GAME DEVELOPMENT WITH APP INVENTOR

HIDE AND SEEK APP

FEATURES OF THE APP

- This tutorial demonstrates how to build a simple game that uses animation, for example the Hide and Seek game.
- In this game, a rabbit pops up at random positions in a field.
- The player scores points by clicking the rabbit before it jumps away.

HIDE AND SEEK APP

CONCEPTS INTRODUCED

Following concepts can be achieved from this exercise:

- Component for touch-sensitive movable images.
- Clock component to move image
- Sound component to activate image
- Procedures to generate repeated behaviour
- Generating a random number

HIDE AND SEEK APP

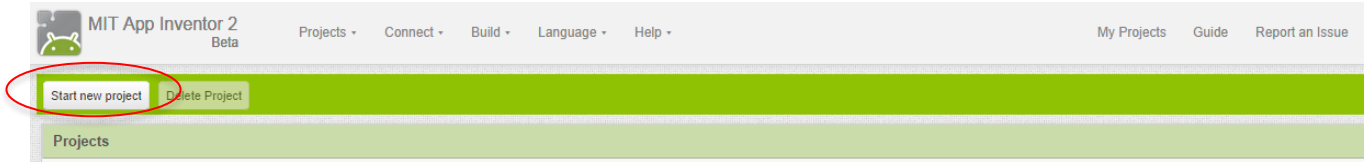
HIDE AND SEEK APP

INITIAL STEPS

1. Go to “http://ai2.appinventor.mit.edu/”

2. Sign-in with your Gmail account. If you do not have a Gmail account, you can create a free account at “https://www.google.com.sg/” .

3. Click ‘Start new project’



4. Enter Application Name for project : HideAndSeekApp

5. You’ll be presented with the Component Designer

BUILDING HIDE AND SEEK APP

COMPONENT DESIGNER

- 6 components are required.
- Drag the components specified in the table below

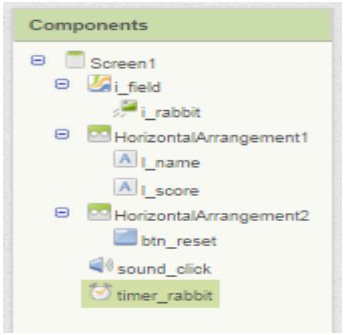
Components	Quantity	Description
Label	1	Displays the score based on the number of times the player hits the rabbit.
Canvas	1	Drawing canvas; the field where the rabbit moves
Button	1	Button to reset the new game.
Timer	1	Timer interval of the rabbit is shown on the screen
Sound	1	To indicate the player has clicked on the rabbit
Sprite	1	Sprite helps the image of the rabbit to move on the screen within a Canvas.

BUILDING HIDE AND SEEK APP

COMPONENT DESIGNER - CANVAS

Step1: Component designer

- Drag the components from the Palette to the Viewer area. Assign their names as shown in the diagram.



- Set the canvas (i_field) dimensions to 300 by 300 pixels (width by height). Set the background image as field.jpg.

- Set the label text of i_name to Score and the btn_reset to Reset .

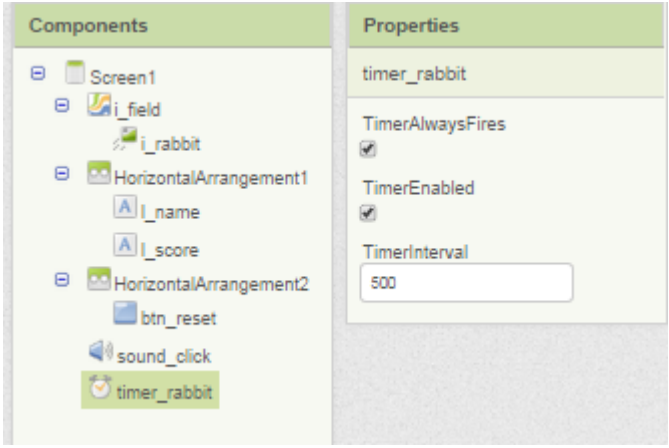
- Also add a sound component and name it click. You'll use sound to make the phone vibrate when the rabbit is clicked.

