

MIT APP INVENTOR

DAY 4

INSTRUCTIONS

Welcome to the course on MIT App Inventor. We will be using Zoom Application for delivering this course. Please adhere to the following instructions during the presentation.

- All of the participates other than the host are requested to mute (Alt+A) their microphone unless otherwise specified.
- Please use the chat window to type in your doubts and response to questions/tasks.

OVERVIEW

❖ Build two App games:


- Multiplication quiz.
- Mini Golf.

□ App 1 : Multiplication quiz

- App must work in this way: two numbers will pop up randomly, if user get it right app will give a message saying it's correct answer, otherwise incorrect answer has to pop up and show the correct answer.
- Drag four labels one by one and rename it both in components and properties.
- Drag a textbox.
- Drag a button.
- Drag a notifier from user inter face.

Viewer

Display hidden components in Viewer
 Check to see Preview on Tablet size.



Screen1

Text for Label1
X
Text for Label2
=

Submit

Components

- Screen1
 - Number1
 - Label1
 - Number2
 - Label2
 - TextBox1
 - Button1
 - Notifier1**

Rename Delete

Media

❑ Blocks editor

- Create variable, you need one variable for each two numbers that the app is multiplying and the correct answer.
- Set a procedure that reset a question whenever user enter the app, answer or give incorrect answer.
- To generate random numbers go to maths and pull out random integer.
- Two random blocks for number 1 and 2, and for the answer we need it to multiply it so pull out multiplication block from math.
- Update the number to the screen because the screen wont update itself.
- Instead copy and paste use call procedures and the app will call out the numbers to the screen.

Blocks Viewer

Built-in

- Control
- Logic
- Math
- Text
- Lists
- Colors
- Variables
- Procedures


Screen1

- Number1
- Label1
- Number2
- Label2
- TextBox1
- Button1
- Notifier1

```
initialize global num1 to 0
initialize global num2 to 0
initialize global answer to 0

when Screen1.Initialize
do call setQuestion

to setQuestion
do
  set global num1 to random integer from 1 to 12
  set global num2 to random integer from 1 to 12
  set global answer to get global num1 × get global num2
  set Number1.Text to get global num1
  set Number2.Text to get global num2
```



❑ Add submit button

- When button is pressed, if answer is correct then send a message, if not send a message too.
- To check the condition use (if statement).
- The app result and the user's must be same otherwise its incorrect.
- Create a question after answering each question.

Blocks Viewer

Built-in

- Control
- Logic
- Math
- Text
- Lists
- Colors
- Variables
- Procedures

Screen1

- Number1
- Label1
- Number2
- Label2
- TextBox1
- Button1
- Notifier1

Initialize global num1 to 0

Initialize global num2 to 0

Initialize global answer to 0

when Screen1.Initialize do call setQuestion


to setQuestion do

- set global num1 to random integer from 1 to 12
- set global num2 to random integer from 1 to 12
- set global answer to $\text{get global num1} \times \text{get global num2}$
- set Number1.Text to get global num1
- set Number2.Text to get global num2


when Button1.Click do

- if $\text{get global answer} = \text{TextBox1.Text}$
- then call Notifier1.ShowDialog
 - message "You got it!"
 - title "Yay!"
 - buttonText "Next question"
- else call Notifier1.ShowDialog
 - message join "Incorrect. The answer was" get global answer
 - title "Aww :{"
 - buttonText "Next question"

call setQuestion



□ App 2: Mini Golf

- Copy the image file to desktop. 
- Start a new project.
- Drag a canvas and change background color, Height to 300pixels and width to fill parent.
- Drag two balls rename them to ball and hole.
- Change ball radius to 10 and Z to 2.
- Change hole radius to 15 and color to black.
- Import image file to the golf area and use it as an obstacle.

Viewer

Display hidden components in Viewer

Components

- Screen 1
 - PuttingCanvas
 - Hole
 - Golfball
 - Obstacle

Rename Delete

Properties

ImageSprite1

Enabled

Heading

Interval

Picture

Rotates

Speed

Visible

X

Y

Z

Width

Media

obstacle.png

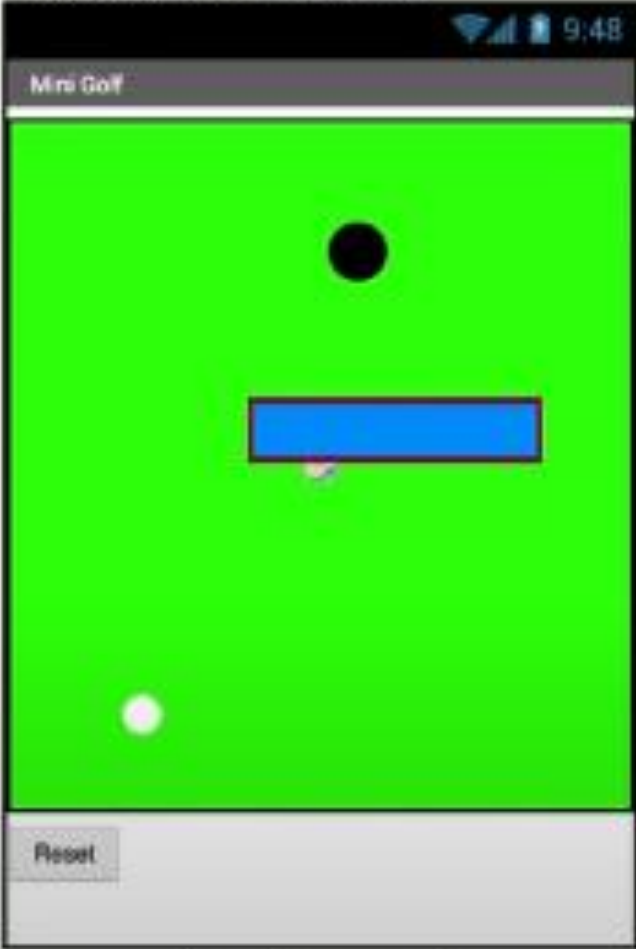
Upload File ...

