`.
- Private Fields:** `int allowedTagCount`.
- Public Methods:** `SetControlledDoor(Door door)`.
- Private Methods:** `OnValidate()`, `EnsureCollider()`, `OnTriggerEnter(Collider other)`, `OnTriggerExit(Collider other)`, `AutoCloseDoor()`.
- Purpose:** Controls door interactions triggered by objects entering or exiting a trigger zone, supporting auto-opening, auto-closing, and collider configuration.

15. LayeredNavMeshObstacle.cs

- Type:** MonoBehaviour Script for Dynamic NavMesh Obstacle Control.
- Namespace:** `ProjectDoors`
- Serialized Fields:** `LayerMask affectedLayer`, `LayerMask priorityLayer`, `float activationRange`.
- Private Fields:** `NavMeshObstacle navMeshObstacle`, `Transform obstacleTransform`.
- Lifecycle Methods:** `Awake()`, `Update()`.
- Private Methods:** `UpdateObstacleState()`, `IsAnyObjectInRange()`, `IsPriorityObjectInRange()`, `EnableObstacle()`, `DisableObstacle()`, `OnDrawGizmosSelected()`.
- Purpose:** Dynamically enables or disables NavMesh obstacles based on the presence of objects in specific layers, with priority overrides for certain layers.

16. DoorReplacer.cs

- Type:** MonoBehaviour Script for Door Replacement.
- Namespace:** `ProjectDoors`
- Serialized Fields:** `GameObject newDoorPrefab`, `bool doorReplaced`.
- Private Fields:** `GameObject currentDoor`, `GameObject newDoor`.
- Public Methods:** `ReplaceDoor()`.
- Private Methods:** `GetUniqueName(string baseName)`, `EnsureComponent<T>(GameObject obj)`, `EnsureRigidbodyWithConstraints(GameObject obj)`, `AddPlayerInputWithDelay()`.
- Purpose:** Provides an editor tool to replace doors, ensuring all necessary components (colliders, proximity stoppers, audio sources, NavMesh obstacles) are added for proper functionality.

17. PlayerActions.cs

- Type:** MonoBehaviour Script for Player Interactions.
- Namespace:** `ProjectDoors`
- Serialized Fields:** `InteractionEventChannel interactionEventChannel`, `string usableTextTag`, `LayerMask useLayers`, `InputActionReference interactActionReference`, `string openText`, `string closeText`, `float yOffset`, `float xOffset`, `float zOffset`, `float activationDistance`, `bool enableLogging`.
- Private Fields:** `Camera camera`, `ClickAgentController agentController`, `TextMeshPro useText`.
- Public Methods:** `GetInteractActionReference()`, `HandleProximityInteraction()`, `HandleKeyInteraction()`, `HandleMouseInteraction()`, `OnUse()`.
- Private Methods:** `AssignInputActionReference()`, `FindInputActionAsset(string assetName)`, `LoadBindingOverrides()`, `HandleDoorInteraction(Door door)`, `FindInteractiveObjectByTag<T>(string tag)`, `Log(string message)`, `LogWarning(string message)`, `LogError(string message)`, `LogOnce(ref bool hasLogged, string message)`, `ResetConditionalLogging()`.
- Purpose:** Manages player interactions with doors using proximity, keys, and mouse clicks, with integration into `ClickAgentController` and `InteractionEventChannel`.

18. PlayerController.cs

- Type:** MonoBehaviour Script for Player Movement Control.
- Namespace:** `ProjectDoors`
- Serialized Fields:** `float agentSpeed`, `LayerMask terrainLayerMask`, `float moveDelay`.
- Private Fields:** `Camera mainCamera`, `NavMeshAgent agent`, `bool isMoving`, `bool isMoveInitiated`.
- Lifecycle Methods:** `Start()`, `Update()`, `LateUpdate()`.
- Private Methods:** `StartMovingAfterDelay(Vector3 destination)`.
- Purpose:** Controls player movement using `NavMeshAgent`, handling input-driven movement and manual agent rotation.

