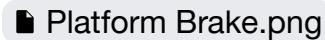


1. Brake



1.1 Variables

- MoveForward
 - Explanation: is the representation of how much the pedal is pressed. If multiple inputs are being pressed, their moveForward values are added (if W and S are pressed, MoveForward = 0)
 - The example above is true if S is mapped to brake and not handbrake, handbrake trumps accelerator.
 - Type: Float
 - Values: range of values from -1 to 1
 - -1 to 0 indicates the brake pedal is being pressed.
 - 0 means that no pedal is pressed or both are being pressed (unless the exception above)
 - And 0 to 1 indicates the accelerator is being pressed.
- ForwardSpeed
 - Explanation: The current forward speed of the vehicle.
 - Type: Float
 - Values: negative infinity to infinity (technically, because the car can move at any speed)
 - A negative value indicated moving backward.
 - The range of values is more likely max speed backward to max speed forward, but those values are variable
- Brake
 - Explanation: This is a Variable that determines if the brake is activated or not (if on, then the variable has the value of true).
 - Type: Boolean
 - Values of either true or false

1.2 Logic of turning on the brake

- a. If MoveForward is negative, the brake is being pressed; therefore, no matter the sign of ForwardSpeed, if moveForward happens to be negative, the brake should turn on.

2. Engine Volume Set (Written by Luka)



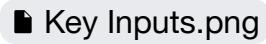
2.1 Variables

- Throttle
 - Explanation: Represents the amount of throttle being applied or how far the accelerator is being pressed (1 correlates to fully pressed)
 - Type: Float
 - Values: values from 0 to 1, inclusive
- Engine Sound
 - Explanation: The audio object corresponds to the sound of the engine when the car is moving and not idle
 - Type: Audio Object Reference
 - Values: Sound file of the engine
- ForwardSpeed
 - Explanation: The current forward speed of the vehicle.
 - Type: Float
 - Values: negative infinity to infinity (technically, because the car can move at any speed)
 - A negative value indicated moving backward.
 - The range of values is more likely max speed backward to max speed forwards, but those values are variable.

2.2 Flow/Logic

- If the product of the throttle and ForwardSpeed are negative, then the car is attempting to move in the opposite direction of motion, so then the engine will not produce a noise.
 - This is not how it is in real life, but just how the code works! (very rare scenario)
- Therefore, if the product is negative, multiply the throttle by zero so the engine sound multiplier is zero and the engine doesn't make noise.
- Else if the product is positive, multiply the throttle by .45 and add .55 to create a multiplier that will determine the volume of the engine sound.
 - Volume Multiplier is then set by using function **Set Volume Multiplier**

3. Key Inputs (Written by Luka)



3.1 Variables

- Space Bar
 - Explanation: When pressed, activate the handbrake.
 - Type: Button Input
 - Value: 0 or 1
- Escape
 - Explanation: When pressed, quit the game.
 - Type: Button Input
 - Value: 0 or 1
- R
 - Explanation: When pressed, load the Vehicle_demo_map
 - Type: Button Input
 - Value: 0 or 1

4. Radio (Written by Selay)

 Radio

4.1 Main goal

- Turn on/off the radio

4.2 Variables

- Num 8
 - Explanation: Turn on the radio via the numpad
 - Type: Button input
 - Values: 0 or 1
- (Keyboard) 8
 - Explanation: Turn on the radio via keyboard
 - Type: Button input
 - Values: 0 or 1

4.3 Object References

- Experiment Music
 - Explanation: Read the value of the variable Experiment Music
 - Type: Audio Component Object Reference
 - Values: Sound of the radio

4.4 Flip Flop

- Alternates between A and B outputs, starting with A

4.5 Functions

- Play
 - Start time: 0.0
 - Explanation: Start the radio sound playing on the audio component
- Stop
 - Explanation: Stop the radio sound playing on the audio component

5. Stop/reverse Lights (Written by Selay)

Stop/reverse lights

5.1 Main goal

- Turn on/off the reverse lights

5.2 Variables

- Brake
 - Explanation: Read the value of the variable brake
 - Type: TODO
 - Values: TODO

5.3 Object References

- Truck
 - Explanation: Read the value of the variable truck
 - Type: TODO
 - Values: TODO
- Simple Wheeled Vehicle Movement
 - Explanation: Read the value of the Simple Wheeled Vehicle Movement Component Object Reference
 - Type: Simple Wheeled Vehicle Movement Component Object Reference

