Coordinator Notes:

Module 6.3: Computers and Coding – Algorithms and Apps

In this Module, students build on their understanding of computer science, programing and coding.

- Students explore the concept of algorithms, and efficiency in coding.
- Students are introduced to 'App' design.
- Computer based and 'unplugged' activities are included.

Session Length:

This Module can be presented in different session durations per your needs.

Lesson plans are provided for:

- A 120 minute session, or, 2 x 60 minute sessions
- 45 minute, 75 minute, and 90 minute sessions

Technology:

PowerPoint: If you do not have access to a data projector and cannot display the PowerPoint presentation, we recommend that you print the most important slides before the session, and either enlarge them onto cardboard to use in place of slides, or create a booklet that students can share in small groups.

The session can be conducted without slides all together, but they offer visual aid in explanation of concepts.

Computers: The use of computers (or laptops, or tablets) and connection to the internet are important for this module.

If computers and the internet are not available for the session, choose the "unplugged" "Island Connection" Challenge Activity to replace the "App Inventor" Challenge.

Videos links: The suggested links to online videos within the session can be helpful with explanation. Notes have been included in the slides if there is an essential component to a video which the facilitator should discuss or demonstrate, if the video cannot be played.

Please read the Module 6 Risk Assessment before proceeding with the activity

Proudly developed by SMART with funding from Inspiring Australia

SMART An Australian Government Initiative Inspiring AUSTRALIA

Module 6.3: Computers and Coding – Algorithms and Apps

Overview

This session further explores algorithms as useful sets of rules and instructions for computers to aid in efficient completion of tasks, such as data sorting. If computers are available, students can explore creation of an App using MIT App Inventor. If computers are not available, students will explore algorithms through an interactive island connection challenge.

Encourage students to take part, share ideas and have a go. Reassure students that trial and error are important aspects of coding.

Content overview:

Concept / Activity	Session Duration (minutes)			
	120	90	75	45
Algorithms – introduces the word, and concept.	*	*	*	*
Types of Algorithms – sorting and searching algorithms.	*	*	*	*
Activity 6.3.1: Bubble Sort Algorithm. Interactive demonstration of the Bubble Sort Algorithm to accompany theory.	*	*	*	*
Activity 6.3.2: 21 Card Trick. Interactive demonstration of a 'magic' card trick which relies on an algorithm.	*	*	-	-
Introduction to 'Apps', and algorithms for app design.	*	*	*	-
Introduction to algorithms for network design	*	*	-	-
Challenge 6.3a: App Inventor Challenge Note: This activity requires computer / internet access.	*	*	*	-
Challenge 6.3b: Island Connection Challenge Unplugged activity, computers and internet not required. Introduces 'Minimal Spanning Tree' algorithms for networks. Recommended for 45 minute session duration.	~	~	~	*
Activity 6.3.3: Tablets of Stone Unplugged computers and internet not required. Introduces 'Network Communication Protocols'. Recommended extension activity for 120, 90 and 75 minute session durations, where Challenge 6.3b is also undertaken.	~	~	~	-

Proudly developed by SMART with funding from Inspiring Australia

Slides:

PowerPoint Slides are available to support the delivery of this module. Slides explain concepts visually, and include short, engaging videos relevant to the topic. A full list of slides and recommended inclusions for each session duration are provided in the table below. Appropriate slides are also noted in lesson plans for each duration.

PowerPoint Presentation: ‘M 6.3 - Master Slides 120 minute Session Duration’		**Session Duration (minutes)**			
Slide	**Content**	**120**	**90**	**75**	**45**
1	Introductory title page for Module 6.3	*	*	*	*
2	Discussion prompt slide: Algorithms	*	*	*	*
3	Video: Sorting and Searching Algorithms	*	*	*	*
4	Discussion: Bubble Sort Algorithm, and **Activity 6.3.1**	*	*	*	*
5	Discussion prompt slide: Algorithms are everywhere	*	*	*	-
6	Discussion prompt slide: Algorithms and Magicians	*	*	-	-
7	Discussion: 21 Card Trick, and **Activity 6.3.2**	*	*	-	-
8	Discussion prompt slide: Algorithms and Networks	*	*	-	-
9	Discussion prompt slide: Algorithms for Apps	*	*	*	*
10	Introductory Slide: **Challenge 6.3a:** App Inventor	*	*	*	*
11	Video: MIT APP Inventor Tutorial	*	*	*	*
12	Overview: MIT App Inventor Designer Editor	*	*	*	*
13	Overview: MIT App Inventor Block Editor	*	*	*	*
14	Notes: Challenge 6.3a	*	*	*	*
15	Prompt Slide: MIT App Inventor ‘Ball Bounce’ App	*	*	*	*
16	Introductory Slide: **Challenge 6.3b:** Island Connection	~	~	~	*
17	Overview: Minimal Spanning Trees	~	~	~	*
18-20	Notes and Rules: Challenge 6.3b	~	~	~	*
21-23	Solution for Challenge 6.3b	~	~	~	*
24	Session References	*	*	*	*

Note:

If computer and internet access not available, replace Challenge 6.3a with Challenge 6.3b for all session durations. Adjust slides to suit as follows:

- Slides for Challenge 6.3a: Slides 10-15.
- Slides for Challenge 6.3b: Slides 16-23.

Module 6.3 Computers and Coding – Algorithms and Apps

Lesson Plan

120 minute session or 2 x 60 minute sessions

High Tech: Use PowerPoint Presentation 'M6.3 - Master Slides'
Note: If computers and internet available, use Challenge 6.3a and slides 10-15.
If UNPLUGGED activities to be used instead, omit slides 10-15 and use slides 16 – 23.

