Neural Sprint

User Manual

**Author:** Edward Moriarty

**Date of Document Completion:** 22/05/2017

**Table of Contents:**

* Abstract description of the project - Page 3
* Installation - Page 4
* User Guide - Page 5
* Screenshots - Page 10

Abstract description of project:

The my fourth year project I developed the program titled “Neural Sprint.”

Neural Sprint is a 2D side scrolling platform game built within the Unity 5 game engine. Also contained within the game is a neural network that I developed using the NEAT (Neuroevolution of augmenting topologies) genetic algorithm.

The purpose of this project is to demonstrate the use and application of a Neural Network (specifically NEAT) within one of the most common forms of media: Video Games.

The game will be able to run within two different modes: “Neural Play” and “Free Play”. Neural Play runs the game under the control of the neural network whereas Free Play allows a user to take control of the Player character within the game to free move around the level with no restrictions to explore and attempt to complete.

On startup the game will present the user with a menu with various buttons that they can click on. These buttons will describe the nature of the project, the instructions for both different types of modes and allow the user to decide which mode to run the game under. It will also allow the user to decide to run the game under Neural Mode with a fresh pool or a previously saved one and continue from there.

Upon starting with either of these modes the game will read in values from a config file which will determine which mode to run under and whether or not to load a previously saved pool (these values are altered automatically as the user selects the corresponding menu buttons) as well as whether to run in a debug mode, the dimensions of the level and the contents of the level itself. These values may be freely altered by the user to allow them to create levels for themselves and the neural network to use. The config file is also checked for any incorrect values and formatting as well if one even exists in the correct directory. If such a situation happens the game will not proceed to the level itself and inform the user of the error at hand.

Under Neural Play after the level has been generated the neural network shall begin running through and evaluating the Genomes in it’s Pool and depending on their performance award them a fitness score which will help decide if they continue to the next generation of Genomes and if they create any offspring.

The testing shall continue indefinitely until the user closes the program or returns to the main menu. During a run the user will also be able to scan the pool and replay the current highest fitness Genome or replay the Champion Genome which is a copy of the highest fitness Genome that has been evaluated.

Installation:

The following steps will guide you to acquiring and setting up the game so that it can be run:

**System requirements:**

These are the system requirements that your system must meet in order to run the game:

* Windows XP with Service Pack 2 or newer.
* A graphics card with DirectX 9 (with shader model 3.0) or DirectX 11 (with feature level 9.3) capabilities.  
  (Graphics cards made from 2002 onwards as that is when DirectX 9 was introduced.)
* A CPU with SSE2 instruction set support.(CPUs made from 2001 onwards as that is when it was introduced.)
* A mouse, keyboard and monitor that can display a minimum resolution of 1280x720 will also be required.
* A connection to the internet to acquire the game.

**Acquiring:**

The game be downloaded as a zip file from the following link: https://drive.google.com/file/d/0BxZMhQwJwljMZVU4elhNQS1FNnM/view?usp=sharing

**Unpacking and setup:**

Once you have the zip file you may unpack by right clicking it and selecting “Extract all..” and then pressing the extract button. You may also use external programs such as 7zip and winrar to unpack to archive if you wish.

The game has now been set up and is now ready to be run.

User Guide:

This section will guide you on how to use the game and what does what.

**File contents:**

Within the extracted folder contains all of the files that are required to run the game and are listed as follows:

* **Neural Sprint\_Data** - This is a folder contains all of the files that the game will use to run the game.
* **Acknowledgements.txt** - This is a text file that will contain acknowledgements to the use of other people’s work within the project namely royalty-free art assets, logos made by an online logo generator and a link to the MIT paper that the algorithm that the program uses was proposed.
* **InputValues.cfg** - This a plain text file with the .config extension. This file contains values that the game will use when it is running and may be freely altered by the user in any text editor.
* **Neural Sprint.exe** - This is the executable file of the game itself. Running this file will start the game.
* **ReadMe for InputValues.cfg.txt** - This is a plain text file that acts as a ReadMe file for the InputValues.cfg file. It details the layout of the config file, what values dictate what and how to correctly format the config use so that it can be used.
* **savedPool.pool** - This file will not be present when you unpack the archive however it will be created as the program runs so it will be mentioned here. This is a plain text file with the .pool extension and it is the file that will contain the saved state of the games pool to be read and used by the program.

