BRAIN HACKERS APP DEVELOPMENT CURRICULUM

Lesson 6
Balloon Pop App – Variables and Random Numbers

SUMMARY

In this lesson, an app will be created that uses variables whose values change as a user plays the game. This section also introduces the random number generator and the use of random numbers. A simple game will be created in which a balloon is inflated, but every time air is pumped into the balloon, there is a risk of popping the balloon and losing the game.

DESIGNER WORKSPACE

- For Screen1, set the BackgroundColor to Black – NOTE: this app will use white text against a black background, which can be thought of as "Reverse Contrast"
    - o Set the ScreenOrientation to "Portrait"

Key Concepts
Reverse Contrast –contrast between features of a design is achieved through the opposite of what would usually be expected such as when white letters are used against a black background, instead of the typical black letters against a white background

- Place a Label to serve as a title at the top of the screen
    - o Set the TextColor to White
    - o Enter the text, "Balloon Pop"
- Place a HorizontalArrangement below the title
    - o Insert a Label
        - ▪ Change the Text to, "Score"
        - ▪ Change the TextColor to white
    - o Insert a second Label – NOTE: this Label will serve as a spacer
        - ▪ Delete the text

- o Insert a third Label - NOTE: this Label will display the score
  - Change the text to, "0"
  - Change the TextColor to white
- Insert a Canvass
  - o Set the Height to 300 pixels and the Width to Fill Parent – NOTE: you may need to adjust the Height of the Canvass depending on the size of your screen
  - o Set the BackgroundColor to White
- From the Drawing and Animation components, drag a Ball onto the Canvass and place the Ball near the center
  - o Change the PaintColor for the Ball to Magenta
- Place a HorizontalArrangement below the Canvass
  - o Set the Width to Fill Parent
  - o Insert a Button – NOTE: a user will tap this Button to inflate the balloon
    - Change the Text on the Button to, "Pump"
  - o Insert a Label to serve as a spacer,
  - o Insert a Button with the text, "Reset" – NOTE: this Button will return the balloon to the size it started
- Adjust the Width of the spacer in the HorizontalArrangement so that the buttons are on the far right and left of the screen – NOTE: the spacer serves to provide separation, which means placing space between two components that you do not want a user to accidentally confuse

Key Concepts

Separation – providing space between components of a visual display, which can minimize the likelihood of the components being confused, such as accidentally pressing the wrong button

- Add a Button
  - o Change the text to, "Restart Game" – NOTE: this Button will allow a user to take the score back to "0"
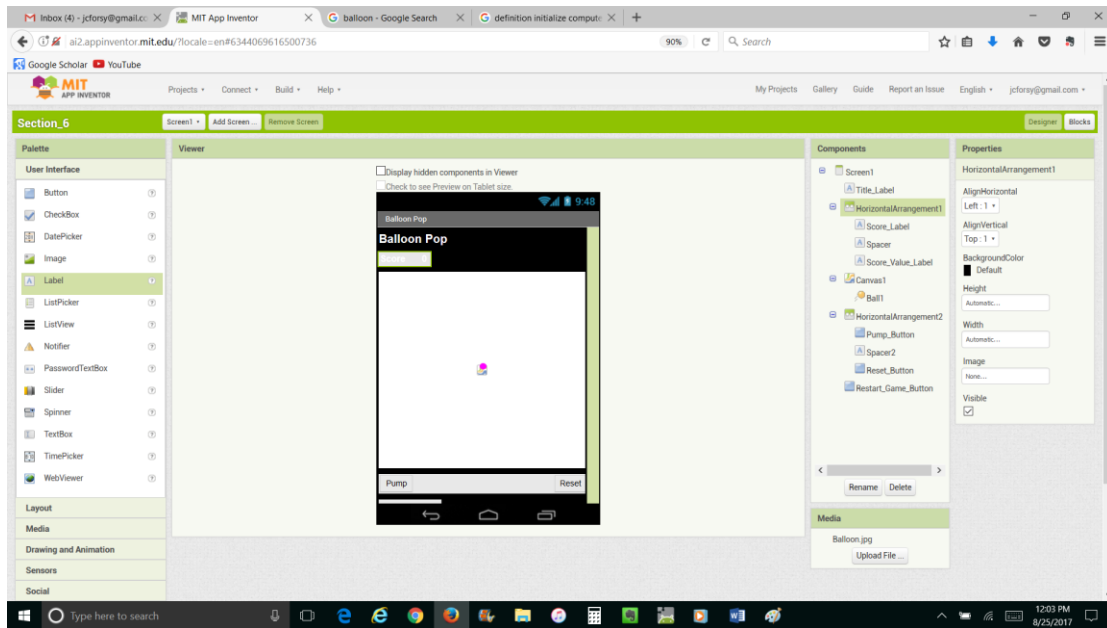
FIGURE 1. This image shows the general layout of the screen, although the Viewer panel within MIT App Inventor does not correctly display the background colors

