BRAIN HACKERS APP DEVELOPMENT CURRICULUM

Lesson 7

Millisecond App – Clock, States and Sliders

SUMMARY

The time scales at which events occur vary with natural systems. People tend to think in seconds, minutes and hours, however a lot can happen at the millisecond scale. This lesson asks students to think at the millisecond scale. This is the scale at which events happen in popular games and other software-based and communications systems. An app will be created in which students can adjust the response time on a millisecond scale and test their reaction time. This section introduces the use of a Clock as a timer, as well as States, and the use of Sliders for user input.

ASSUMPTIONS

For this exercise, students will need access to the Internet to download an image of a target or bullseye.

Key Concepts

Millisecond – one/thousandth of a second, there are 1,000 milliseconds in a second

DESIGNER WORKSPACE

- On the Properties for Screen 1, set the ScreenOrientation to Portrait
- Place a Label at the top of the screen with the text, "Tap When You See the Target" – NOTE: in this case, the Label serves to provide instructions to a user

Helpful Tips

Assume a user will not know how to use your app, misunderstanding what may seem obvious. However, for regular users, instructions take up screen space and can be annoying. Consequently, instructions should be simple and direct.

- Insert a HorizontalArrangement
    - Set the AlignHorizontal to “Center”
    - Set the Height to “50” pixels and the Width to “Fill Parent” – NOTE: the Horizontal Arrangement will be used to center the target image on the screen

Helpful Tips

HorizontalArrangements offer an effective way to center an object on the screen

- Place a Button in the HorizontalArrangement
    - Set the Image for the Button to be the image of a target
    - Set the Height to “50” pixels and the Width to “50” pixels
    - Delete the Text
    - Remove the check in the Visible checkbox – NOTE: for this app, the target will only appear for brief periods of time and will usually be non-visible

Key Concepts

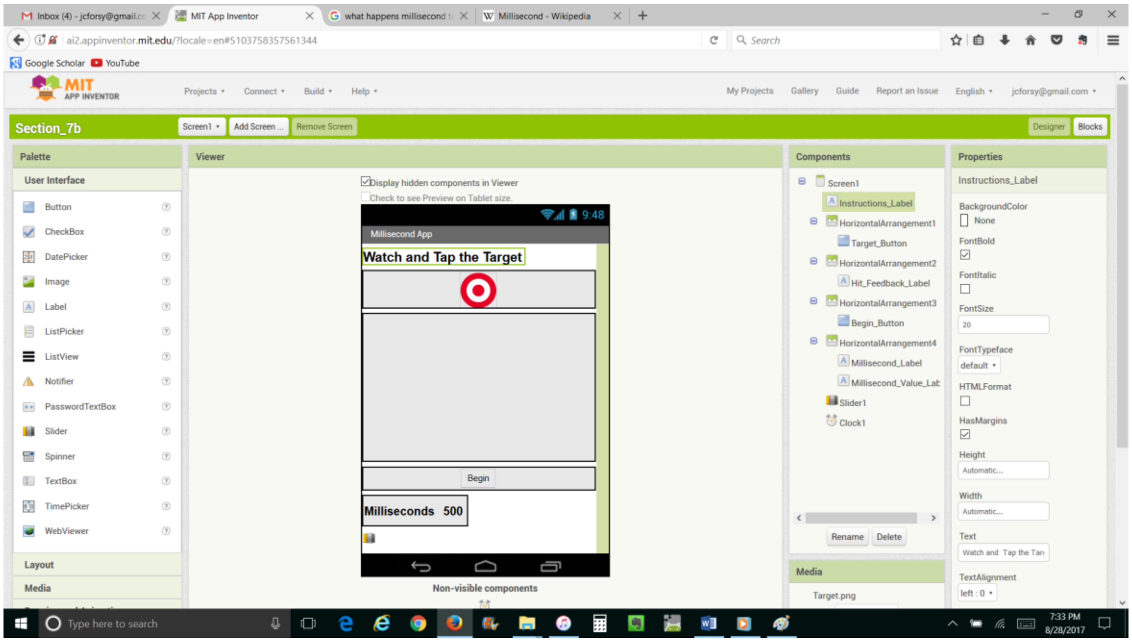
Non-visible component – visibility is a property of a component, and with many apps, while the component is always present, it may shift back and forth from visible to non-visible

