Event Driven Programming

Data in event-driven programming, the flow of the program is controlled by events.
Events can be user actions such as:

- Mouse clicks (or the touchscreen equivalent)
- Key presses
- Hovering over a picture
- Voice input ("OK Google...")

Events can also be also be triggered by:

- Sensors (e.g. if movement is sensed, turn the light on)

```
onEvent("startbutton", "click", function() {
  setScreen("Game");
  setTimeout( function() {
    setScreen("Score");
  }, 5000);
});
```

Source: code.org

Graphical User Interface

The GUI (Graphical User Interface) is the screen that the user interacts with. It provides the interface between the user and the computer or program.

The Tappy Tap App will need three screens – Home screen, Game screen and Score screen.

Key Vocabulary

Key Word	Definition
Analysis	A through study doing a careful analysis of a problem.
Debug	The process of identifying and fixing errors in code.
Event handler	Used to determine when to collect data and what to do with it once it has been collected.
GUI	Graphical User Interface
Logic error	When the computer is able to run the program, but it does not work properly because the logic of the code is incorrect
Selection Statement	Used when there is more than one option for the user to select from. IF, ELSE, ELIF
Sub Routine	Small blocks of code in a modular program designed to perform a particular task.
Syntax error	When the computer does not understand the code because it has been typed incorrectly and doesn't follow the rules or grammar of the programming language.

Subroutines

Subroutines are **small blocks of code in a modular program designed to perform a particular task.**

getText("id") is a built-in subroutine that collects the text entered into a textbox; "id" is to be replaced with the name given to the text box.

```
var x = getText("id");
```

Event handler

You can use an **event handler** to determine when to collect the data and what to do with it once it has been collected and linked with a variable.

```
onEvent("login", "click", function() {
  var username = getText("username");
});
```

Debugging is the process of identifying and fixing errors in code. This is usually done as the code is being built, each new section of code will be tested is the program develops.

Syntax errors – When the computer doesn't understand the code because it has been typed incorrectly and doesn't follow the rules or grammar of the programming language.

Logic errors – When the computer is able to run the program, but it does work properly because the logic of the code is incorrect. These sorts of errors are harder to spot and more difficult to debug and fix.

Selection

A selection statement is used when there are more than one option for a user to select from. It causes the program to make a choice and flow in a given direction. In these examples they used the statements if, else, else if.

Example

These two blocks of code have the same purpose, which is to provide feedback to the user at the end of a game.

Code block 1

```
if (score > 10) {
  setText("feedback_label", "Great Work");
} else {
  setText("feedback_label", "Hard Luck");
}
```

Code block 2

```
if (score > 10) {
  setText("feedback_label", "Great Work");
} else if ((score > 6)) {
  setText("feedback_label", "Not Bad");
} else {
  setText("feedback_label", "Hard Luck");
}
```