19. Door.cs

- Type:** MonoBehaviour Script for Door Control.
- Namespace:** `ProjectDoors`
- Serialized Fields:** `Pivot Settings:` `string pivotName`, `Transform pivot`, `bool EnableGridSnapping`, `float GridSize`, `bool limitPivotToBounds`.
`Door Settings:` `GameObject doorObject`, `PositionPreset positionPreset`, `GeometryAlignment objectGeometryAlignment`, `float openDelay`, `float closeDelay`, `GameObject lever`, `bool isLeverControlled`, `bool isDualDoor`, `Door secondDoor`, `bool disableColliderWhileMoving`, `bool disableNavMeshObstacle`.
`Rotation and Sliding:` `Axis rotationAxis`, `bool isRotatingDoor`, `float Speed`, `float RotationAmount`, `float ForwardDirection`, `Vector3 SlideDirection`, `float SlideAmount`, `bool rotateWhileSliding`, `Vector3 rotationAxisDuringSlide`, `float rotationSpeedDuringSlide`, `bool invertRotationOnClose`, `bool allowCustomRotationSpeed`.
`Audio and Animation:` `AnimationCurve doorAnimationOnOpen`, `AnimationCurve doorAnimationOnClose`, `List<AudioClip> openSounds`, `List<AudioClip> closeSounds`, `bool matchSpeedToSound`, `float volume`.
`Mouse Settings:` `Texture2D customCursor`, `Vector2 cursorHotspot`.
`Interactivity:` `List<ComponentConfig> componentConfigs`, `List<ObjectConfig> objectConfigs`, `UnityEvent onDoorOpen`, `UnityEvent onDoorClose`.
`Bounds:` `Vector3 autoMinBounds`, `Vector3 autoMaxBounds`, `Vector3 manualMinBounds`, `Vector3 manualMaxBounds`, `bool useManualBounds`, `float xLeeway`, `Color BoundsGizmoColor`.
`Knobs:` `bool limitKnobToBounds`, `bool knobEnableGridSnapping`, `float knobGridSize`.
`Highlighting:` `Material highlightMaterial`, `float brightnessFactor`, `float fadeDuration`, `bool loopBrightnessEffect`.
- Public Methods:** `Open(Vector3 userPosition)`, `Close()`, `ForceOpen(Vector3 userPosition)`, `ForceClose()`, `RotateAroundPivot(Vector3 eulerAngles)`, `SlideRelativePivot(Vector3 direction, float amount)`, `ApplyPositionPreset()`, `UpdateDoorBounds()`, `FindPivot()`, `EnsureAttachedObject()`, `GetEventChannel()`, `GetPivot()`, `SetPivotName(string newPivotName)`, `ToggleLeverControl(bool leverControlState)`.
- Purpose:** Comprehensive door control system supporting pivot adjustments, rotation, sliding, animations, sounds, interactivity, bounds management, and visual feedback.

20. DoorEditor.cs

- Type:** Custom Unity Editor Script.
- Namespace:** `ProjectDoors`
- Serialized Fields:** `GameObject leverInstance`, `int selectedTab`, `Texture2D generalIcon`, `Texture2D hingeIcon`, `Texture2D rotationsIcon`, `Texture2D slidingsIcon`, `Texture2D animationsIcon`, `Texture2D soundsIcon`, `Texture2D mouseIcon`, `Texture2D interactivityIcon`, `Texture2D knobIcon`, `Texture2D highlightIcon`, `GUIContent[] tabContentsFull`, `GUIContent[] tabContentsIcon`, `float[] tabTextProgress`, `double lastUpdateTime`, `bool pendingEventChannelCreation`, `Door doorPendingChannel`, `Dictionary<GameObject, (Vector3 position, Quaternion rotation)> knobStateMemory`.
- Public Methods:** `OnInspectorGUI()`.
- Private Methods:** `OnEnable()`, `OnDisable()`, `OnUndoRedoPerformed()`, `ValidateSerializedObjects()`, `UpdateTabTextAnimation()`, `DrawTabsWithWrap(int currentSelectedTab)`, `CreateNewDoorEventChannel(Door door)`, `DrawGeneralSettings(Door door)`, `DrawHingeSettings(Door door)`, `DrawRotationSettings(Door door)`, `DrawSlidingSettings(Door door)`, `DrawAnimationSettings(Door door)`, `DrawSoundSettings()`, `DrawMouseSettings(Door door)`, `DrawInteractivitySettings()`, `DrawKnobSettings()`, `DrawMaterialSettings()`.
- Purpose:** Provides an enhanced Unity Editor interface for editing all aspects of doors, including pivot settings, rotation, sliding, animations, sounds, mouse interactions, interactivity, knobs, and material highlighting, with animated tab transitions for improved user experience.