Key Learning Area	Topic
Maths, Computer Science, Digital Technology	Algorithms

Timing	Running Time (hh:mm)	Procedure	Materials
		Lesson Introduction	
2 min	00:02	Welcome! Recap how we program. Introduce the word 'Algorithm'.	M6.3 PowerPoint (Slides 1-2)
		Body of Lesson **(Lesson 1, 2 x 60 minute sessions)**	
3 min	00:05	Discuss usefulness of algorithms for managing data. Introduce SORTING and SEARCHING algorithms. Show video. If video unable to be played, ensure you watch prior to session and explain steps to students.	M6.3 PowerPoint (Slide 3)
10 min	00:15	Introduce the Bubble Sort Algorithm. Discuss the concept. Students may choose to write a number array and practice the concept. Undertake Activity 6.3.1.	M6.3 PowerPoint (Slide 4) Optional: pencils, paper
2 min	00:17	Algorithms are everywhere! Brainstorm where algorithms might be useful tools.	M6.3 PowerPoint (Slide 5)
(5min)	(00:22)	*Optional Video, (allow an extra 5 minutes). 'What's an Algorithm By David J Malan'. Suitable for experienced coders and/or senior primary / high school students.*	

Proudly developed by SMART with funding from Inspiring Australia

SMART An Australian Government Initiative Inspiring AUSTRALIA

10 min	00:27 (00:32)	Algorithms and Magic. Introduce the 21 card trick. Demonstrate Activity 6.3.2. Support students to try out the card trick if time permits.	M6.3 PowerPoint (Slides 6-7) Playing cards, minimum 1 partial deck of 21 cards.
2 min	00:29 (00:34)	Discuss Networks, and the need for algorithms to assist in network design.	M6.3 PowerPoint (Slide 8)
3min	00:32 (00:37)	Discuss Apps. What is an App? Types of Apps? Discuss what Algorithms may be useful for App design.	M6.3 PowerPoint (Slide 9)
20 min	00:52 (00:57)	Introduce the MIT App Inventor Challenge* 6.3a Announce Challenge. Show the Video Tutorial. If video unable to be played, ensure you watch prior to session and explain steps to students. Step through the Designer and Block Editors. Support students to access computers/internet and navigate to the MIT App Inventor website. Students will need an email address to log-in to MIT App Inventor. Support students to work through steps to complete the Ball Bounce activity. ** Note: If computers are not available, introduce and start the Island Connection Challenge 6.3b, discuss Minimal Spanning Trees.* **(Break for 2 x 60 minute sessions)**	M6.3 PowerPoint (Slides 10-14) Computers, Internet access, Email Accounts, Ball Bounce Tutorial Sheets Android devices (phones, tablets) are optional to support this activity.

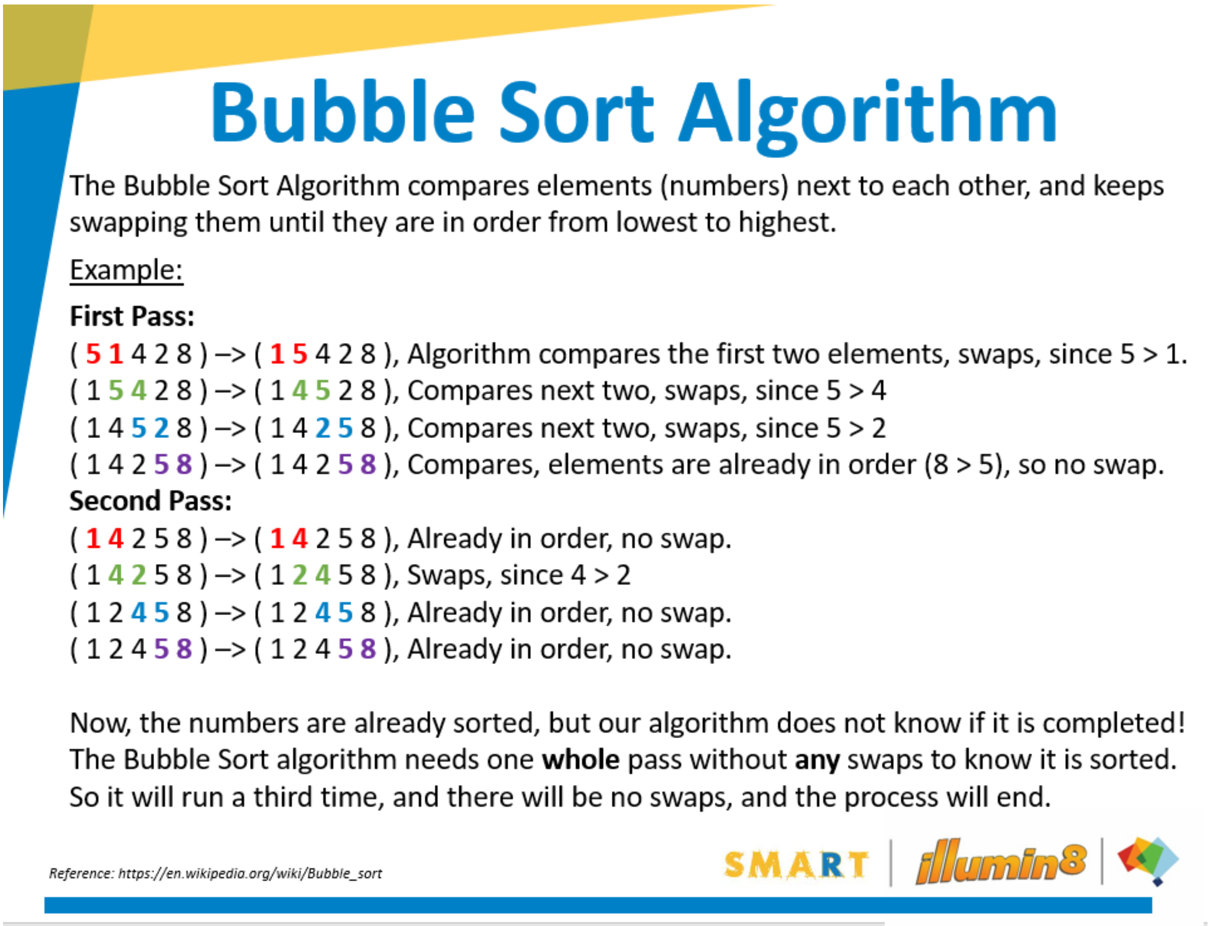
		(Lesson 2, 2 x 60 minute sessions)	
3 min	00:03/ 01:03	Recap the Ball Bounce Activity** Who was successful? What challenged were experienced? How were they overcome? Would anyone like to share their App / coding with the group? ***Note: Alternately, recap the Island Connection Challenge.*	PowerPoint M6.3 (Slide 15)
45 min	00:48/ 01:48	Mini Golf App Design*** Support students to access computers/internet and navigate to the MIT App Inventor website. Students will need an email address to log-in to MIT App Inventor. Support students to work through steps to complete the Mini Golf activity.	PowerPoint M6.3 (Slide 10) Computers, Internet access, Email Accounts, Mini Golf Tutorial Sheets Android devices (phones, tablets) are optional to support this activity.
5 min	00:53/ 01:53	Encourage students to try out each other's Mini Golf games. ****Note: Alternately, undertake unplugged activity 6.3.3 "Tablets of Stone"*	
		Lesson Conclusion	
7 min	01:00/ 02:00	Pack up and discussion, relevant to activities undertaken. Example: Did everyone's Mini-Golf game work in the same way? Did everyone's coding look the same, or were there differences?	Ensure all students have saved their work and logged out of their MIT App Inventor accounts, emails and computers.

