Note the names of those who wrote the documentation next to each section, so that we know who wrote what

1. Visual Occlusion Toggle

Visual Occlusion toggle.png

1.1 Main goal

- Enable or disable the visual occlusion entirely
- When occlusion is disabled, pressing any actuation method will have no effect
- Only when occlusion is enabled will the actuation methods function properly

1.2 Variables

- isActivated
    - Explanation: Determine if the fade screen should be created and added to the viewport or removed completely
    - Type: Boolean
    - Default Value: False
- Keyboard 9
    - Explanation: Turn on visual occlusion via keyboard
    - Type: Button input
    - Values: 0 or 1

1.3 Branch

- Executable input: Keyboard 9
- Condition: `isActivated`
- True: `Remove All Widgets of Class`
    - Reason: `isActivated` is set to True, fade screen is already active on the viewport, and input is set to remove every instance of the object
- False: `Create Fade Screen Widget`
    - Reason: `isActivated` is set to False, fade screen has not been created
    - `Fade Screen`: A black screen component to cover the screen

1.4 Functions

- Remove All Widgets of Class
    - Implemented: From UE4 library
    - Explanation: Remove any instance of fade screen class from the viewport

- Create Fade Screen Widget
    - Implemented: From UE4 library
    - Explanation: Create object from `fadescreen.c` found in Yehorivka blueprints
    - Found on:
- Add to Viewport
    - Implemented: From UE4 library
    - Explanation: Add object Fade Screen Widget to Viewport

2. Visual Occlusion Adaptive

Visual Occlusion Adaptive pt.1.png

Visual Occlusion Adaptive pt.2.png

2.1 Main goal

- Temporarily disable visual occlusion to allow the user to see the road, utilizing the button inputs outlined below
    - We are experimenting with various actuation methods to determine the most effective one.
- The duration for which the occlusion is disabled is adaptive
    - It compensates based on previous durations to ensure balance
    - For instance, if a previous occlusion lasted only 0.49 seconds, the next will extend to 0.51 seconds.

2.2 Variables

- Button inputs
    - GenericUSBController Button 2 – Steering Wheel Button B
    - GenericUSBController Button 4 – Steering Wheel Button Y
    - GenericUSBController Button 5 – Steering Wheel Left Paddle
    - GenericUSBController Button 6 – Steering Wheel Right Paddle
    - Left Ctrl – Mouse left button
    - Num 0 to 6, +, -, /, * – Numpad inputs from 0 to 6, +, -, /, *
    - Backspace – Numpad Backspace
    - Tab – Numpad Tab
- isActivated
    - Explanation: Set from “Visual Occlusion Toggle”
    - Type: Boolean
    - Range: True / False
- Handle:
    - Explanation: `FTimerHandler` used to reference an existing timer
        - `FTimerHandler`: Manage timed events, acts as a handle or a reference to a timer that is set up in the Blueprint
    - Type: Timer Handle Structure
    - Implemented: UE4 Blueprint
- Start
    - Explanation: Set when occlusion is temporarily deactivated

    - Type: Float
    - Set: Func: Get Game Time in Seconds
- End
    - Explanation: Set when occlusion is reactivated
    - Type: Float
    - Set: Func: `Get Game Time in Seconds`
- Pre Duration
    - Explanation: Duration in seconds of previous visual occlusion deactivated
    - Type: Float
- CustomEvent Start timer
    - Explanation: user-defined event to execute a specific set of actions, only triggered by an event-driven approach instead of being called directly
    - Use Case: Event triggered by timer set in the Set Timer by event
      Runs once PreDuration seconds have passed after occlusion was turned off

2.3 Branch 1

- Executable input: Button Input
- Condition: Manually set to either true or false
- True: Use the adaptive version of the visual occlusion
- False: Use the non-adaptive version of the visual occlusion

2.4 Branch 2

- Executable input: Branch 1 True Input
- Condition: `isActivated`
- True: `Remove All Widgets of Class`
    - Reason: (Fade screen is on), remove instance from view and start timers to calculate adaptation of visual occlusion durations depending on previous values to be consistently set/ averaged to 0.5 seconds
- False: `Create Fade Screen Widget`
    - Reason: (Fade screen is not on), just remove any instance of the object from view

2.5 Branch 3

- Executable input: Output from func "`Remove all Widgets from Class`"
- Condition: `isValid` function
- True: `Clear and Invalidate Timer by Handle`

    - Reason: `isValid` is set to True, variable already exists. Remove it to prevent corruption
- False: `Set Timer By Event`
    - Reason: `isValid` is set to False, no previous variable stored, just continue to timer handling

2.6 Functions

- Remove All Widgets of Class
    - Implemented: From UE4 library
    - Explanation: Remove any instance of fade screen class from the viewport
- Create Fade Screen Widget
    - Implemented: From UE4 library
    - Explanation: Create object from `fadescreen.c` found in Yehorivka blueprints
- Add to Viewport
    - Implemented: From UE4 library
    - Explanation: Add object Fade Screen Widget to Viewport
- isValid
    - Input: `Handle`
      Timer Handle Structure
    - Explanation: it checks if the `Handle` is currently associated with a running timer
    - Output: True or False
        - If timer is stopped, cleared, paused or never set up, `isValid` is false
- Clear and Invalidate Time Handler
    - Input: Boolean
      Branch 2 Output
    - Explanation: Remove any previously stored value in handler variable