BUILDING HIDE AND SEEK APP

COMPONENT DESIGNER – CLOCK COMPONENT

THE RABBIT SHOULD POP UP AT RANDOM POSITIONS ON A PLAYING FIELD.

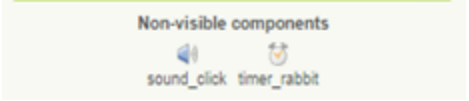
- Configure the rabbit to jump periodically, and you'll do this with the aid of a Clock component.



- The Clock component provides various operations dealing with time, like telling you what the date is. Here, you'll use the component as a timer that fires at regular intervals.

- The firing interval is determined by the Clock 's TimerInterval property.

- Drag a Clock component to the Viewer area; it will go into the non-visible components area.



  - Name it timer_rabbit .
  - Set the TimeInterval as 500 milliseconds to make the rabbit move every half second.
  - Ensure that TimeEnabled is checked

BUILDING HIDE AND SEEK APP

COMPONENT DESIGNER –

THE RABBIT SHOULD POP UP AT RANDOM POSITIONS ON A PLAYING FIELD.

- To add the moving rabbit we'll use a sprite .
- Sprites are images that can move on the screen within a Canvas .
- Each sprite has a speed and a heading, and also an interval that determines how often the sprite moves at its designated speed.
- Sprites can also detect when they are touched. In Hide and seek , the rabbit has a speed zero, so it won't move by itself.
- Instead, you'll be setting the rabbit's position each time the timer fires.

BUILDING HIDE AND SEEK APP

COMPONENT DESIGNER –

THE RABBIT SHOULD POP UP AT RANDOM POSITIONS ON A PLAYING FIELD.

- Component Designer – Sprite

- To add the moving rabbit we'll use a *sprite* .
- Sprites are *images that can move on the screen within a Canvas* .
- Each sprite has a speed and a heading, and also an interval that determines how often the sprite moves at its designated speed.
- Sprites can also detect when they are touched. In the app, the rabbit has a speed zero, so it won't move by itself. Instead, you'll be *setting the rabbit's position each time the timer fires*.

BUILDING HIDE AND SEEK APP

COMPONENT DESIGNER –

THE RABBIT SHOULD POP UP AT RANDOM POSITIONS ON A PLAYING FIELD.