Activity 6.3.1: Bubble Sort

Activity Notes:

Bubble Sort "is a simple sorting algorithm that repeatedly steps through the list to be sorted, compares each pair of adjacent items and swaps them if they are in the wrong order. The pass through the list is repeated until no swaps are needed, which indicates that the list is sorted." *Reference: https://en.wikipedia.org/wiki/Bubble_sort*

Talk through the Bubble Sort algorithm process using the slide provided (slide image below). The slide shows an example of the Bubble Sort algorithm sorting a set, or "array" of 5 numbers from smallest to largest.



Bubble Sort Algorithm

The Bubble Sort Algorithm compares elements (numbers) next to each other, and keeps swapping them until they are in order from lowest to highest.

Example:

First Pass:

(**5 1** 4 2 8) –> (**1 5** 4 2 8), Algorithm compares the first two elements, swaps, since 5 > 1.
(1 **5 4** 2 8) –> (1 **4 5** 2 8), Compares next two, swaps, since 5 > 4
(1 4 **5 2** 8) –> (1 4 **2 5** 8), Compares next two, swaps, since 5 > 2
(1 4 2 **5 8**) –> (1 4 2 **5 8**), Compares, elements are already in order (8 > 5), so no swap.

Second Pass:

(**1 4** 2 5 8) –> (**1 4** 2 5 8), Already in order, no swap.
(1 **4 2** 5 8) –> (1 **2 4** 5 8), Swaps, since 4 > 2
(1 2 **4 5** 8) –> (1 2 **4 5** 8), Already in order, no swap.
(1 2 4 **5 8**) –> (1 2 4 **5 8**), Already in order, no swap.

Now, the numbers are already sorted, but our algorithm does not know if it is completed! The Bubble Sort algorithm needs one **whole** pass without **any** swaps to know it is sorted. So it will run a third time, and there will be no swaps, and the process will end.

Reference: https://en.wikipedia.org/wiki/Bubble_sort

SMART | illumin8

If time permits encourage students to try out the concept with a number array of their creation, provide pens / pencils and paper for students to practice the algorithm (allow for 2 – 5 minutes).

To complete the group activity:

Ask all students to stand up and form a line. Next, ask the students to sort themselves into height order, using only the Bubble Sort Algorithm. They will need to choose which end they are starting at, then compare the two adjacent people, decide if they swap, and repeat along the line. Then, repeat the process, until the line of students is in height order.

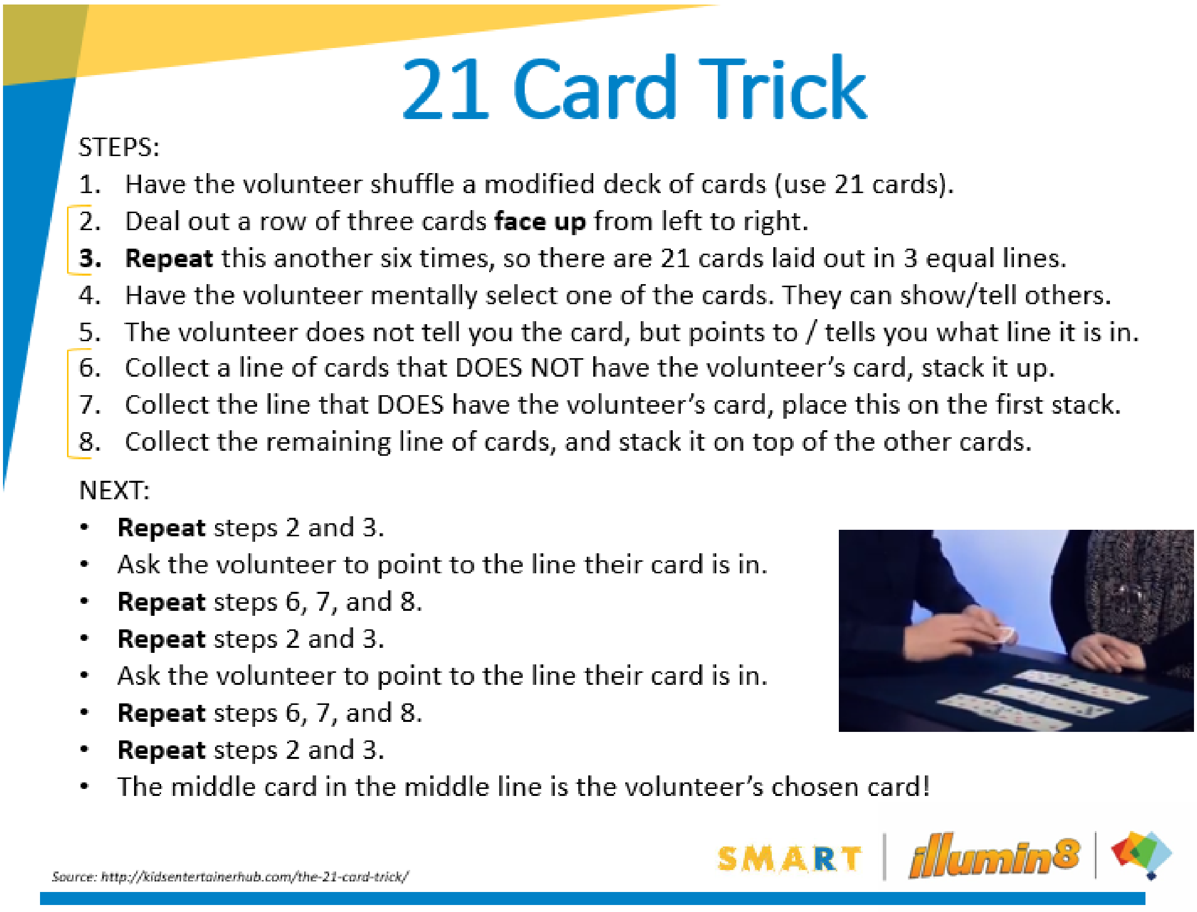
Note: You could form 2 or 3 lines for larger groups, perhaps making it a competition to see which group can sort into height the fastest.

Activity 6.3.2: 21 Card Trick

Activity Notes:

Magic tricks, especially card tricks, often use algorithms! Let's take a look at a magic trick called the "21 card trick" and see if we can observe the algorithm.