**Starting the game:**

To start the game the user must run the Neural Sprint.exe file. When they do so they will be prompted by a window to select several setting. Ensure that the resolution is 1280x720, the box for “Windowed” is ticked and the monitor that the user wishes to run the game on is selected from the drop down box. Once this has been confirmed they may then press the “Play!” button to progress.

**Playing the game:**

The first thing the user will see is the main menu that contains the game's title and 5 different buttons which do the following when clicked:

* **Neural Play Instructions** - Will give the user a detailed explanation of what the Neural Play mode entails, looks like and what buttons the user can press to effect the game in this mode.
* **Free Play Instructions** - Will give the user a detailed explanation of what the Free Play mode entails and what buttons the user can press to effect the game in this mode.
* **What is Neural Sprint?** - Will give the user an explanation of what the project and the game is and what it does.
* **Free Play** - Will start the game in Free Play mode.
* **Neural Play** - Will transition to another screen asking the user if they want to start with a fresh pool or load a previously saved one.

When the user attempts to press the Free Play or Neural Play button the program will check for the existence of the InputValues.cfgfile and should the file not exist it will transition to a different menu informing the user as such. The program will also generate a sample/dummy file in its place so that the user may still run the program regardless when they attempt to press the Free Play or Neural Play button again.

If the InputValues.cfg has not been formatted correctly the user will be brought to a menu informing them of this and that the file must be formatted correctly for the game to accept it telling them to refer back to the “ReadMe for InputValues.cfg.txt” file on how to format it correctly.

Should the file exist and formatted correctly the game will launch into the main part of the game in either Free Play or Neural Play Mode.

**Free Play:**

In this mode the user will have full control of the Player character with no inputs or restrictions from the neural network. The user may explore and attempt to complete the level as much as they want and when they are satisfied they may close the game or return to the main menu by pressing the Escape key.

**Player controls:**

* **W** - If this button is pressed together it allows the player to jump higher than normal, this increase in height stacks with the bonus from J.
* **A** - If this button is pressed it will move the player left.
* **S** - If this button is pressed it will make the player duck and prevent further horizontal movement.
* **D** - If this button is pressed it will move the player right.
* **J** - If this button is pressed it increases the player's movement speed and jump height, this increase in height stacks with the bonus from W.
* **K** - If this button is pressed the player will jump up.
* \* - If both A and D are pressed at the same time the they cancel each other out.

**In-Game Objects:**

* **The Player** - This is the Player character that the user/neural network controls with the goal of guiding them to the end of the level at the rightmost part of the level.
* **Tiles:** There are different Tiles in the game that have different functions which are as follows:
  + **Background Tile** - A Tile that has a sprite and has no active colliders, used for background decoration.
  + **Ground/Platform Tile** - A Tile that has a sprite and an active collider. The player may stand and touch this tile with no consequence.
  + **Lava Tile** - A Tile that has a sprite and an active collider. Should the player come into contact with this Tile the player is killed.
  + **Empty Tile** - A Tile that has on sprite and no collider. Used internally to pad out the surrounding area of the level. Can be made visible in debug mode (activated in the config file).
  + **Border Tile** - A Tile that has a sprite and an active collider. Used internally to make a border around the level to ensure that the Player can never leave the confines of said level.
* **The Enemy(s)** - This is the object that represents an enemy character in the game. They will move in a direction until they encounter another enemy, a wall or are about to walk off a platform at which point they will turn around and move in the opposite direction. If the Player comes into contact with an enemy it will kill the Player.

Should the Player reach the end of the level or are killed the level is restarted. The Level may also be manually restarted by pressing the R key.