BLOCKS WORKSPACE

- From the Variables blocks, add an ***initialize global ___ to*** block – NOTE: this block creates a variable that will be the limit at which the balloon will pop
    - o Assign the variable the name, "Limit"

Key Concepts
Variable – a symbol that is used to represent a value, with the potential for the value to change.

- o From the Math blocks, attach a ***random integer from ___ to ___*** block to the slot of the block initializing the Limit variable – NOTE: this block selects a random number from the given range and assigns that number to the Limit variable

Key Concepts
Random Number – a number for which it is impossible to predict its value based on the past or the present.

- ▪ Set the range for the random number so it is 20 to 100 – NOTE: a number will be selected that is between 20 and 100

- Add a second *initialize global ___ to* bock – NOTE: this variable will be the player's ongoing score
    - o Name the variable "Score"
    - o Assign the variable a value of "0"
- Add a third *initialize global ___ to* block – NOTE: this variable will be the number of points gained in a round and will be subtracted from the player's score if the balloon pops
    - o Name the variable, "Gain"
    - o Assign the variable a value of "0"

FIGURE 2. The blocks initializing the three variables should look like this.

- Add a *when ___ .Click do* block for the Pump Button
    - o From the blocks for the Ball, add a *set ___ .Radius to* block – NOTE: this block increases the size of the ball each time the Pump Button is tapped
        - ▪ Attach an addition Math block that increases the current radius by 5
    - o From the Variables blocks, add a *set ___ to* block – NOTE: this expression increases the points awarded as the player pumps the balloon bigger and bigger
        - ▪ Change the *set ___ to* block to the Score variable
            - • Attach an addition Math block that increases the current score by the current radius of the ball
    - o Add a *set ___ to* block – NOTE: this block accumulates how many points have been awarded for the current round so these points can be subtracted from the score if the balloon pops
        - ▪ Change the *set ___ to* block to the Gain variable
            - • Attach an addition Math block that increases the current gain by the current radius of the ball
    - o From the blocks for the score value Label, add a *set ___ .Text to* block – NOTE: this block updates the score that is displayed each time the Pump Button is tapped
        - ▪ Attach a *get ___* block for the current score

- From the Control blocks, add an ***if then*** block – NOTE: this block provides instructions for the condition where the balloon exceeds the limit and pops

Key Concepts
If then – an expression that checks to see if some condition exists and instructs the program what to do if the condition exists