Perform the 21 card trick with a student volunteer. The steps are outlined on the slide image below, and a video of this trick being performed available online at: https://youtu.be/KpMWlhrYO4A



21 Card Trick

STEPS:

1. Have the volunteer shuffle a modified deck of cards (use 21 cards).
2. Deal out a row of three cards **face up** from left to right.
3. **Repeat** this another six times, so there are 21 cards laid out in 3 equal lines.
4. Have the volunteer mentally select one of the cards. They can show/tell others.
5. The volunteer does not tell you the card, but points to / tells you what line it is in.
6. Collect a line of cards that DOES NOT have the volunteer's card, stack it up.
7. Collect the line that DOES have the volunteer's card, place this on the first stack.
8. Collect the remaining line of cards, and stack it on top of the other cards.

NEXT:

- **Repeat** steps 2 and 3.
- Ask the volunteer to point to the line their card is in.
- **Repeat** steps 6, 7, and 8.
- **Repeat** steps 2 and 3.
- Ask the volunteer to point to the line their card is in.
- **Repeat** steps 6, 7, and 8.
- **Repeat** steps 2 and 3.
- The middle card in the middle line is the volunteer's chosen card!

Source: http://kidsentertainerhub.com/the-21-card-trick/

Activity Discussion:

Ask for a second volunteer and perform the card trick again, whilst explaining / discussing the algorithm for the trick. You'll see some "repeats" and when to "stop".

There are many ways to write the Algorithm for this magic trick! It could be written like this:

- Perform Steps 1 to 8 once.
- Then, repeat Steps 2 and 3 once.
- Ask volunteer which line their card is in.
- Then, repeat Steps 6, 7, and 8 once.
- Then, repeat Steps 2 and 3 once.
- Ask volunteer which line their card is in.
- Then, repeat Steps 6, 7, and 8 for the final time.
- Then, repeat Steps 2 and 3 for the final time.
- Select the middle card from the middle line. End.

Challenge M6.3a – App Inventor

Coordinator Notes

Scoring:

A scoring mechanism is not included in this module. There are no right or wrong answers!

Before hosting the session:

Visit: http://appinventor.mit.edu/explore/beginner-tutorials-short.html

Work through the tutorials to ensure you are familiar with the MIT App Inventor platform. Start with: 'Talk to Me' to familiarise yourself with starting a new project.

If Android phones / tablets will be available for app testing in your session, ensure you can install the MIT App Inventor and a QR Code reader on an Android phone or tablet. If an Android phone / tablet will not be available for your session, practice setting up the Android Emulator on a computer.

Printable step-by-step tutorial notes for getting started are available for download at:
http://appinventor.mit.edu/explore/sites/all/files/hourofcode/TalkToMePart1.pdf
It is recommended you take a copy of these step-by-step instructions to the session.

Activity Notes:

- Before students arrive check that each device: computers, laptops and / or tablets to be used in the session have power and have access to the internet. At least 90 minutes of available battery power is recommended.
- Sound is optional, however some component blocks can enable sounds to be included in the programming. Check that available speakers are working, if you'll be encouraging the inclusion of sound elements.
- Open up an internet browser on each computer / device, and pre-type in the link for the activity. Check for internet connection.
- Link: http://appinventor.mit.edu/explore/get-started
- Students may like to work in pairs, individually or small groups of 3, depending on the computers / devices available.
- Students will need an email address (school or Gmail) to log-in and save their work.
- Google accounts for Gmail can be created online at: https://accounts.google.com/SignUp
- Students should complete the Ball Bounce activity prior to moving on to the Mini Golf Activity.
- Encourage students to try out each other's games and view each other's coding, if time permits.

Resources:
It is recommended the Tutorial Sheets for **'Ball Bounce'** be printed and provided for students to support participation in this activity. Print at least 1 set of tutorial sheets per group. They are available for download at:
http://appinventor.mit.edu/explore/sites/all/files/hourofcode/BallBounceTutorial.pdf
The **Mini Golf** Game App Design is a logical progression / extension from the Ball Bounce App. Tutorial sheets have been provided on the next page. Print at least 1 set of tutorial sheets per group.

Full instructions for the Mini Golf and an example downloadable App file are available online at:
http://appinventor.mit.edu/explore/ai2/minigolf.html

Mini Golf Game Tutorial Sheets

Adapted from http://appinventor.mit.edu/explore/ai2/minigolf.html
By the University of Newcastle's SMART Program

STEP 1: Create a Mini Golf Screen

We'll build this app in stages, adding a little bit of the game at a time.

Log into the App Inventor and open a new project. Give it a name like 'MiniGolfNAME' (put your name here instead of 'NAME').

When the Design window opens, notice that App Inventor automatically names the screen "Screen1". You can set the Title of the screen, which will show up in the top bar of the app. Think of a title related to Mini Golf, or feel free to use the suggested title "Fling It Mini Golf", and type it into the Properties pane on the right side of the Designer.

In the Screen Properties (shown in right-hand pane): Uncheck the checkbox labelled "Scrollable", so that the screen will not scroll when the app is running. Screens that are set to scroll do not have a height. We'll need our screen to have a defined height in order to set up the golf course properly.

STEP 2: Add the following components to your app in the Designer Editor

Note: 'Palette Group' will help direct you to where the components are found in the menu.

Component Type	Palette Group	What You'll Name It	Purpose	Properties
Canvas	Drawing and Animation	Canvas1	The canvas serves as the golf course	Height: 300 Width: FillParent Background Colour: Green (or whatever you like!)
Ball	Drawing and Animation	GolfBall	This is the ball the player will fling to try to hit the Hole	Radius = 10 Colour: White (or your choice!) Speed: 0 Interval: 1 (ms) Z = 2 (when sprites are overlapping, the one with the higher z will appear on top)
Ball	Drawing and Animation	Hole	This will be the target for the GolfBall	Radius = 15 Colour: Black Speed: 0
Clock	Sensors	Clock1	The clock will fire continuously to control the movement of the ball	Timer Always Fires Timer Enabled Timer Interval: 100

Proudly developed by SMART with funding from Inspiring Australia SMART An Australian Government Initiative Inspiring AUSTRALIA