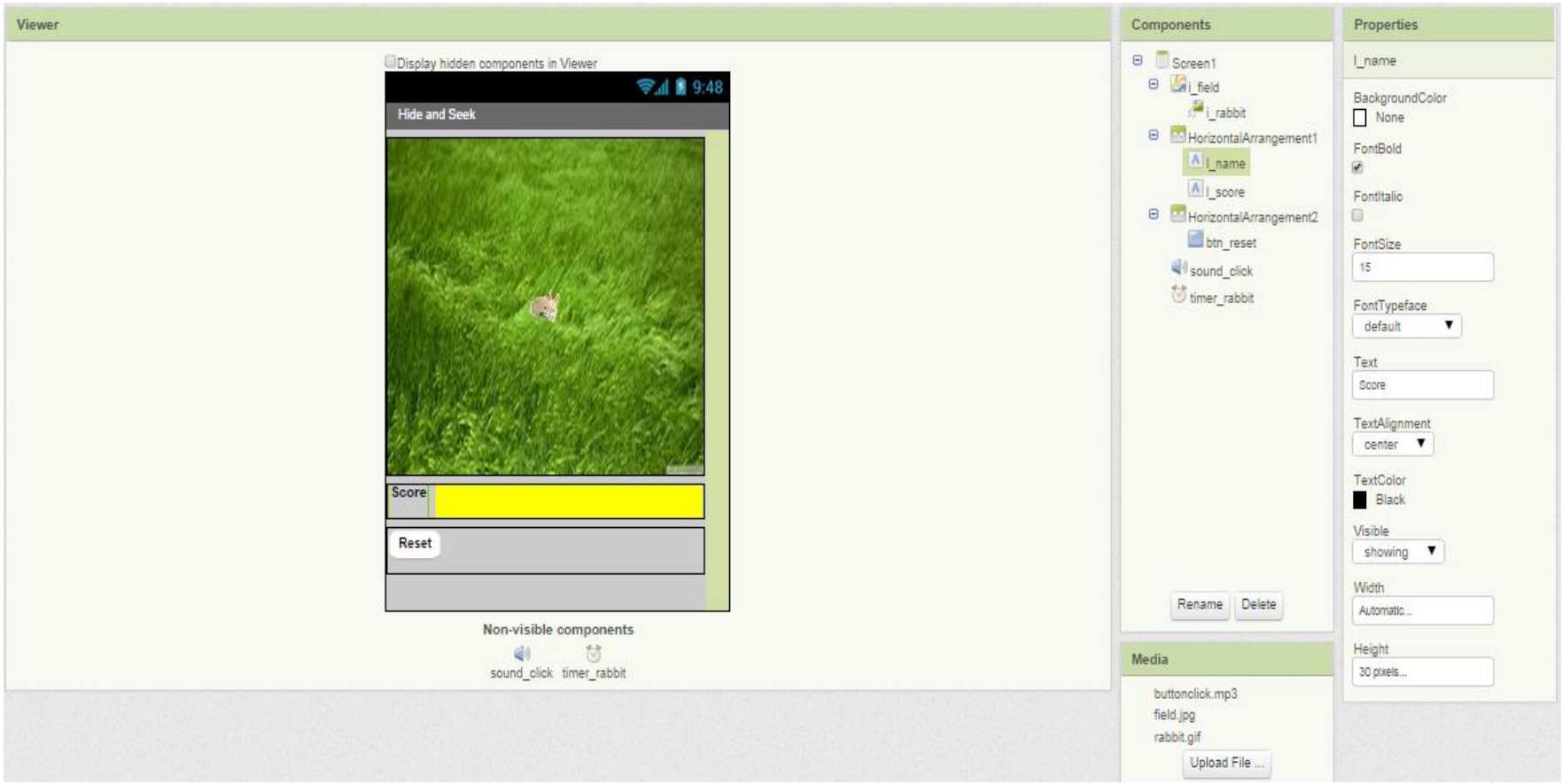
- Drag an ImageSprite component onto the Viewer . You'll find this component in the *Animation category* of the Palette .

- Place it within i_field area and rename as i_rabbit. Set these properties for the i_rabbit sprite:
  - Picture : Use rabbit.gif, which you downloaded to your computer at the beginning of this tutorial.
  - Enabled : checked
  - Interval : 500 The interval doesn't matter here, because the mouse's speed is zero: it's not moving by itself.
  - Heading : 0
    The heading doesn't matter here either, because the speed is 0.
  - Speed : 0.0
  - Visible : checked

BUILDING HIDE AND SEEK APP

COMPONENT DESIGNER –

THE COMPONENT DESIGNER MUST APPEAR AS SHOWN BELOW AFTER ADDING ALL THE COMPONENTS



BUILDING HIDE AND SEEK APP

BLOCK EDITOR

Step2: Create the logic to perform the functions as follows.

- The rabbit should pop up at random positions on a playing field.
- When the player is able to hit the rabbit within the time it disappears the score must increment by 1.
- When the reset button is clicked the score must become zero.

BUILDING HIDE AND SEEK APP

BLOCK EDITOR - PROCEDURE

Component Behavior - This introduces some new App Inventor ideas. The first is the idea of a procedure .

A procedure is a sequence of statements that you can refer to all at once as single command.

