A screenshot of a computer screen

Description automatically generated with medium confidenceA picture containing logo

Description automatically generated

A screenshot of a video game

Description automatically generated

Hello Dan!

For this session we used an analytical challenge to really get into a programming frame of mind!

Part of being a coder is looking at how other games are made and examining their code to learn even more! We began by diving into the script for our project called Colour Change and saw how it was created. It’s all about breaking a concept down- building out a piece of it, testing it, editing, then seeing what needs to be done next!

“+CHILDNAME+” began with a familiar concept, but this time in LUA, something called a table! A table is essentially a list- you give it a name to indicate what it stores, just like an Array in JavaScript! The main difference being In Lua, a table’s index starts with 1, meaning the first item in the list has an index of 1 (ninjas familiar with other coding languages may be more used to tables/lists/arrays starting with an index of 0).

In this game, our objects that we stand on became transparent at different intervals so that we fall through and lose, oh no! “+CHILDNAME+” used the function.GetChildren to get all the child objects of Obstacles and store them in a table.  Then, the code controlling when a tile gets turned off and on was edited with a while loop, so that the obstacles now follow suit!

Finally “+heshe+” got to our ‘hacking’ part! We are going to add a secret item - one that may be a little tricky to reach- but if you do, you are practically invincible!  Touching the secret object will make the tiles always have collision and some transparency. When we touch it, we want all our tiles to always have Collision on so we can't fall, well done to the ‘hacking’ master!

~ Sensei Chris

A screenshot of a computer

Description automatically generated with medium confidenceGraphical user interface

Description automatically generated

Hello Dan!

Today “+CHILDNAME+” got to really delve into 3D scene creation with the Create Studio!

The tool we used this session allowed “+CHILDNAME+” to really stretch “+hisher+” imagination! The Terrain Editor can be used to craft  flat plains, rolling like hills, or mountains, valleys and caves!

Roblox lets us create all these unique terrain features using this tool by morphing the land’s surface - moving it up or down and painting  different materials! We added terrain by sculpting like a paint brush with our mouse onto the scene. Then “+heshe+” manipulated it by clicking, holding, and dragging the mouse. To subtract terrain we used  the same movement with the mouse - click, hold, and dragging to then remove terrain height.  This will remove any terrain that overlaps with our cursor; therefore, lowering mountains or creating valleys!

Finally we investigated how to build new worlds on a smaller scale. So, when building our house, it is important to keep in mind the scale of the player.  You don’t want to get lost in building a house, only to playtest and find the house is tiny compared to the player!  While you can playtest to periodically check the scale, there is a method that lets you compare the scale while building.  You can add a dummy player to your scene!  This adds an avatar, as if it was a part, so it is visible for reference as “+CHILDNAME+” built ! Wooo!

Fantastic work and creativity today superstar!

~ Sensei Chris