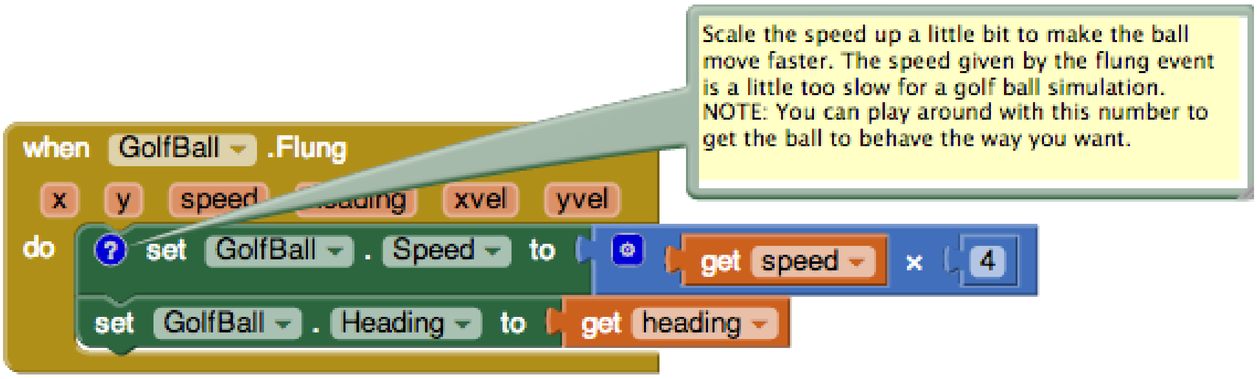
STEP 3: Open the Blocks Editor and program the behaviour of the Ball

First, use the GolfBall.Flung event handler to move the golf ball when it is flung. Drag this block into the viewer.

Notice how this block is an event handler, and can take in 6 different arguments:

- ***x,*** the x position on the Canvas grid of the user's finger
- ***y,*** the y position on the Canvas grid of the user's finger
- ***speed,*** the speed of the user's flinging gesture
- ***heading,*** the direction (in degrees) of the user's fling gesture
- ***xvel,*** the speed in the x direction of the user's fling
- ***yvel,*** the speed in the y direction of the user's fling

Essentially, you want to set the GolfBall's **speed** and **heading** to match the speed and heading of the player's fling gesture. You may want to scale up the speed a little bit, because the speed of the fling is a little slower than how a golf ball would move. You can play with this "scaling factor" to make the ball more or less responsive to a fling.



```
when GolfBall.Flung
  x  y  speed  heading  xvel  yvel
do  set GolfBall.Speed to  get speed × 4
    set GolfBall.Heading to  get heading
```

Scale the speed up a little bit to make the ball move faster. The speed given by the flung event is a little too slow for a golf ball simulation. NOTE: You can play around with this number to get the ball to behave the way you want.

STEP 4: Test it out!

Open the Emulator or connection to your android device, and see how your game is looking so far. Can you see any problems?

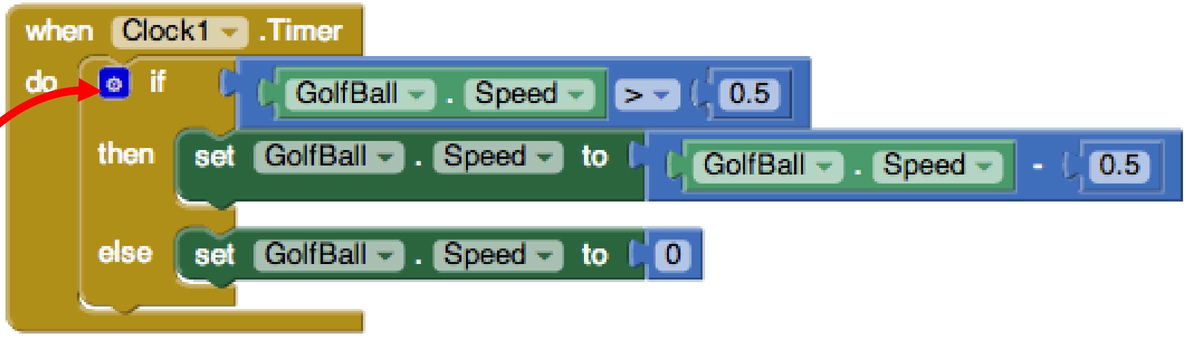
So far, the ball should move when you fling it, but it won't bounce off the edges, and it will keep moving forever.

It also doesn't do anything if you get the ball into the hole. Continue through the tutorial to solve these problems and add to your game.

STEP 5: Program the behaviour of the Clock

Use timer event to slow ball down so it doesn't bounce around forever.

Each time the clock fires, it will reduce the speed of the ball slightly. Notice that if the ball is not moving then these blocks will do nothing. If you don't have this then the ball will just bounce forever. You'll need to use the **mutator function** to change the if block into an if-else block.



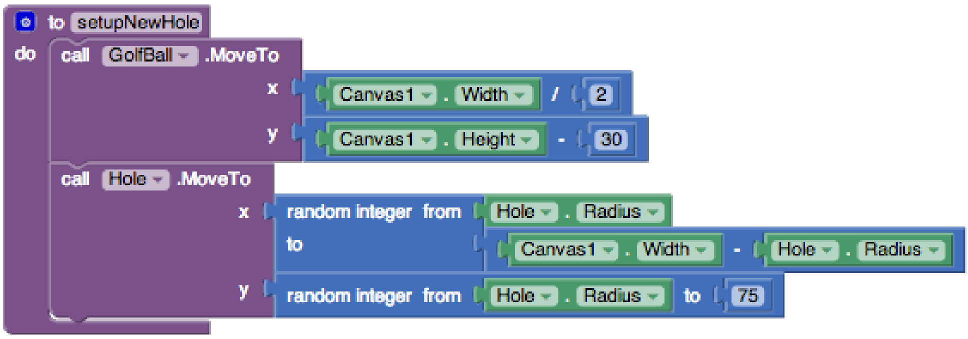
```
when Clock1 .Timer
do  if    GolfBall . Speed  >  0.5
    then  set GolfBall . Speed to  GolfBall . Speed  -  0.5
    else  set GolfBall . Speed to  0
```

*What does a **mutator** do? App Inventor 2 introduced a new feature that allows certain blocks to expand, shrink, or even change their functionality. Any block that has a blue box with a white gear on top is considered a **mutator block.***

Mutators change shape. By clicking the blue icon, the user can drag additional smaller blocks into the larger mutator block, thus changing the shape and functionality of the original block. Clicking the icon again will minimize the extra blocks window, and show the modified block. For more information, check out the mutators page *at:* *http://appinventor.mit.edu/explore/ai2/support/concepts/mutators*

STEP 6: Program a new procedure called SetupNewHole

This procedure will be called when a hole is scored and the ball has to be placed back at the starting point. Note that the Hole.MoveTo block sets the hole up in a new random location for the next play.