**Neural Play:**

Neural Play Mode allows the neural network to take over control of the player.

The neural network will generate Genomes (Specimens that are doing the thinking and controlling the player) divided into Species and placed into a Pool.

Once every Genome (300 in each Generation) has had a chance to complete the level the neural network will remove the weak and stale (don't improve over a period of time) Genomes and Species and will make a new Generation of Genomes from the survivors.

This will continue indefinitely, even if one reaches the end of the level it can still always try and do it faster and get a better score.

The controls are as follows in addition to the controls available in Free Play:

* **T** - Scan the Pool and replay the highest scoring Genome that still exists.
* **Y** - Replay the highest scoring Genome that has ever performed a.k.a. the Champion Genome. This is performed at the start of every new Generation to ensure consistency.
* **U** - Manually save the state of the Pool to a file. This is done

automatically after each new generation.

**UI Overview:**

* **Gen:** Displays the current generation or the generation the Champion Genome is from if it is being run.
* **Species:** Displays the current Species the running Genome is from.
* **Genome:** Displays the current genome being run.
* **Top Pool Fitness:** Shows the fitness of the highest scoring Genome that has been tested currently in the pool.
* **Run Fitness:** Shows the current fitness of the Genome being run at that point in time.
* **Staleness:** Shows the current Staleness value of the Species being run. [A score of 15 indicates a stale Species.]
* **Timeout:** Shows the current time left in a Genome's run. Resets when progress is made.
* **Bonus Timeout:** A Bonus is added to Timeout for how long the run has lasted thus far [Automatically added to Timeout].
* **End of Level Bonus:** A Fitness bonus received if the player reaches the end of the Level. Reduces over time.
* **Champion Genome Fitness:** Displays the fitness of the Champion Genome.  
  

The above picture shows the graphical representation of the Genome currently being run.

On the left hand side we have the **Input Nodes** which display the inputs of the Neural Network and what is around the player at any given time.

In the middle we have the **Hidden layer** which may contain more Nodes that exist between the Input and Output Nodes.

On the right we have the **Output Nodes** which determine which button to press depending on the value that they end up with up.

Connecting all of the Nodes together are Genes which take a value from a Node, alter it and then pass it along to the adjoining Node at the end of it.

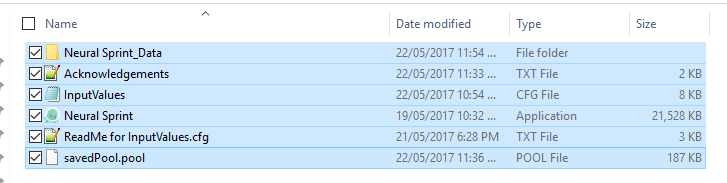
**What the Colors Signify:**

* **White/Transparent:** A Node or Gene of this color has a value/weight between -1 and 0.
* **Red:** A Node or Gene of this color has a value/weight of -1 or less.
* **Green:** A Node or Gene of this color has a value/weight of greater than 0.
* **Blue:** This node represents where the player is a has a value of 0.
* Input Nodes always have a value of -1, 0 or 1.
* Output Nodes will trigger a button press if it is Green.
* **Bias Node:** The Bias Node is the last Input Node and is shown next to the Input Nodes. It always has a value of 1.

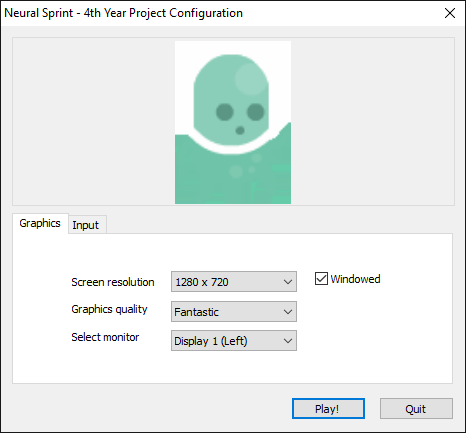
Screenshots:

This section will display screenshots that act as complementary information for the User Guide listed above.