If you have a sequence that you need to use more than once in a program, you can define that as a procedure, and then you don't have to repeat the sequence each time you use it.

Procedures in App Inventor can take arguments and return values.

BUILDING HIDE AND SEEK APP

BLOCK EDITOR - PROCEDURE

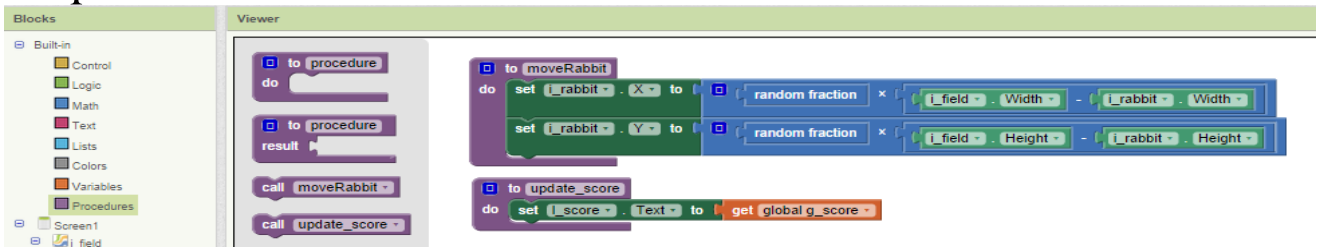
Define two procedures:

- *MoveRabbit* - moves the rabbit sprite to a new random position on the canvas.

- *UpdateScore* - shows the score, by changing the text of the ScoreLabel

- Start with MoveRabbit

- In the Blocks Editor, under Built-In , open the Definition drawer. Drag out a to procedure block



- Change the label procedure to MoveRabbit .

BUILDING HIDE AND SEEK APP

BLOCK EDITOR - MOVERABBIT

- In this case there will be two statements:
  - one to set the rabbit's *x* position
  - one to set its *y* position.

- In each case, you'll set the position to be a random fraction, between 0 and 1, of the difference between the size of the canvas and the size of the mouse. You create that value using blocks for random -fraction and multiplication and subtraction. You can find these in the Math drawer.

- Leave the argument socket for MoveRabbit empty because MoveRabbit procedure does not take any arguments.

BUILDING HIDE AND SEEK APP

BLOCK EDITOR - MOVERABBIT

- In this case there will be two statements:
  - one to set the rabbit's *x* position
  - one to set its rabbit's *y* position.

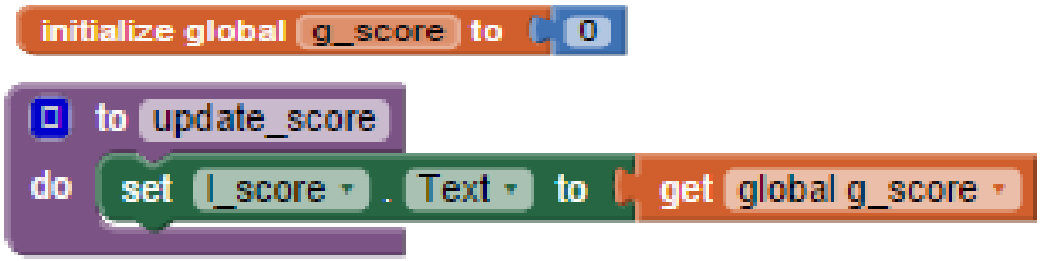
- In each case, you'll set the position to be a random fraction, between 0 and 1, of the difference between the size of the canvas and the size of the mouse. You create that value using blocks for random -fraction and multiplication and subtraction. You can find these in the Math drawer.

- Leave the argument socket for MoveRabbit empty because MoveRabbit procedure does not take any arguments.

BUILDING HIDE AND SEEK APP

BLOCK EDITOR – UPDATE SCORE

- Define a procedure *UpdateScore* that shows the score in ScoreLabel .
- Drag the procedure block from the Definition drawer and name it update_score.
- Drag the l_score.text block from the l_score drawer
- Drag the variable block from the definition drawer and change the block as g_score.
- Drag the number block from the math drawer and change the value to 0.