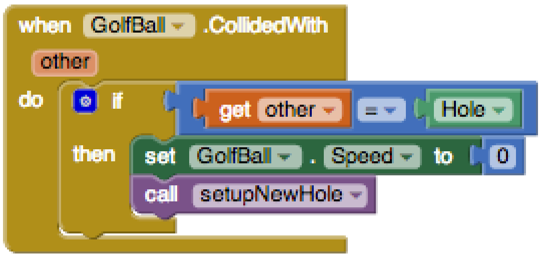
```
to setupNewHole
do  call GolfBall .MoveTo
              x  Canvas1 . Width  /  2
              y  Canvas1 . Height  -  30
    call Hole .MoveTo
              x  random integer from  Hole . Radius
                                to    Canvas1 . Width  -  Hole . Radius
              y  random integer from  Hole . Radius  to  75
```

STEP 7: Program the Behaviour of the Hole

When the ball collides with the hole, the ball disappears and resets at the bottom of the screen.

Note: When you first drag out the GolfBall.CollidedWith event handler, the named parameter is called "other". Notice that the if then block tests to see if the object involved in the collision with the golf ball (other) is the black ball sprite representing the hole. You can't just put a text block with the word "Hole" in it, you must use the Hole block, that can be found in the drawer for the Hole image sprite. Do not use a text block here.

You can also set the ball to stop over the hole rather than beside it. To do this, set the golf ball to the hole's X and Y coordinates.



```
when GolfBall .CollidedWith
  other
do  if   get other = Hole
    then set GolfBall . Speed to 0
         call setupNewHole
```

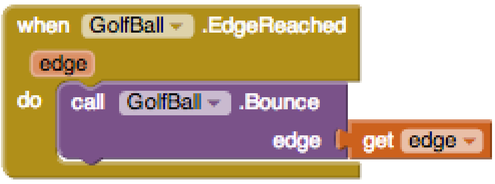
STEP 8: Test it out!

Open the Emulator or connection to your android device, and see how your game is looking so far. Can you see any problems? When you fling the ball, it should move in the direction of your fling, with a speed similar to the strength of your fling. The ball should slow down as it moves, eventually stopping. When the ball hits the hole, the ball should reset at the bottom of the screen, and the hole should move to a new random location.

If your game isn't working as you would expect, investigate the part of the game which isn't working and go back to the code to see how you could fix it!

Does your ball get stuck if it hits the edge?

This is easy to fix with the when EdgeReached event. Note that you can find the "edge" value block by using a get block and selecting "edge" from the dropdown.



```
when GolfBall .EdgeReached
  edge
do  call GolfBall .Bounce
              edge  get edge
```

Double check to make sure your code is right: fling the ball a few times and see that that ball now bounces off the edges of the course.

Proudly developed by SMART with funding from Inspiring Australia SMART An Australian Government Initiative Inspiring AUSTRALIA

Congratulations Coder!

You have just created your first Mini Golf Gaming App!

Now… let's see how we can improve on the basic game!

See if you can add the following to your app:

A. An obstacle which the ball bounces off
B. A reset button which makes the ball go back to the start of the course
C. A sound which plays to alert the player that they have won
D. … ?? Be creative, what else could you add to improve your game?

Hints & Tips:

A. To make an obstacle, you need to add an image sprite (found in the Drawing and Animation palette) called 'obstacle'. It could be any image you draw in Paint.

Then think about which blocks you could use to tell the ball to bounce off the obstacle… (You need a brown event handler block relating to the obstacle).

B. Firstly you need to add a button. Have a look through the palettes in the Designer menu to find a button. Then rename the button something like 'Reset'. Now go to your blocks editor, and instruct the computer what it needs to do when the button is pressed.

C. Add a sound in the Designer screen. You can upload any sound you like! Now go back to the Blocks editor and see if you can work out how to link the sound with the ball going into the hole during your game.

D. Be creative and add another component to your game. You can add a new golf course, a 'congratulations you won!' message, a score keeper, or even add a sensor to use as part of the game, such as the accelerometer or gyroscope (note: you will need to be connected to a tablet or phone, rather than the emulator, to use sensors).

For further tips on adding extra features to your Mini Golf app, visit:
http://appinventor.mit.edu/explore/ai2/minigolf.html

Challenge M6.3b – Island Connection

This UNPLUGGED activity explores ‘Minimum Spanning Trees’. It does not require computers or internet. Use this activity in place of the App Inventor activities, if computers and internet access are not available for your session.

Scoring:

- Points will be awarded for teamwork
- Points will be awarded for having a complete map of all routes between islands
- Points will be awarded for discovering the “minimum spanning tree” of the tropical paradise

Activity notes:

- This challenge is best performed in pairs.
- Modifying the challenge to have fewer or more islands can simplify the session for younger groups or extend the activity for more complexity for older groups.
- For a large group, it may be necessary to have two or more different island sets.
- This challenge involves students creating an ‘Island Connection’ for their new island tour company. Students must travel the islands, identifying all possible routes, and then identify the “minimum spanning tree”.

PART 1:

- There are a total of 7 Islands. At each island, students choose either ship Route A, Route B or at some islands, Route C.
- Each island is best represented by a person holding up a card with the island name (see note 1 below).
- Students choose a card holder to start at, and ask the card holder for the details of either ship route A, B or C. The card holder will tell them which island to go to next, and the cost of the journey, corresponding with their selected route A, B or C.
- The students should note down the island they start at, the route they choose (A, B or C) the cost of the route, and which island this takes them to.
- Not all islands will have a Route C available!
- Students should continue the exercise until they are confident they have learned / tried out all routes between the islands.
- Students should then work with their team mate to draw a ‘Tropical Paradise’ map of the islands, and mark down all of the possible routes between them.

Note 1: If there are not enough people to represent the islands, then an “island” may consist of 3 to 4 pieces of paper:

- 1st sheet: containing the name of the island,
- 2nd sheet: containing ship route A details (with “SHIP ROUTE A” written on one side of the page, face up, and the cost and next island destination on the face down side)
- 3rd sheet: containing ship route B details
- 4th sheet: containing ship route C details (if route C exists for that island).