T**he contents of the game’s directory:**



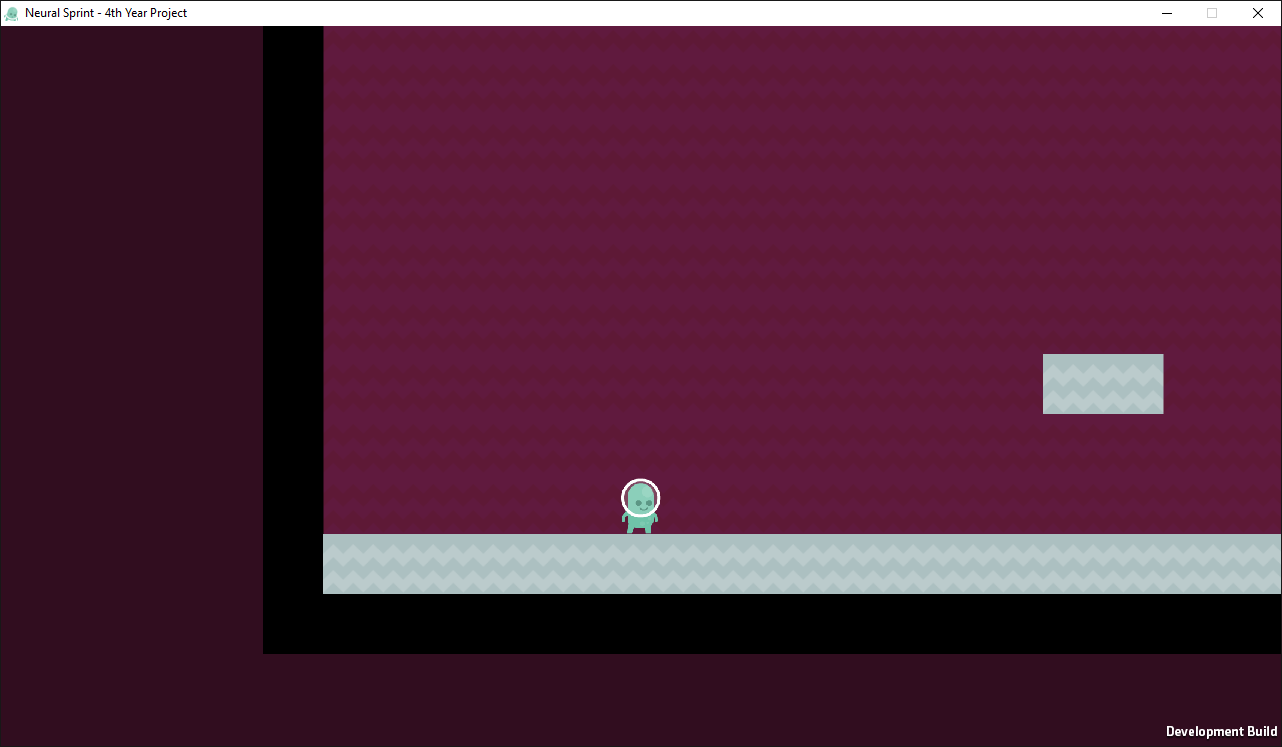
**The correct setting to be used when launching the game:**



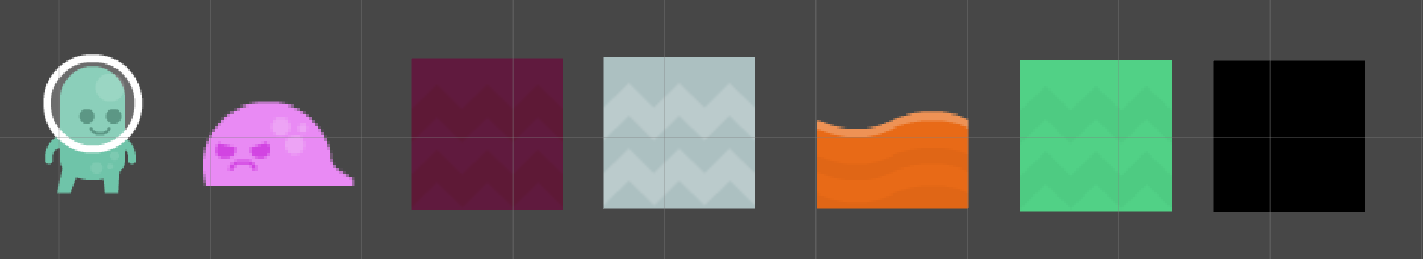
**The main menu presented to the user:**