- Insert a HorizontalArrangement
    - Set the AlignHorizontal and AlignVertical to “Center”
    - Set the Height to “Fill Parent” and the Width to “200” pixels
    - Place a Label in the HorizontalArrangement – NOTE: this Label will serve as feedback to the player when they hit the target in the allotted time period
        - Place a check in the FontBold checkbox
        - Set the FontSize to “35”
        - Delete the Text

Key Concepts

Feedback – informs a user of the effectiveness of their actions, effective and timely feedback can be critical to designing an app that produces an enjoyable user/player experience

- Insert a HorizontalArrangement
    - Set the AlignHorizontal to “Center”
    - Set the Width to “Fill Parent”
    - Insert a Button
        - Change the text to, “Begin” – NOTE: the user will tap this Button to begin each trial
- Insert a HorizontalArrangement
    - Set the AlignVertical to “Center”
    - Set the Height to “40” pixels
    - Insert a Label
        - Change the text to “Milliseconds”
    - Insert a second Label
        - Change the text to “500” – NOTE: this Label will be used to show the current setting for the number of milliseconds
- From the User Interface components, insert a Slider
    - Set the Width to “Fill Parent”
    - Set the MaxValue to “1000”
    - Set the ThumbPosition to “500” – NOTE: the Slider will be used to adjust the number of milliseconds that the target is visible
- From the Sensors components, add a Clock – NOTE: the Clock will be a non-visible component



BackgroundColor: None
FontBold: ☑
FontItalic: □
FontSize: 20
FontTypeface: default
HTMLFormat: □
HasMargins: ☑
Height: Automatic...
Width: Automatic...
Text: Watch and Tap the Tar
TextAlignment: left : 0

Figure 1. This image shows the general layout of the screen, including the non-visible components of the user interface

BLOCKS WORKSPACE

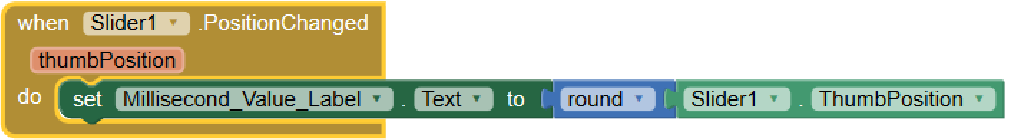
- Add an ***Initialize global ___ to*** block
  - Name the Variable "State"
  - Assign a value of "1" – NOTE: for this app, when the State is set to "1" the game will be in a condition where the target is not visible, with the target visible when the value of the State variable is "0"

Key Concepts

State – refers to the current condition of a program which may be described by the values of its variables and settings

- From the Slider blocks, add a ***when ___ .PositionChnaged*** block
  - Insert a ***set ___ .Text to*** block for the Millisecond Value Label
  - From the Math blocks, insert a Round block into the slot – NOTE: this block will round the number from the Slider to display a whole number

    - From the Slider blocks, insert into the slot a ___ ***.ThumbPosition*** block – NOTE: this will change the value of milliseconds based on the position of the Slider on a scale from 10 to 1000



```
when Slider1 .PositionChanged
  thumbPosition
do set Millisecond_Value_Label . Text to round Slider1 . ThumbPosition
```

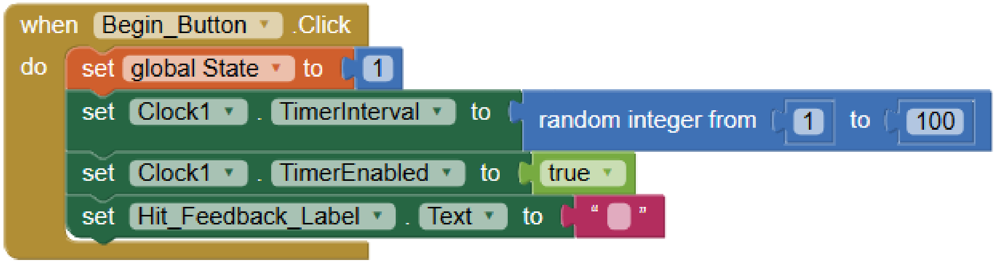
Figure 2. The ***when*** ___ ***.PositionChnaged*** block for the Slider should appear as shown

