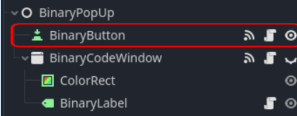
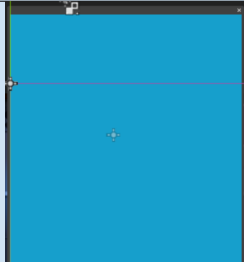
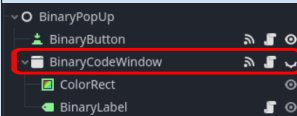
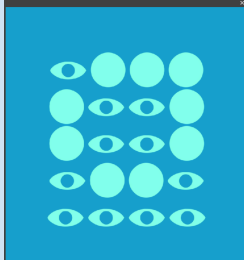
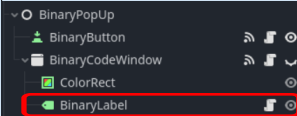
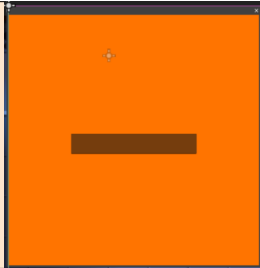
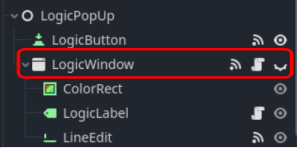

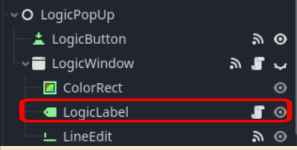
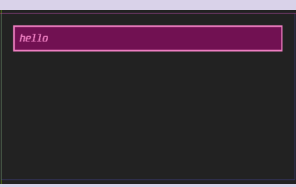

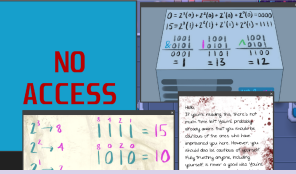
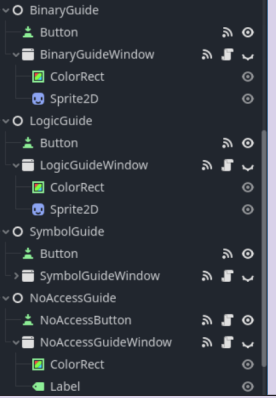
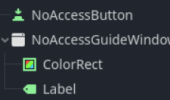



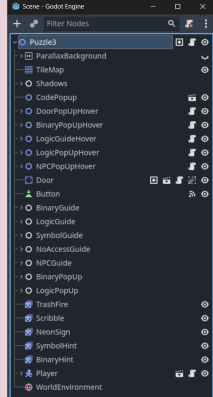


Script	Scene/Node	Purpose	Scene Picture	Hierarchy Picture	
global.gd (Autoload)	N/A	<p>Functions used in Level 3:</p> <p>logic_puzzle_solved: a global flag that indicates if the logic/bitwise operation puzzle has been solved. Initially set to false on game start.</p> <p>func set_logic_question() -> String: generates and sets a random logic/bitwise operation question to be displayed (eg: (3 & 2) ^ (5 7))</p> <p>func get_logic_question() -> String: gets last randomly generated logic question</p> <p>func solve_logic_question() -> int: solves logic question to be compared to user input</p> <p>func get_logic_result() -> int: gets last solved logic question</p> <p>func logic_puzzle_flag_set_true(): sets the logic_puzzle_solved flag to true.</p> <p>func logic_puzzle_flag_set_false(): sets the logic_puzzle_solved flag to false.</p>	N/A	N/A	Autoloads
AudioManager.gd	AudioManager.tscn (Autoload)	<p>Manages UI sounds for Puzzle 3 and for "(Return to Main Menu)" button that appears in the top left corner of every puzzle.</p> <p>A global scene and script is needed for scenes with tilemaps that are larger than the viewport (blue box in 2D scenes). Sounds can't be heard if a player moves outside of the viewport in play mode.</p>	N/A		
BinaryButton.gd	BinaryButton (Button Node)	On ready, if the logic_puzzle_solved flag is set to true, it ensures that the button that displays the binary puzzle to the player is visible in the scene. If the logic puzzle has not been solved, then the button is hidden. Therefore, the player will not be able to see the binary puzzle without first solving the logic puzzle.	N/A (invisible in scene since modulate alpha is set to 0)		Binary Puzzle
BinaryCodeWindow.gd	BinaryCodeWindow (Window Node)	<p>On ready, the BinaryCodeWindow is hidden and can only be viewed if the BinaryButton is clicked. Signals from BinaryButton are connected here:</p> <p>func_on_binary_button_pressed(): if the logic puzzle has been solved, then the BinaryCodeWindow is set to being shown</p> <p>func_on_close_requested(): BinaryCodeWindow is closed</p>			
BinaryLabel.gd	BinaryLabel (Label Node)	<p>On ready, the get_passcode() function in global.gd is accessed to obtain the 5-digit passcode that unlocks the door that leads to level 4.</p> <p>func_convert_to_binary(number): converts 5-digit passcode to 5 separate binary numbers</p> <p>The 0s are converted to "u"s and the 1s to "c"s, which show up as eyes and circles on the label due to a custom font variation being used for the label's font style. The image to the right is an example of what the BinaryLabel looks like in action.</p>			