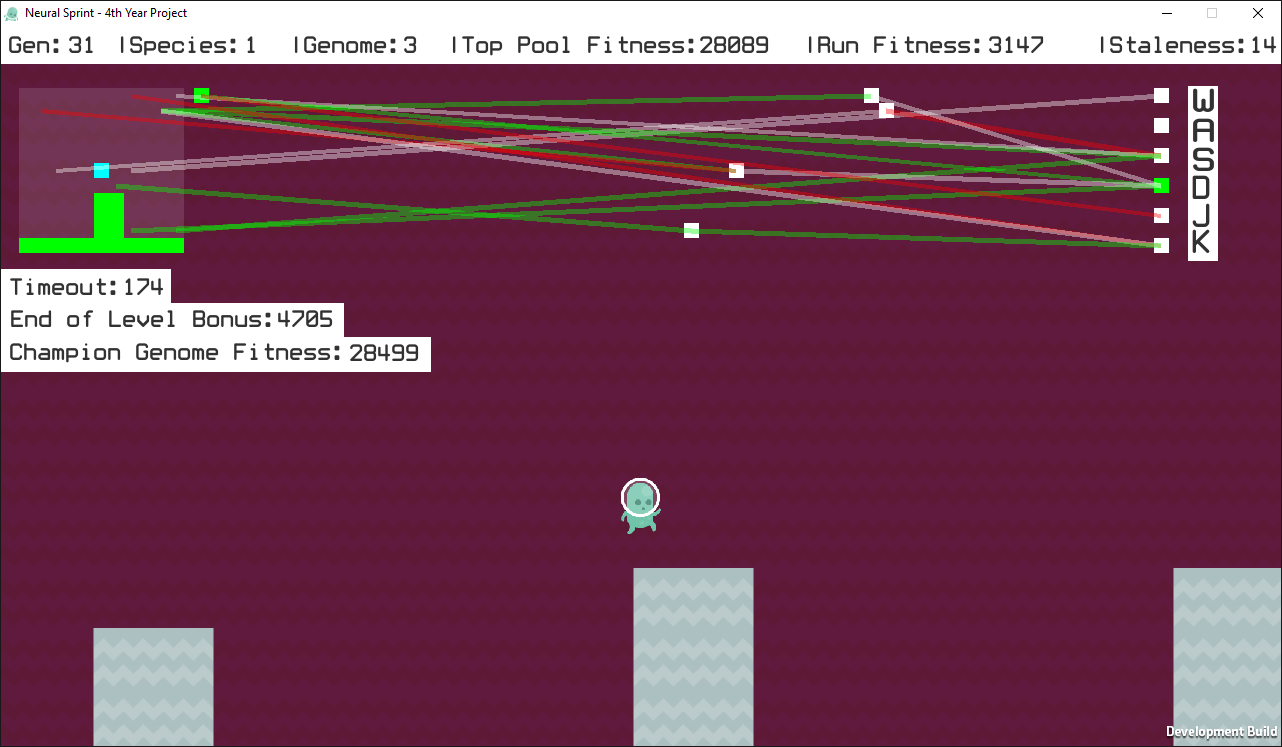
**A screenshot of the game in Free Play mode:**



**The arrangement of Objects in the Game, from left to right:** [Player, Enemy, Background Tile, Ground/Platform Tile, Lava Tile, Empty Tile(rendered) and Border Tile]



**A screenshot of the game in Neural Play mode:**

****