- Add a ***when*** ___ ***.Click do*** block for the Begin Button
    - Insert a ***set*** ___ ***to*** block for the State variable
        - Attach a number block setting the value to “1”
    - From the Clock blocks, insert a ***set*** ___ ***.TimeInterval to*** block
        - Attach a ***random integer from*** ___ ***to*** ___ block
            - Set the range of “1000” to “3000” – NOTE: this blocks creates a “Random Onset,” preventing a player from anticipating when the target will appear

Key Concepts

Random Onset – by making the time at which events occur random, players will be unable to anticipate events

        - From the Clock blocks, insert a ***set*** ___ ***.TimeEnable to*** block
            - From the Logic blocks, attach a True block – NOTE: by setting TimeEnabled to “True,” the clock will start and run for the duration set in the previous block
    - Insert a ***set*** ___ ***.Text to*** block for the Hit Feedback Label
        - Attach an empty text string – NOTE: with this block, nothing appears for the Hit Feedback Label during the period after a trial has begun and prior to the player hitting the target



```
when Begin_Button .Click
do  set global State to 1
    set Clock1 . TimerInterval to random integer from 1 to 100
    set Clock1 . TimerEnabled to true
    set Hit_Feedback_Label . Text to " "
```

Figure 3. This figure shows how the ***when*** **___** ***.Click do*** block should appear for the Begin Button

- Add ***when*** **___** ***.Click do*** block for the Target Button
  - Insert ***set*** **___** ***.Text*** block for the Hit Feedback Label
  - Attach a text string with the text, "HIT" – NOTE: when the players hits the target within the allotted time period, this block changes the feedback Label to show "HIT"
- From the Clock blocks add a ***when*** **___** ***.Timer do*** block – NOTE: this block instructs the program what to do when the Clock has timed out

Key Concepts

Clock – when triggered, the Clock counts down for the time specified and when done, the instructions within the Clock block are executed

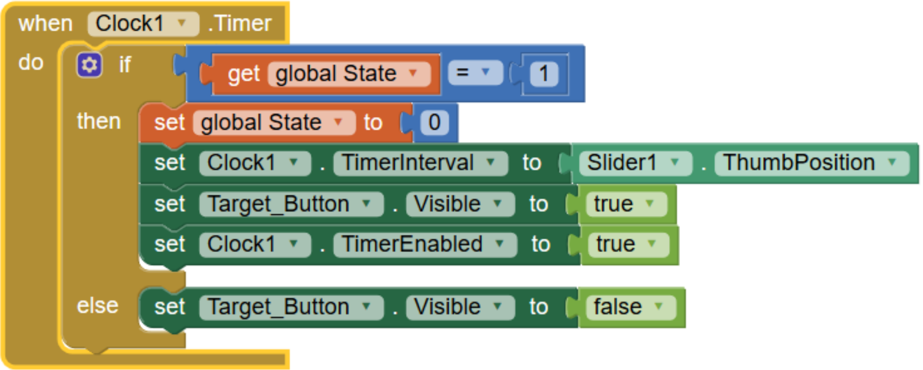
- From the Control blocks, insert an ***if then*** block and add an ***else*** slot – NOTE: this block provides separate instructions for the two conditions of the "State" variable

Key Concepts

If then else – this statement expresses what instructions should be performed for either of two conditions