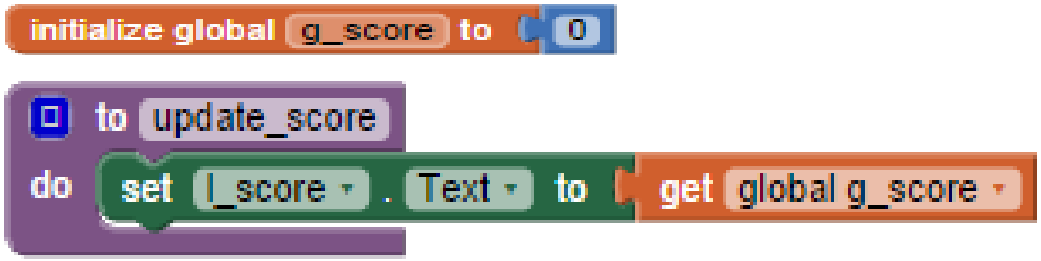
BUILDING HIDE AND SEEK APP

BLOCK EDITOR – UPDATE SCORE

The actual contents to be shown in l_score will be the value of the score .

- Drag out a text block from the Text drawer. Change the block to read "Score" rather than "text".
- Use a join block to attach this to a block that gives the value of the score variable. You can find the join block in the Text drawer

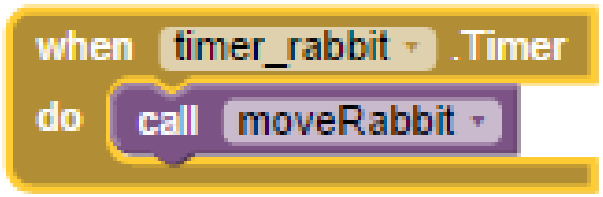
Update Score block should be as shown below



BUILDING HIDE AND SEEK APP

BLOCK EDITOR – RABBIT TIMER

- The next step is to make the rabbit keep moving.
- Here's where you'll use timer_rabbit .
- Clock components have an event handler called when timer_rabbit that triggers repeatedly at a rate determined by the TimerInterval .
- Drag call moveRabbit procedure.
- Set up timer_rabbit to call moveRabbit procedure each time the timer fires, by building the event handler like this:



