This document is designed to be an aide to teachers. It will describe the basic concepts each minigame is designed to teach. There are also a few hardware facts scattered throughout the game that players can click on to learn about, but this is an optional action.

1 Logic Minigame

The goal of this game is to teach children the basics of binary logic. The tubes and machines the player constructs exactly match a digital logic circuit. The reverser, green smoosher, and blue smoosher, correspond to a not gate, and gate, and or gate respectively. The early levels explain what each gate does, while later levels require to the player to understand how to combine them to create somewhat arbitrary truth tables.

2 Robot Maze Minigame

The Robot Maze Minigame introduce kids to basic programming through the use of a list of simple commands and loops. This game starts out in a programming phase, where the player can see the maze that the robot has to complete. The player programs the robot by dragging these buttons, which represent the various commands, into the spots in the instructions area. The game also has a simple loop structure built in, so the player can put commands in this subroutine area and then insert those commands into the program using the subroutine icon. The player has a limited number of spaces available, so they have to figure out how to use this loop structure to make the code more efficient and not run over their limit. Once the player is done writing the program, they hit go, and then they get to watch the robot run through the maze, executing their commands.

The game is supposed to be a fun and visual introduction to programming. It uses prewritten commands for the user, to make it easier to learn, and it uses buttons with images instead of words to make it more appealing to kids. This game gets the player familiar with a very simple programming language that has a short list of commands and a very simple looping structure. They must mentally keep track of the position and direction of the robot, the variables that the program is updating, and perform steps that work in that context. If they run into a wall or obstacle, they blow up, but they get to try again.

3 Memory Matching Minigame

The primary educational goal of the memory matching game is to teach some of the basic concepts behind memory storage systems, in particular the various levels in which data is stored and how that data is accessed.

Computers use various levels of memory to store information. Some levels are larger and able to hold a great amount of information, however it takes a long time to access that information. Then there are other levels that are much smaller, allowing for much quicker

acquisition of data. To Illustrate this point the memory matching game is broken up into three stages, the Hard Disk, Cache and CPU Register, each of which represent a stylized level of memory: The Hard disk is the largest, contains the most data slots, and will thus take the longest to complete; The CPU Register is very small, contains only a few data slots, and can be completed very quickly; and the Cache is somewhere in the middle.

To access data stored in memory, computers use binary addresses to keep track of where a piece of data is stored. In the memory matching game, each row and column of data slots is labelled with a binary number. In addition to reinforcing the concepts taught by the Binary minigame, the combination of row and column number serves as the binary address for each data slot. As the player goes through the level and reveals the data in each slot, Gladys keeps track of the addresses of each data type, and will give the player hints about the location of the matching piece of that data type. Using these hints to locate pieces of blueprint data will reinforce through repetition the idea of memory addressing.

4 Binary Minigame

This games purpose is to give kids familiarity with binary numbers. Each barrel represents a binary digit: 0 if the barrel is empty; 1 if the barrel is full. For the first few levels, the goal is given in decimal, so the player gains experience in translating from decimal to binary. On the higher difficulties, the goal is given in binary, so the player also learns how to read binary.

Ideally, most of this is done under the hood, so to speak. The player is not being told that he or she is working in a different base system for counting. Rather, the concept is introduced in an easy to understand way.

5 Robot Building Minigame

Like the Robot Maze Minigame, this game is designed to teach about program flow. This is the language computers understand and how you tell computers to do something. The basic idea is that the player must give the computer a recipe, or program, but unlike humans computers can only read instructions that are put in certain ways.

This game is designed to give a basic intuition into the proscriptive nature of programming. This is the idea that there are rules on how a program can be written. This is done by using a simple picture based language that the player uses to build robots. The rules of the language is that only certain commands can go in certain places, a part of the robot must have its shape decided before it can be painted, etc. If the player tries to build the robot and there are errors, the compiler, in this case in the form of Gladys, will tell the player what they have done wrong. Since the language is so simple, the error messages can be exact in the way a compiler cannot.

At later levels, a simple loop command is available. This loop is not like most modern languages in that it also requires another command to finish it off. This is because of the lack of indentation of brackets that are used by Java, C, and python languages to name a few. However, there are languages that have this loop structure that are being used in the world today.