□ Add clock


- Button to reset the game and rename it to reset.
- Drag a clock from user sensors.
- Change timerInterval to 100 to slow the golf ball.

Viewer

Display hidden components in Viewer



Non-visible components


 TimerClock

Components

- Screen1
 - PuttingCanvas
 - Hole
 - GolfBall
 - Obstacle
 - ResetButton
 - TimerClock

Properties

Clock1

TimerAlwaysFires

TimerEnabled

Timerinterval

Media

obstacle.png

❑ Blocks editor

- User want to throw the ball to reach the hole .
- Need to control the speed and destination.
- If speed of the ball is not convenient use math to increase or decrease speed.

The image shows a graphical user interface for a programming environment, divided into two main sections: "Blocks" and "Viewer".

Blocks Panel:

- Built-in:**
 - Control
 - Logic
 - Math
 - Text
 - Lists
 - Colors
 - Variables
 - Procedures
- Screen1:**
 - PuttingCanvas
 - Hole
 - Obstacle
 - GolfBall
 - RestButton
 - TimerClock
 - Any component

Viewer Panel:

The Viewer displays a script for a "GolfBall" object. The script is as follows:

```
when GolfBall . Flung
  x y speed heading xvel yvel
do
  set GolfBall . Heading = to get heading =
  set GolfBall . Speed = to get speed = x 7
```

The script is implemented using colored blocks: a yellow "when" block, a green "do" block, a green "set" block for heading, and a blue "set" block for speed that includes a multiplication operation with a constant value of 7.

□ Stop the ball

- The part before can't stop the ball.
- So if ball speed is greater than a particular number it will decrease or the ball will stop.

Blocks

- Built-in
 - Control
 - Logic
 - Math
 - Text
 - Lists
 - Colors
 - Variables
 - Procedures
- Screen1
 - PuttingCanvas
 - Hole
 - Obstacle
 - GolfBall
 - RestButton
 - TimerClock

Viewer

```
when GolfBall - .Flung
  x y speed heading xvel yvel
do
  set GolfBall - .Heading - to get heading -
  set GolfBall - .Speed - to (get speed - * 7)

when TimerClock - .Timer
do
  if GolfBall - .Speed - >= 0.5
  then set GolfBall - .Speed - to (GolfBall - .Speed - - 0.5)
  else set GolfBall - .Speed - to 0
```

The image shows a Scratch code editor with two scripts. The first script, triggered by the 'GolfBall' object's 'Flung' event, sets the ball's heading to its current heading and multiplies its speed by 7. The second script, triggered by the 'TimerClock' object's 'Timer' event, uses an if-then-else structure to decrease the ball's speed by 0.5 if it is greater than or equal to 0.5, or set it to 0 otherwise.

❑ Bounce off the edge and obstacle

- There was an example on the introduction of this course (MIT DAY 1) using event handler.
- Using event handler make it bounce off the obstacle.
- Change the ball heading to be opposite and easiest way to do that is by subtracting the ball heading .

Blocks

- Built-in
 - Control
 - Logic
 - Math
 - Text
 - Lists
 - Colors
 - Variables
 - Procedures
- Screen1
 - PuttingCanvas
 - Hole
 - Obstacle
 - GolfBall
 - RestButton
 - TimerClock

Viewer

```
when GolfBall - Flung
  x y speed heading xvel yvel
  do
    set GolfBall - Heading - to get heading -
    set GolfBall - Speed - to 0 + get speed - * 7

when TimerClock - .Timer
  do
    if GolfBall - Speed - >= 0.5
    then
      set GolfBall - Speed - to GolfBall - Speed - - 0.5
    else
      set GolfBall - Speed - to 0
```

```
when Obstacle - .CollidedWith
  other
  do
    set GolfBall - Heading - to 0 - GolfBall - Heading -
```

□ Goal

- Detect when the ball collided with the hole that means it reached goal .
- If the ball collided with the hole ball speed must be zero “stop”.
- Ball X and Y position should match the hole X and Y position.

Blocks

- Built-in
 - Control
 - Logic
 - Math**
 - Text
 - Lists
 - Colors
 - Variables
 - Procedures
- Screen1
 - PuttingCanvas
 - Hole
 - Obstacle
 - GolfBall
 - RestButton
 - TimerClock

Viewer

```

when GolfBall - Flung
  x y speed heading xvel yvel
  do
    set GolfBall - Heading - to get heading -
    set GolfBall - Speed - to 0 + get speed - * 7

when TimerClock - Timer
  do
    if GolfBall - Speed - > 0.5
    then set GolfBall - Speed - to GolfBall - Speed - - 0.5
    else set GolfBall - Speed - to 0
  
```

```

when Obstacle - CollidedWith
  other
  do
    set GolfBall - Heading - to 0 - GolfBall - Heading -

when GolfBall - CollidedWith
  other
  do
    if get other - == Hole -
    then
      set GolfBall - Speed - to 0
      set GolfBall - X - to Hole - X -
      set GolfBall - Y - to Hole - Y -
    
```

NEXT DAY

❖ Fidget Spinner :

- Using speed and friction in the app.
- Activity for student (build your own app).

THANK YOU!