5.4 Functions

- Get Forward Speed
 - Explanation: Takes the forward speed of the vehicle from the Wheeled Vehicle Movement Component Object Reference
- Set Scalar Parameter Value on Materials
 - Explanation: Sets all occurrences of Scalar Material Parameters with ParameterName in the set of materials of the SkeletalMesh ParameterValue
 - Parameter Target: Mesh component Object Reference
 - Parameter Name 1: Light_stop
 - Parameter Name 2: Light_reverse
 - Parameter Value: Float

5.5 Operations

- Dot (●)
 - Explanation: Converts the bool brake value to a float
- Less than (<)
 - Explanation: Returns true if the forward speed is less than -5, meaning that the vehicle needs to turn its reverse lights on

6. Shift Gears (Written by Luka)

Shift Gears

6.1 Variables

- Current Gear
 - Explanation: Holds the number of the current gear.
 - Type: int
 - Values: 1 to 5
- RPM
 - Explanation: Holds the value of the current amount of RPMs.
 - Type: Float
 - Values: from 0 to max RPM of the truck.
- ForwardSpeed
 - Explanation: The current forward speed of the vehicle.
 - Type: Float
 - Values: negative infinity to infinity (technically, because the car can move at any speed)
 - A negative value indicated moving backward.
 - The range of values is more likely max speed backward to max speed forward, but those values are variable.
- Simple Wheeled Vehicle Movement
 - Explanation: An object reference that references the object containing most of the properties of the movement of the truck.
 - Type: Object reference
 - Values: In this case, a float, which is ForwardSpeed.

6.2 Branches/Logic

- Branch 1
 - Executable input: Set Vector Parameter function from the Exhaust Control section.
 - Condition: If the current gear is not gear 5 AND the RPMs are greater than 3100, then true, else false.
 - True: shift the car up one gear (add 1 to Current Gear), making sure to clamp the Current Gear Value between 1 and 5 as the max gear is 5 and min gear is 1. Also, set the RPMs to 3070 and execute Branch 2.
 - False: Execute Branch 2.
- Branch 2

- Executable input: Either the false branch of Branch 1 or the setting of the RPMs to 3070 from the true branch.
 - Condition: If the RPMs are less than 2000, then true, else false.
 - True: shift the car down one gear (subtract 1 to Current Gear), making sure to clamp the Current Gear Value between 1 and 5 as the maximum gear is 5 and the minimum gear is 1. Also, execute Branch 3.
 - False: Execute Branch 3.
- Branch 3:
 - Executable input: Either the downshifting of Current Gear from the true branch of Branch 2 or the false branch of Branch 2.
 - Condition: If ForwardSpeed is less than -0.2, then true, else false.
 - True: shift the car down one gear (subtract 1 to Current Gear) and execute the branch in Gear Ratio and RPM.
 - False: Execute the branch in the Gear Ratio and RPM.

7. Exhaust Control (Written by Luka)

Exhaust Control

7.1 Variables

- Exhaust
 - Explanation: The variable represents the particle system of the Exhaust smoke.
 - Type: Particle System (Object/component?).
 - Values: ???
- RPM
 - Explanation: Holds the value of the current amount of RPMs.
 - Type: Float
 - Values: from 0 to max RPM of the truck.

7.2 Functions

- Set Float Parameter
 - Executable Input: after the volume multiplier is set to a new value (Set Volume Multiplier Function is called in Engine Volume Set).
 - Explanation: Takes the volume multiplier calculated during Engine Volume Set and sets it as the rate of the Exhaust object.
 - Target: Exhaust.
 - Parameter 1: Rate
 - Parameter Value: Volume multiplier from Engine Volume Set (see documentation for that section to determine the exact value).
- Set Vector Parameter
 - Executable Input: After the Set Float Parameter Function has run, call this function.
 - Explanation: Takes a vector with random X and Y components and a z component that's determined from the RPM and calls it the Velocity vector and sets it as the vector parameter of Exhaust.
 - Target: Exhaust
 - Parameter 1: Velo
 - Parameter 1 Value: The X and Y components of the vector are random floats between -100 and 100, while the Z component is calculated by dividing the RPM by 30 and adding 250.
 - Explanation of Value: When the car is moving faster (AKA more RPM), the car will produce more exhaust, so the velocity

vector of Z is going to be higher. X and Y are random variables to simulate the smoke being shot out in random directions instead of in one direction.