Building Hide and Seek App

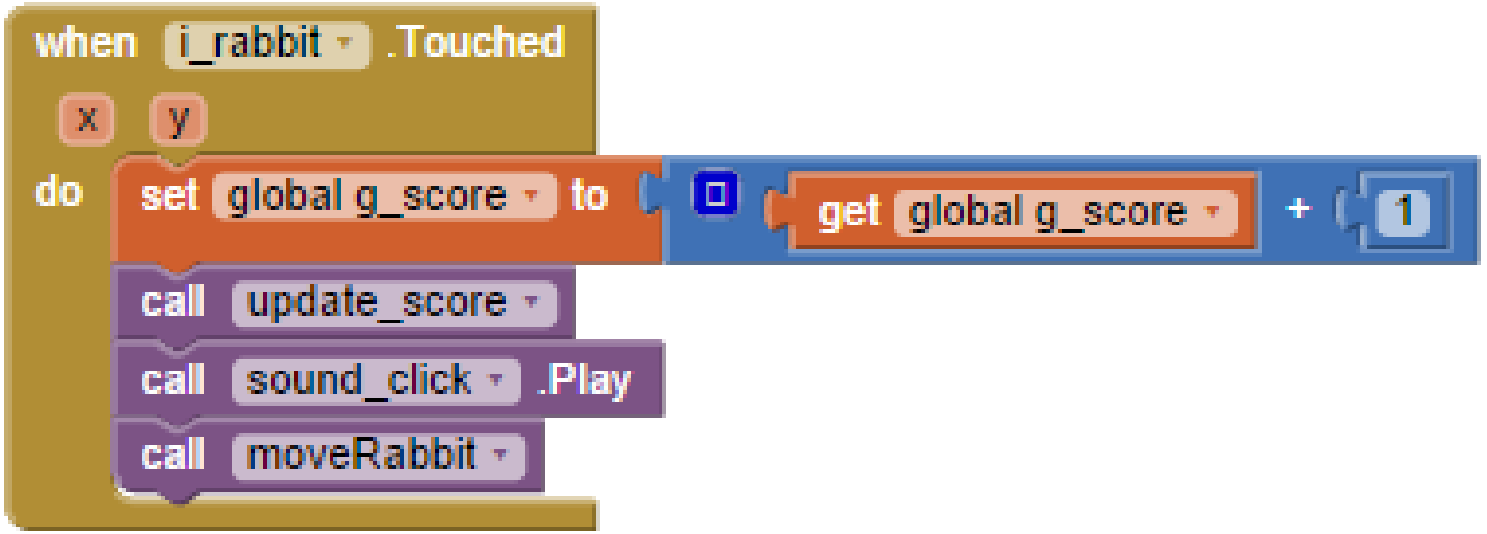
Block Editor – Rabbit Touch Event

- The program should increment the score each time the rabbit is touched.
- Sprites, like canvases, respond to touch events. So create a touch event handler for rabbit that is touched:
  - Increments the score.
  - Calls update_score to show the new score.
  - Calls sound_click to make a noise of the rabbit.
  - Calls moveRabbit so that the mouse moves right away, rather than waiting for the timer.

BUILDING HIDE AND SEEK APP

BLOCK EDITOR – RABBIT TOUCH EVENT

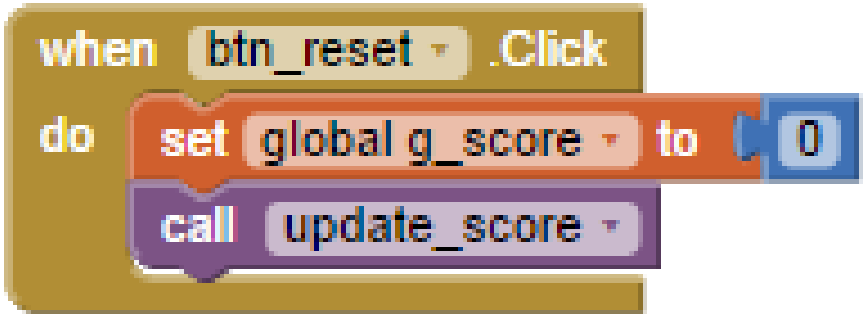
- The touch event block should be as shown below



