A picture containing logo

Description automatically generated

Hello Dan

CHILDNAME+” has once again learned and applied new techniques in Lua to ‘hack’ “+hisher+” own game!

In this session, “+CHILDNAME+” made “+hisher+” previous project much harder by adding in some new ‘random’ code! If a random number between 1 and 0 was lower than 0.2, we disabled the blocks collision! So lets break this down. The chance a random number between 1 and 0 is lower than 0.2, is 20%. This is how we tell the computer to only do something 20% of the time. The next step was to disable the blocks collision, this simply means that we can fall through it! So 20% of our bricks we could fall through in our bridge, very risky!

Once this was finished, we wanted to add a secret way to cross the bridge safely! “+CHILDNAME+” created a new part, and called it ‘button’. We then used a collision function on this object to run a new function. This new function contained a for loop that would generate twenty bricks in the same way as before, except this time there was no funny business! Just plain colourful blocks that we could run across safely.

CHILDNAME+” is doing really well during “+hisher+” sessions and it's great to see how they are evolving their skills alongside learning a new language.

~ Sensei Chris

A red frisbee on a rock in a grassy area

Description automatically generated with low confidenceText

Description automatically generatedA picture containing grass

Description automatically generated

A screenshot of a computer screen

Description automatically generated with medium confidenceA 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