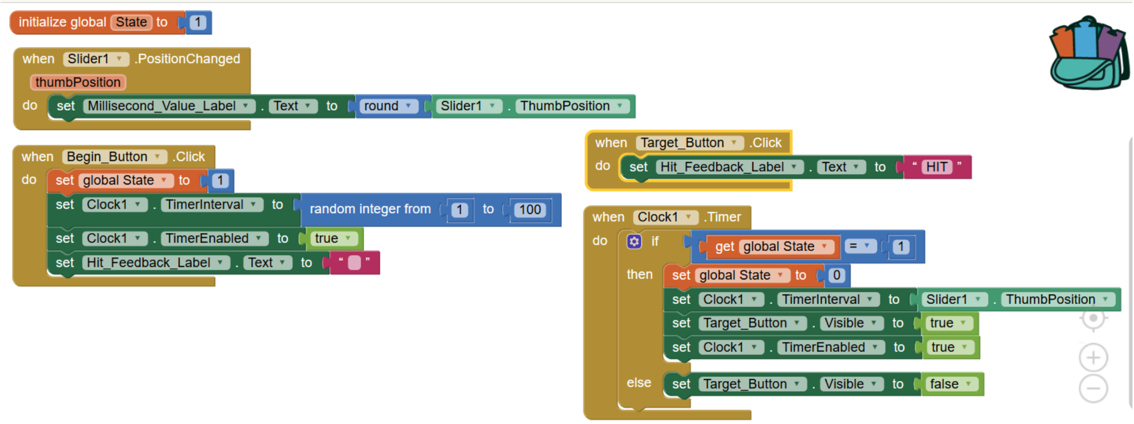
- For the ***if*** slot, insert a Math equivalence block
  - Insert a ***get*** **___** block as a pointer to the variable "State"
  - Insert a number block with the number, "1" – NOTE: this block is for the condition of the State variable in which the player is either between trials or has started a trial and the target has not been presented

- For the ***then*** slot, insert a ***set ___ to*** block for the variable “State”
    - Insert a number block with the value “0” – NOTE: when the timer has timed out for the delay period, the value of the State variable is changed and the target is made visible
    - Insert a ***set ___ .TimeInterval to*** block
        - Attach a ***___ .ThumbPosition*** block – NOTE: this block sets the time interval for the target presentation to the value of the Slider
    - Insert a ***set ___ .Visible to*** for the Target Button
        - Attach a True block – NOTE: this block makes the target Button visible
    - Insert a ***set ___ .TimeEnabled to*** block
        - Attach a True block – NOTE: this block starts the timer to count down the duration the target is visible
- For the ***else*** slot, insert a ***set ___ .Visible to*** block for the Target Button
    - Attach a False block – NOTE: when the duration for the target presentation is complete, the target Button is switched to non-visible



```
when Clock1 .Timer
do  if    get global State = 1
    then  set global State to 0
          set Clock1 . TimerInterval to Slider1 . ThumbPosition
          set Target_Button . Visible to true
          set Clock1 . TimerEnabled to true
    else  set Target_Button . Visible to false
```

Figure 4. This figure shows how the ***when ___ .Timer*** block should appear



```
initialize global State to 1

when Slider1 .PositionChanged
  thumbPosition
do  set Millisecond_Value_Label . Text to round Slider1 . ThumbPosition

when Begin_Button .Click
do  set global State to 1
    set Clock1 . TimerInterval to random integer from 1 to 100
    set Clock1 . TimerEnabled to true
    set Hit_Feedback_Label . Text to " "

when Target_Button .Click
do  set Hit_Feedback_Label . Text to " HIT "

when Clock1 .Timer
do  if get global State = 1
    then set global State to 0
         set Clock1 . TimerInterval to Slider1 . ThumbPosition
         set Target_Button . Visible to true
         set Clock1 . TimerEnabled to true
    else set Target_Button . Visible to false
```

Figure 5. This image shows how the Blocks workspace should appear.

MILLISECOND APP EXERCISE

For the assigned topic, identify events that occur on a millisecond time scale. Examples of events might include: duration of a single movie frame, the time for the fastest baseball pitches to reach a batter, the time for a signal to reach a satellite. Develop a version of the Millisecond app in which a user can test themselves to see if they can react in the timescale of the selected events. You may want to create additional variables to accumulate hits and keep track of the player’s score.