BUILDING HIDE AND SEEK APP

BLOCK EDITOR – RESET BUTTON

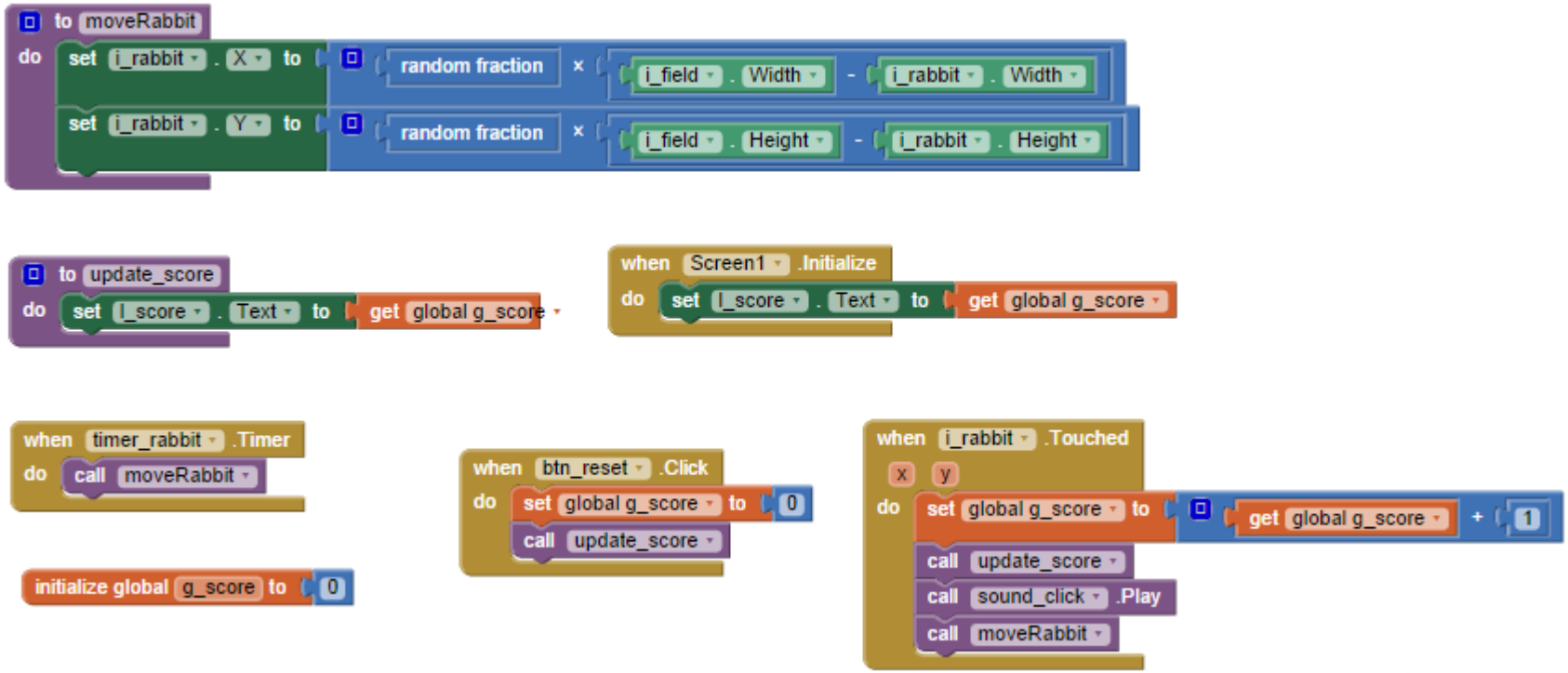
- One final detail is resetting the score when the reset button is clicked.
  - ➢ Reset button change the score to zero
  - ➢ Call update_score
- Here's how Reset button event blocks should look:



BUILDING HIDE AND SEEK APP

BLOCK EDITOR – RESET BUTTON

- All the blocks of the app are as shown below.



Building Hide and Seek App

Variations

Once you get the game working, you might want to explore some variations.

For example:

- Make the game vary the speed of the mouse in response to how well the player is doing. To vary how quickly the mouse moves, you'll need to change the timer_rabbit's Interval property.

- Keep track of when the player hits the rabbit and when the player misses the rabbit, and show a score with both hits and misses. To do this, you'll need do define touched handlers both for rabbit, same as now, and for i_field canvas . One subtle issue, if the player touches the rabbit, does that also count as a touch for the Canvas? The answer is yes. Both touch events will register.

BUILDING HIDE AND SEEK APP

SUMMARY

- Here are some of the ideas covered in this project:

- Sprites are touch-sensitive shapes that you can program to move around on a Canvas.

- The Clock component can be used as a time to make events that happen at regular intervals.

- Procedures are defined using to blocks.

- For each procedure you define, App Inventor automatically creates an associated call block.

- Calling a random-fraction produces a number between 0 and 1

- Text blocks specify literal text, similar to the way that number blocks specify literal numbers

- Type blocking is a way to create blocks quickly, by typing a block's name

THANK YOU

SEE YOU AGAIN