- For the **if** slot, attach a comparison Math block – NOTE: this block asks if the balloon has exceeded the limit and popped
  - Set the comparison so it asks if the ball radius is greater than the Limit variable
- For the ***then*** slot, attach a ***set ___ .Radius to*** block returning the radius of the ball to "5" – NOTE: this block returns the ball to its starting size
- For the ***then*** slot, attach a ***set ___ to*** block that subtracts the Score by the Gain – NOTE: this block reduces the score by the number of points earned prior to popping the balloon
- For the ***then*** slot, attach a ***set ___ to*** block that returns the Gain to "0" – NOTE: this block returns the Gain variable to zero for the next round
- For the ***then*** slot, attach a ***set ___ .Text to*** block for the score value Label and attach a ***get ___*** block for the current score – NOTE: this block updates the displayed score after the balloon pops
- For the ***then*** slot, attach a ***set ___ to*** block for the Limit and attach a ***random integer from ___ to ___*** block with a range of 20 to 100 – NOTE: this block sets a new value of the Limit variable for the next round
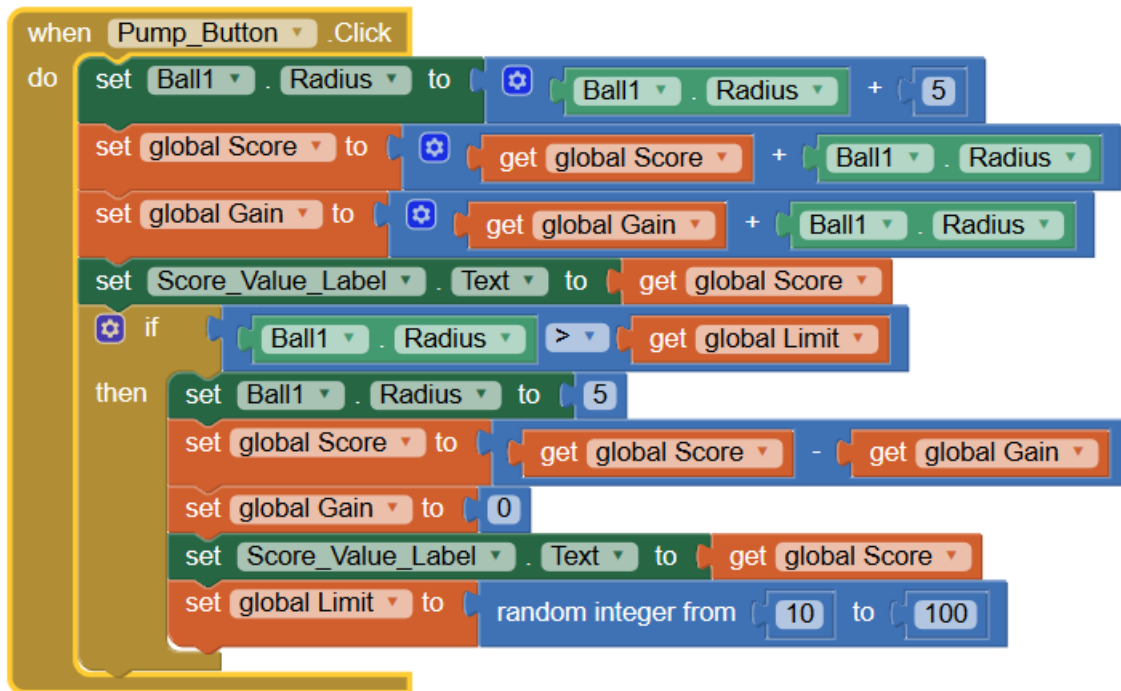
FIGURE 3. This image shows the blocks inserted into the **when ___ .Click do** block for the Pump Button.

- Add a **when ___ .Click do** block for the Reset Button – NOTE: this block allows the user to return to the beginning without losing any points
  - ○ Insert a **set ___ .Radius to** block returning the radius of the ball to "5"
  - ○ Insert a **set ___ to** block that returns the Gain to "0"
  - ○ Insert a **set ___ to** block for the Limit and attach a **random integer from ___ to ___** block with a range of 20 to 100
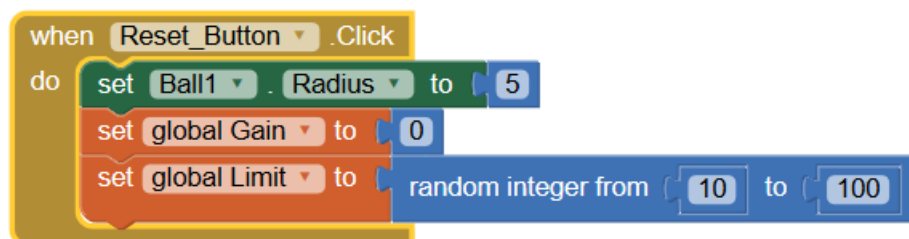

FIGURE 4. This image shows the blocks inserted into the **when ___ .Click do** block for the Reset Button