LogicWindow.gd	LogicWindow (Window node)	<p>Accesses the <code>get_logic_question()</code> and <code>get_logic_result()</code> in the <code>global.gd</code> script in order to compare the player's input answer with the correct answer to the logic puzzle</p> <p><code>func_on_logic_button_pressed()</code>: connected to the LogicButton node that displays the LogicWindow when pressed.</p> <p><code>func_on_close_requested()</code>: hides the LogicWindow when pressed.</p> <p><code>func_on_line_edit_text_submitted(text: String) -> void:</code> checks to see if the answer entered by the user matches the <code>solve_logic_questions()</code>'s generated answer. Also sets the <code>logic_puzzle_solved</code> flag to true.</p>			Bitwise Operation Puzzle
LogicLabel.gd	LogicLabel (Label node)	<p>Accesses the <code>set_logic_question()</code> function in the <code>global.gd</code> script in order to display the generated bitwise operation question on the LogicLabel. The image to the right is an example of what the LogicLabel looks like in action.</p>			
Dialogue.gd	Dialogue.tscn	<p>Displays a streamline of text when a certain NPC/2D sprite is clicked.</p> <p><code>_on_npc_button_pressed()</code>: The text gives players instructions on how to solve the puzzles around the room/scene. The text remains on screen for 5 seconds before disappearing. The text can be re-displayed at anytime by clicking on the NPC.</p>			Guides
GuideWindowPopUp.gd	BinaryGuideWindow LogicGuideWindow SymbolGuideWindow NoAccessGuideWindow (Window Nodes)	<p>On ready, all guide (hint) windows are hidden.</p> <p><code>func_on_button_pressed()</code>: connected to four different buttons that display their respective guide window pop up when pressed.</p> <p><code>func_on_close_requested()</code>: connected to four different close buttons on the Window nodes that close/hide the guide window pop-ups when pressed.</p>			
NoAccessButton.gd	NoAccessButton (Button node)	<p>On ready, if the <code>logic_puzzle_solved</code> flag is set to false, the NoAccessButton is visible to be clicked to display the NoAccessGuideWindow. Otherwise, the button is hidden and cannot be clicked.</p>	N/A (invisible in scene since modulate alpha is set to 0)		

PopUpHover.gd	DoorPopUpHover BinaryPopUpHover LogicGuideHover LogicPopUpHover NPCPopUpHover (Node2D Node)	Script that controls when certain objects light up when a mouse is hovered over them func _on_area_2d_mouse_entered() : when a mouse hovers over any of the 5 connected Area2Ds, the sprite connected to it lights up to indicate it is clickable. func _on_area_2d_mouse_exited() : returns the highlighted sprites to their original colors when the mouse is no longer hovering.			
Puzzle3_TEST.gd	Puzzle3 (Node2D Node)	On ready, plays the animations of AnimatedSprite2Ds in the scene. The animated sprites that play: - TrashFire - NeonSign - SymbolHint - BinaryHint - SlugNPC			Puzzle3_TEST.tscn