8. Gear Ratio and RPM (Written by Luka)

Gear Ratio and RPM

8.1 Variables

- ForwardSpeed
 - Explanation: The current forward speed of the vehicle.
 - Type: Float
 - Values: negative infinity to infinity (technically, because the car can move at any speed)
 - A negative value indicated moving backward.
 - The range of values is more likely max speed backward to max speed forward, but those values are variable.
- Current Gear
 - Explanation: Holds the number of the current gear.
 - Type: int
 - Values: 1 to 5
- Current Gear Ratio
 - Explanation: Tells the current gear ratio of the car; when the car is in lower-numbered gears, it needs a higher torque and, therefore, a higher gear ratio.
 - Type: Float
 - Value: -3.0 to 1.4^4
 - NEED TO FURTHER CHECK, NOT ALL VALUES WITHIN THE RANGE ARE POSSIBLE, Mainly because Current Gear is an int from 1 to 5 and not a range of values (so potentially 5-6 different values)
- RPM
 - Explanation: Holds the value of the current amount of RPMs.
 - Type: Float
 - Values: from 0 to max RPM of the truck.

8.2 Functions

- F Interp To
 -

8.3 Branches/Logic

- Branch 1

- Executable Input: When Branch 3 from Shift Gears executes, this Branch will execute as well.
- Condition: If ForwardSpeed multiplied by 408.399994 (Wheel RPS) and by 6 (Shaft RPS) is greater than or equal to zero, then true, else false.
 - True would mean the car is stationary or moving forward; false means you're moving backward.
- True: Set the Current Gear Ratio to a value determined by the following steps:
 - Take 5 minus Current Gear and turn into a float, and then take that value and raise 1.4 to that power, and set that answer to Current Gear Ratio.
 - Example: We are in gear 3, so the Current Gear Ratio is now 1.4^2
- False: Set Current Gear Ratio to -3.0
- After Branch 1:
 - After Branch 1 paths, no matter if evaluated to true or false, Set the RPM to a value determined by the process below:
 - Multiply the ForwardSpeed expression from before $(ForwardSpeed * Wheel RPS * Shaft RPS)$ by the RPM and 60.
 - Then use FInterp To to smoothly change the current RPM to the value calculated in the previous step, incrementing it towards that value by a time change determined by the World properties.
 - The value from FInterp To is then, however, bounded between 600 and 3500 RPMs, so you cannot go below or above this.

9. Drive Torque (Written by Luka)

 Apply Wheel Torque

 Drive Torque

9.1 Variables

- ForwardSpeed
 - Explanation: The current forward speed of the vehicle.
 - Type: Float
 - Values: negative infinity to infinity (technically, because the car can move at any speed)
 - A negative value indicated moving backward.
 - The range of values is more likely max speed backward to max speed forward, but those values are variable.
- Drive Torque
 - Explanation: The torque produced by the engine is measured at the wheel (Torque is lost in the transmission and crank).
 - Type: Float
 - Value: 0 to the max torque at the wheels.
- Engine Torque
 - Explanation: The torque produced by the engine is measured at the engine.
 - Type: Float
 - Value: 0 to max torque produced by the engine.
- RPM
 - Explanation: Holds the value of the current amount of RPMs.
 - Type: Float
 - Values: from 0 to max RPM of the truck.
- Throttle
 - Explanation: Represents the amount of throttle being applied or how far the accelerator is being pressed (1 correlates to fully pressed)
 - Type: Float
 - Values: values from 0 to 1, inclusive.
- Current Gear Ratio
 - Explanation: Tells the current gear ratio of the car; when the car is in lower-numbered gears, it needs a higher torque and, therefore, a higher gear ratio.
 - Type: Float

- Value: -3.0 to 1.4^4
 - NEED TO FURTHER CHECK, NOT ALL VALUES WITHIN THE RANGE ARE POSSIBLE, Mainly because Current Gear is an int from 1 to 5 and not a range of values (so potentially 5-6 different values)
- Engine Power (%)
- Explanation: The percent of the engine power that's being used (100% → full power)
 - Type: Float
 - Value: 0 → 100