2.7 Block Comments

- Calculate and print time for temporary occlusion
    - Variable inputs: `End`, `Start`, `Pre Duration`
    - Sets: `Pre Duration` = `Pre Duration` - (`End` - `Start`)
    - Prints the (`End` - `Start`), converted from float to string
- Occlusion turned off, so start timer
    - Pre Duration

        - Explanation: Set to 0.5 seconds
        - Type: Float
        - Additional: Connected to Time Input from Func: `Set Timer by Event`
    - Set Time by event
        - Inputs:
            - Event: Called when the timer finishes
            - Time: Duration of the timer (Var: `Pre Duration`)
        - Output: handle representing the timer, for pausing/stopping the time
    - SET (Timer Handle)
        - Explanation: Stores timer's handle into a variable called
    - Get Game Time in Seconds
        - Explanation: Retrieves current in-game time (s)
    - SET (Start Time)
        - Explanation: Stores the current game time into a Start variable
- Add black screen to view port
    - Re-create the black screen from the Fade Screen Class, add it to the viewport

3. Visual Occlusion Non-adaptive

Visual Occlusion Non-adaptive.png

3.1 Main goal

- "Temporarily" turn off visual occlusion for the user to see the road
- How long the occlusion stays off is "non-adaptive", meaning occlusion will always stay off for only 0.5 seconds

3.2 Variables

- Button inputs (Inputs taken from inputs in the adaptive section)
    - GenericUSBController Button 2 – Steering Wheel Button B
    - GenericUSBController Button 4 – Steering Wheel Button Y
    - GenericUSBController Button 5 – Steering Wheel Left Paddle
    - GenericUSBController Button 6 – Steering Wheel Right Paddle
    - Left Ctrl – Mouse left button
    - Num 0 to 6, +, -, /, * – Numpad inputs from 0 to 6, +, -, /, *
    - Backspace – Numpad Backspace
    - Tab – Numpad Tab
- isActivated
    - Explanation: Set from "Visual Occlusion Toggle"
    - Type: Boolean
    - Range: True / False
- Handler:
    - Explanation: `FTimerHandler` used to reference an existing timer
    - Type: Timer Handle Structure
    - Implemented: UE4 Blueprint
- Start
    - Explanation: Set when occlusion is temporarily deactivated
    - Type: Float
    - Set: Func: Get Game Time in Seconds
- End
    - Explanation: Set when occlusion is reactivated
    - Type: Float
    - Set: Func: Get Game Time in Seconds
- Pre Duration

    - Explanation: Duration in seconds of previous temporarily deactivated visual occlusion to track what the duration of the next iteration should be to maintain 0.5 second average
    - Type: Float
- CustomEvent Start timer
    - Explanation: user-defined event to execute a specific set of actions, only triggered by an event-driven approach instead of being called directly
    - Use Case: Event triggered by timer set in the Set Timer by event
        Runs once PreDuration seconds have passed after occlusion was turned off

3.3 Branch 1

- Executable input: Adaptive's Branch 1 True Input
- Condition: `isActivated`
- True: Remove All Widgets of Class
    - Reason: (Fade screen is on), remove instance from view and start timers to calculate adaptation of visual occlusion durations depending on previous values to be consistently set/ averaged to 0.5 seconds
- False:Remove All Widgets of Class
    - Reason: Just in case, remove all widgets to ensure that even if error occurs, road will not be blocked
        Will not lead anywhere

3.4 Branch 2

- Executable input: From previous node, a result of isValid node
    Checks if the timer handle exists
- Condition: `isValid`
- True: Executes "Clear and Invalidate Timer by Handle"
    - Result: Stops the timer and invalidates the handle
- False: Does nothing

3.5 Functions

- Remove All Widgets of Class
    - Implemented: From UE4 library
    - Explanation: Remove any instance of fade screen class from the viewport
- Create Fade Screen Widget
    - Implemented: From UE4 library

    - Explanation: Create object from `fadescreen.c` found in Yehorivka blueprints
- Add to Viewport
    - Implemented: From UE4 library
    - Explanation: Add object Fade Screen Widget to Viewport
- isValid
    - Input: Handle
        Timer Handle Structure
    - Explanation: the handle of the timer to check validity of
- Clear and Invalidate Time Handler
    - Input: Boolean
        Branch 2 Output
    - Explanation: Remove any previously stored value in handler variable

3.7 Block Comments

- Calculate duration of occlusion
    - Variable inputs: `End`, `Start`
    - Prints the (`End` - `Start`), converted from float to string
- Set start of when occlusion was turned off
    - Variable Inputs: True from clear and invalidate timer by handle, `Branch 2 False`
    - Event: `CustomEvent Start Timer`
    - Time: PreDuration
    - Return Value: Resets Handle variable
    - Sets: the Start in game time (s) of when occlusion is turned off
- Set ‘end’ for time in game when occlusion stops, create another black screen
    - Variable Inputs: Trigger from `Custom Event`
    - Output: Sets end of occlusion being off from value taken in Get Game Time in Seconds
        Create a widget from of Fade Screen Widget
        Add this new widget to the viewport