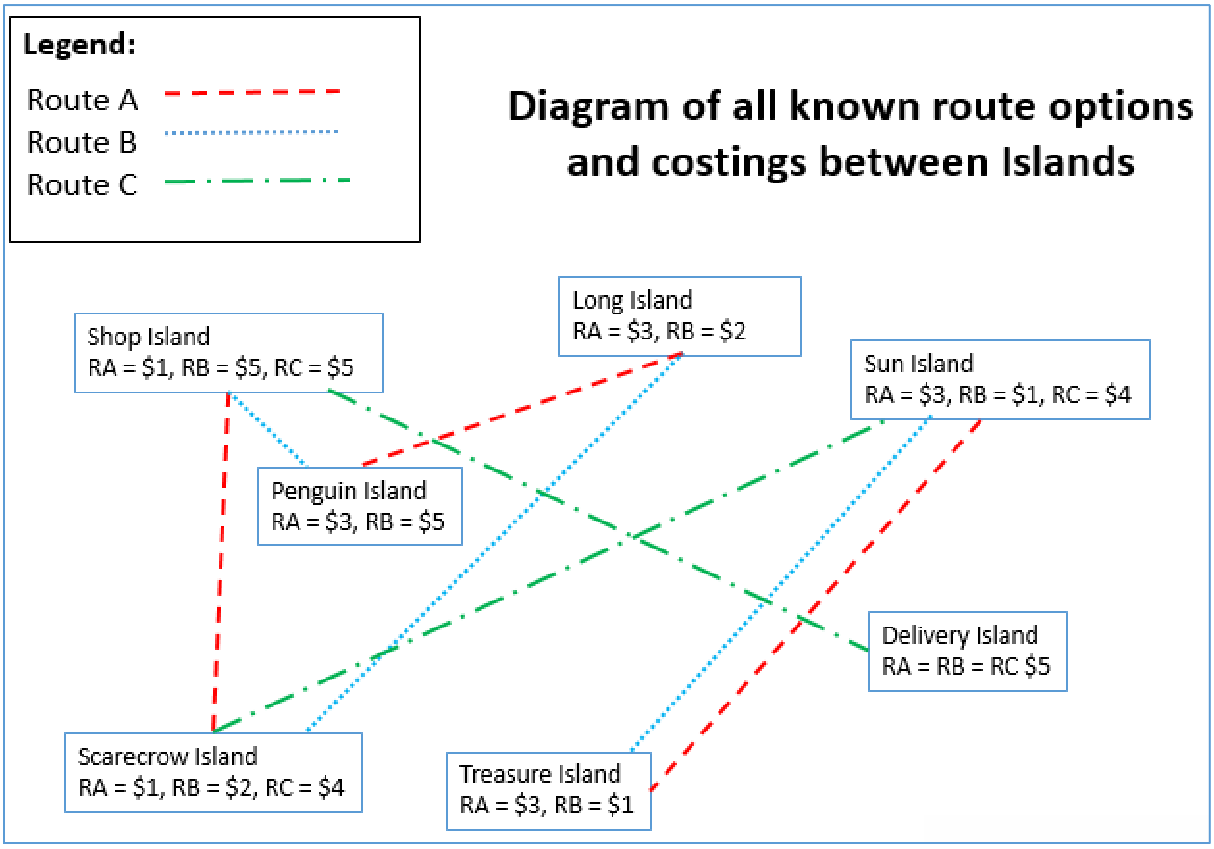
PART 2:

- If time allows, encourage students who have completed their maps to try and identify the most effective, cheapest way to travel to all of the islands that is, the 'minimum spanning tree'.
- Students should only travel between two islands once.
- Students can choose to start at any island.
- Students should tally up the total cost of the routes they include in their 'minimal spanning tree'.
- Students who complete this task may like to see if their result is different if they start from another island. Can they get the costs down?
- Students who complete this task may like to try and write out an algorithm to represent their minimum spanning tree.

Rules:

- Students will need to work in pairs (with their tour company partner!) to discover the "best", cheapest way to visit all islands.
- Students will be given a map of the islands, and information on the possible sailing routes between them. There are a total of 7 islands.
- Students will need to discover all of the different sailing routes between the islands and draw them on their map. Each route will have a different cost.
- Students may begin sailing from any island!
- Once started, students may only move between islands using the known sailing routes. Each island has 2 or 3 possible routes to choose from (A, B or C).
- Students may choose one route from each island at a time.
- Running is not allowed on the boats! No running between islands, only sailing!

Keep the information on island routes and costings hidden from the students who are working on creating their maps. Only provide the information to the island card holders, or, to create sheets for each location if card holders will not be participating in the activity.



Island cards for printing are available on the next pages.

Proudly developed by SMART with funding from Inspiring Australia
SMART An Australian Government Initiative Inspiring AUSTRALIA

Challenge 6.3b: Island Connections – Island Cards

Print cards, and cut along the horizontal lines.

Then, fold cards in in half along the vertical lines, so on one side (the face up side) the Island and the route name is visible, and on the other side (the face down side) the route cost and destination is visible.

Shop Island Route A	Destination: Scarecrow Island Cost: $1
Shop Island Route B	Destination: Penguin Island Cost: $5
Shop Island Route C	Destination: Delivery Island Cost: $5

Scarecrow Island Route A	Destination: Shop Island Cost: $1
Scarecrow Island Route B	Destination: Long Island Cost: $2
Scarecrow Island Route C	Destination: Sun Island Cost: $4
Penguin Island Route A	Destination: Long Island Cost: $3

Proudly developed by SMART with funding from Inspiring Australia

An Australian Government Initiative

Penguin Island Route B	Destination: Shop Island Cost: $5
Long Island Route A	Destination: Penguin Island Cost: $3
Long Island Route B	Destination: Scarecrow Island Cost: $2
Sun Island Route A	Destination: Treasure Island Cost: $3