- Add a *when ___ .Click do* block for the Restart Game Button – NOTE: this button allows a player to start a new game
  - o Insert a *set ___ to* block that returns the Score to "0"
  - o Insert a *set ___ .Text to* block for the Score Value Label and attach a *get ___* block for the current score
  - o Insert a *set ___ .Radius to* block returning the radius of the ball to "5"
  - o Insert a *set ___ to* block that returns to Gain to "0"
  - o Insert a *set ___ to* block for the Limit and attach a *random integer from ___ to ___* block with a range of 20 to 100
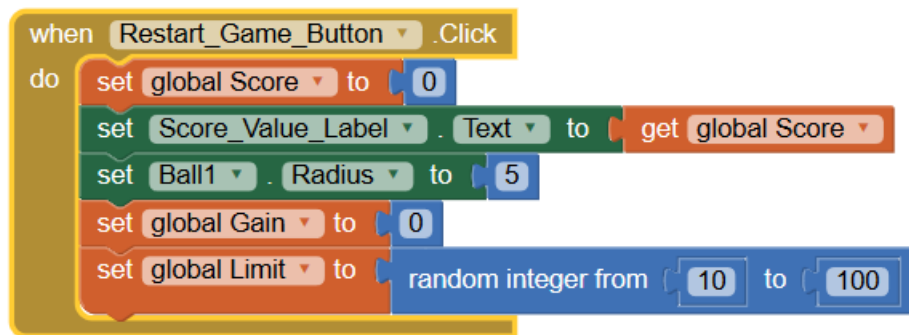

FIGURE 5. This image shows the blocks inserted into the *when ___ .Click do* block for the Restart Game Button
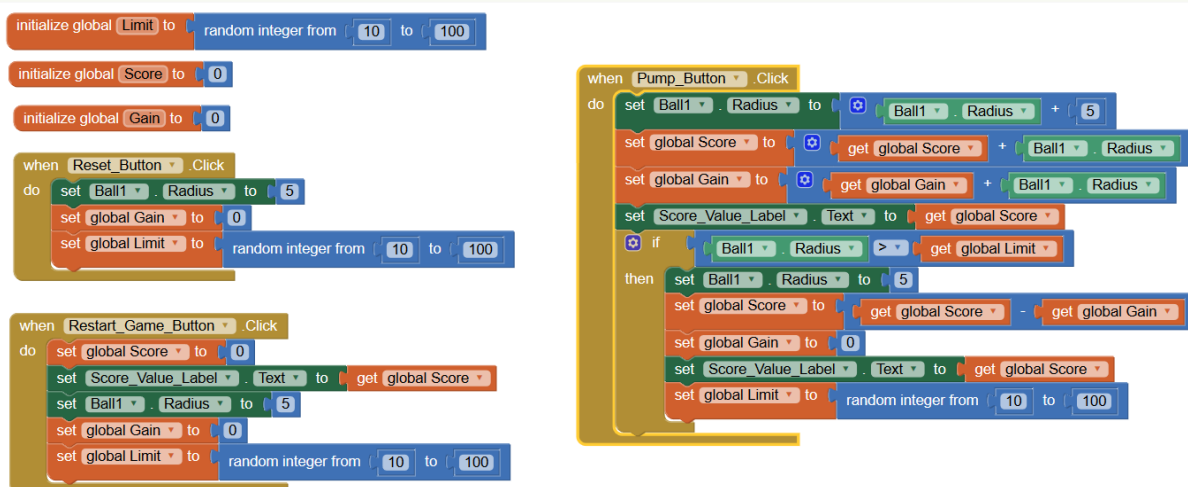

FIGURE 6. This image shows how the Blocks workspace should appear

BALLOON POP APP EXERCISE

When the balloon pops, it is equivalent to a system crashing. It is comparable to when a bridge collapses due to excessive weight or a population of animals crashes due to the introduction of a new predator. However, in both natural and man-made systems, there are mechanisms in place that lessen the likelihood the system will crash. For example, people might notice cracks in the bridge long before it collapses or animals might learn to avoid predators. Based on the topic assigned, identify a system that could crash if placed under too much pressure and create a game like the Balloon Pop game that simulates that system. Through this activity, you are creating a model that reflects beliefs about how the system operates, with the game being a simulation of the system.