9.2 Functions

- Apply Wheel Torque
 - Executable Input: This function is executed when the Drive Torque has its value set.
 - Explanation: Set the appropriate brake torque and drive torque on to the vehicle Depending on if the brake pedal is pressed or not (if it is then apply brake torque to wheels, if not then apply drive torque to wheels).
 - Input/Output: Has none
 - If the brake pedal is pressed (Brake Section):
 - Take the Brake boolean variable/value and convert it to a float, multiply it by 5000, and then apply that product as units of torque to the brakes of each wheel on the truck (therefore, if the brake isn't pressed, 0 torque is applied).
 - Similarly, take the Hand Brake boolean value/variable and turn it to a float, multiply it by 8000, and add that as units of torque to the back 4 wheels of the truck (so back wheels will have more brake torque if both hand brake and brake are activated).
 - If the brake pedal is not pressed (Drive Torque Section):
 - Take the Brake boolean and apply the NOT operator on it and then convert that value to a float. Multiply that by the current Drive Torque and then apply that value to be the new Drive Torque on each wheel of the truck.
 - So if the Brake is pressed, drive torque on the wheels becomes zero; if the Brake isn't pressed, drive torque on the wheels will be what's calculated in the drive torque code.
- Sign (float)

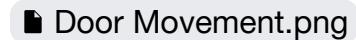
- Explanation: Takes a float and returns a number corresponding to the sign of the float.
- Value: -1 if the input is negative, 0 if the input is zero, and 1 if the input is positive.

9.3 Branches/Logic

- Branch 1
 - Executable Input: After the RPM value has been changed at the end of Gear Ratio and RPM, execute this branch.
 - Condition: If RPM is greater than 3000, then true, else false.
 - True: Set the Engine Torque to a value based on the following steps:
 - Take the RPM and subtract it from 4000, and then divide it by 1000.
 - At the same time, multiply the value of the Throttle variable by 600 and then multiply that product by the value calculated in the previous step involving the RPM.
 - Set this value to be the new Engine Torque.
 - False: Set the Engine Torque to a value based on the following steps:
 - Set the Engine Torque to be the value of the Throttle variable multiplied by 600.
- After Branch 1
 - After Branch 1, the Value of the Drive Torque is determined to be put onto the wheels. It first calculates 4 different components/values and adds them together and then multiplies that sum by a value to determine the Drive torque.
 - Engine Torque Component
 - Multiply the values of Engine Torque and Engine Power by each other and then divide by 100 to get the amount of Engine torque currently being produced by the engine.
 - Wheel Friction Component
 - Take the sign of FowardSpeed by using Sign (float) and multiply that by -200.
 - If the car is moving forward, AKA positive ForwardSpeed, then the friction because of the wheels must be negative.
 - Air Resistance Component
 - Take the value of ForwardSpeed and square it, and then divide it by 16,000.

- Multiply that previously calculated value by the sign of FowardSpeed by using Sign (float).
- Engine Friction Component
 - Take the value of the Current Gear Ratio and RPM and multiply them by each other, and then multiply that product by -200 and multiply by 0.00025 (engine friction factors).
- Now add up all of the four previously calculated components and multiply by the float value determined by the boolean value of the condition RPM < 3500.
 - If RPM is less than 3500, multiply the sum of the components by 1, else, multiply the sum by 0.
- Finally, set that product to be the new value of Drive Torque.
- After setting the new value of Drive Torque, execute and call the Torque function.

10. Door Movement (Written by Jared)



10.1 Variables

- Doors Open Amount
 - Explanation: “amount” the door is open, with the largest value being 1 and smallest being 0
 - Type: float
 - Value: 0, 1
- Doors Open
 - Explanation: Tracks whether the door is currently open
 - Type: Boolean Variable
 - Values: True (door is open), False (door is closed)
 - Flow logic: The operation “not” is implemented on the variable to invert the current door state. Then, the `Doors Open` variable is set to be open by the `Door Open` custom event.

10.2 Functions

- Lerp
 - This takes in values from the Door Open Amount and the Doors open to determine how to open the door; If the Doors Open Amount is 1 and Doors Open is 0, it will smoothly close the door using the alpha value using interpolation.

10.3 Branches/Logic

- Smooth opening of doors
 - Using the interpolation from lerp and the differences in values of A and B, over a various amount of frames, a value of a smaller/larger Doors Open Amount will be set, which calls to the Door Open event. The blueprint is also targeted to help make a smooth animation for the door closing/opening, facilitating as a way to move the doors according to whether the door control requires it to be open or not.
 - NOTE: This likely will have to be slightly different for different vehicles, although this is not necessarily a priority for the simulator as a whole

11. Light switch in-game (Written by Selay)

 Lights switch in game

11.1 Variables

- L
 - Explanation: Tracks whether the lights are on (true) or off (false)
 - Type: Boolean
 - Values: 0 or 1
- Lights
 - Explanation: Triggered when the "L" key is pressed
 - Type: Input Action Event
 - Values: Pressed or released

11.2 Custom Events

- Light Switch (Blue)
 - Label: RELIABLE Replicated To All (if server)
 - Explanation: Triggers light switching across all instances
 - Target: Self, which is the vehicle
- Light switch (Red)
 - Type: Multicast
 - Label: Executes On All
 - Explanation: Called on all clients to trigger the actual light logic and visuals
- Light Switch On Server (Blue)
 - Type: Custom Event (Client to Server)
 - Explanation: Sends a light toggle request to the server
- Light switch on server (Red)
 - Type: Custom Event (Server Only)
 - Explanation: Executes logic server-side when called from "Light Switch On Server"

11.3 Flow Control

- Switch has authority
 - Explanation: Checks if the current machine has authority
 - Outputs:
 - Authority: If this instance is the server.
 - Remote: If this instance is a client.

11.4 Functions

- Set Scalar Parameter Value on Materials
 - Explanation: Visually changes the material, in this case glows the lights when on. We are modifying the materials applied to the truck mesh.
 - Target: Truck (Mesh component)
 - Parameter Name: Lights
 - Parameter Value: 0 (off) or 1 (on), based on Lights boolean
- Set visibility
 - Explanation: Sets visibility for each light component
 - Target: Scene components such as the Dash lights and the Spot lights
 - Inputs:
 - New Visibility: Boolean from Lights
 - Propagate to Children: Set to false

11.5 Scene Components

- Dash lights:
 - Description: The actual lights attached to the vehicle
 - Usage: Passed into Set Visibility to turn them on/off
- Spot lights:
 - Description: The actual lights attached to the vehicle
 - Usage: Passed into Set Visibility to turn them on/off

11.6 Mesh Component

- Truck:
 - Explanation: Refers to the skeletal mesh or static mesh of the military truck in the Blueprint

11.7 Logic

- The player presses “L”
- Switch Has Authority checks:
 - If server:
 - Call “Light Switch” directly
 - If client:
 - Call “Light Switch On Server”, which triggers “Light switch on server” on the server, then calls “Light Switch”
- “Light Switch” multicasts “Light switch” to all clients

- “Light switch”:
 - Toggles Lights variable between true or false
 - Sets material light state using “Set Scalar Parameter Value”
 - Toggles visibility of all individual light components

12. Door open control (Written by Selay)

Door Open Control

12.1 Variables

- Enter
 - Explanation: Triggered when the "Enter" key is pressed
 - Type: Input Action Event
 - Values: Pressed or released
- Doors Open
 - Explanation: Tracks whether the door is currently open
 - Type: Boolean Variable
 - Values: True (door is open), False (door is closed)
 - Flow logic: The operation “not” is implemented on the variable to invert the current door state. Then, the `Doors Open` variable is set to be open by the `Door Open` custom event.

12.2 Custom Events

- Door Open (Red)
 - Implemented: From UE4 library
 - Explanation: The server triggers this event to open the door
- Door Open On Server (Red)
 - Implemented: From UE4 library
 - Explanation: Called by the client and sent reliably to the server as a part of the Door Open function while letting the client request something from the server
- Door Open (Blue)
 - Target: The current self, which is the military truck object
 - Explanation: Gets called to open the door when triggered by the server
 - Label: RELIABLE Replicated To All (if server)
- Door Open On Server (Blue)
 - Target: The current self, which is the military truck object
 - Explanation: When a client requests the action of opening a door from the server
 - Label: "RELIABLE Replicated To Server (if owning client)"

12.3 Flow Control

- Switch has authority

- Explanation: Checks if the current machine has authority
- Outputs:
 - Authority: Executes if run on the server
 - Remote: Executes if run on a client

13. Handle Camera Switching (Written by Jared Xin)

Handle Camera Switching

13.1 Variables

- C key
 - Explanation: This reads any input on the keyboard switch under c for the computer running Unreal Engine. Once pressed, the rest of the code is able to run. This is primarily mapped as a toggle switch for changing the camera's perspective.
 - Type: input
 - Values: Pressed or Released
- In Car Camera

13.2 Logic/Branches

- C input
 - Once pressed, the camera switch function is called, which switches the camera
 - SET In Car Camera updates the variable to the new state depending on the new state:
 - If True: Switches to In-Car camera view
 - If False: Switches to Military Track view