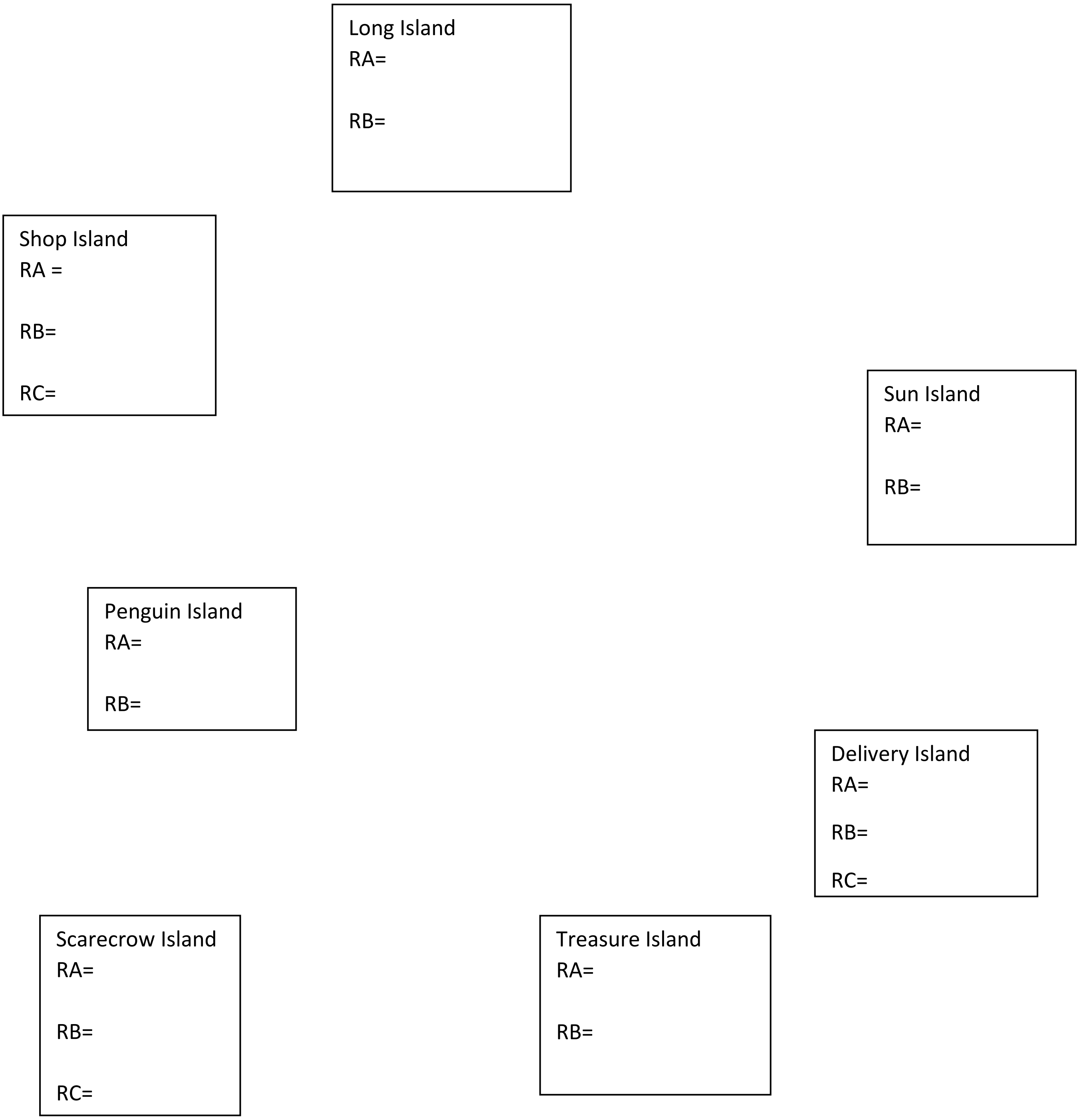
Sun Island Route B	Destination: Treasure Island Cost: $1
Sun Island Route C	Destination: Scarecrow Island Cost: $4
Treasure Island Route A	Destination: Sun Island Cost:$3
Treasure Island Route B	Destination: Sun Island Cost: $1

Delivery Island Route A	Destination: Shop Island Cost: $5
Delivery Island Route B	Destination: Shop Island Cost: $5
Delivery Island Route C	Destination: Shop Island Cost: $5

Proudly developed by SMART with funding from Inspiring Australia

SMART An Australian Government Initiative Inspiring AUSTRALIA

Challenge 6.3b: Island Connections - Tropical Paradise Map



Long Island
RA=

RB=

Shop Island
RA =

RB=

RC=

Sun Island
RA=

RB=

Penguin Island
RA=

RB=

Delivery Island
RA=

RB=

RC=

Scarecrow Island
RA=

RB=

RC=

Treasure Island
RA=

RB=

Challenge 6.3b: Island Connections – Score Sheets

Tour Company Name:		
Team Member Names:		
Scoring	**Maximum Available**	**Points allocated**
Teamwork	5 points	
No running!	5 points	
Completed map (1 point for each sailing route shown)	8 points	
Minimum spanning tree Total cost to visit all islands: Less than $14 = 20 points $14 = 15 points More than $14 = 10 points	10, 15 or 20 points	
Final Score	**Max 38**	

Tour Company Name:		
Team Member Names:		
Scoring	**Maximum Available**	**Points allocated**
Teamwork	5 points	
No running!	5 points	
Completed map (1 point for each sailing route shown)	8 points	
Minimum spanning tree Total cost to visit all islands: Less than $14 = 20 points $14 = 15 points More than $14 = 10 points	10, 15 or 20 points	
Final Score	**Max 38**	

Activity 6.3.3: Tablets of Stone

*This activity is adapted from: **CSUnplugged.org** CS Education Research Group at the University of Canterbury, NZ, Creative Commons BY-NC-SA 4.0 licence.*

This UNPLUGGED activity explores 'Network Communication Protocols'. It does not require computers or internet.

Computers talk to each other over the internet via messages. However, the internet is not reliable and sometimes these messages get lost. There are certain bits of information we can add to messages to make sure they are sent. This information makes up a protocol.

In this activity students consider how different methods of communication operate successfully. By looking at rules and procedures in place, students are introduced to communication protocols. By working through a role-play scenario, pupils test their own protocol operating in an unreliable environment similar to that found in packet switching on the Internet, specifically, TCP/ IP.

Background Story (Optional)

"In an ancient city there are a number of very important Governors. These Governors decide how the city is run and make very important decisions. They each live in different houses all over the city.

The Governors often want to communicate, they need to send and receive messages all over the city. Governors are identified by their house number and they all have access to a group of messengers whose job it is to deliver the messages.

The only way to send messages is by writing them on large rectangular stone tablets, which the messengers carry to their destination. The stone tablets are of a fixed size and can only fit 6 pieces of information on them. One piece of information can be one letter or one number. Messages are often split over a number of tablets, and as these tablets are very heavy they can only be carried one at a time.

The messengers cannot be trusted to always deliver the message correctly as they are forgetful and lazy. They often stop for long breaks during working hours and even try to escape from the city.

The Governors want to find a way of making their communication reliable, they want to develop a set of rules that they will all follow. By doing this they can tell whether or not their message has been delivered and if the message was correct.

The Governors have already decided that the destination should be written on the tablet.
In your groups your task is develop the rules that the Governors will use to communicate…"

Proudly developed by SMART with funding from Inspiring Australia

Materials needed:

- Messenger Action Cards, minimum 1 set per messenger, minimum 1 messenger per 10 students. Available for printing on the next pages.
- Tablet Cards, 2 to 4 sheets (16 to 32 cards) per student. Available for printing on the next pages.
- Pens / pencils.
- Print out and cut up the cards before the session.

Note: The action cards are three types; delay, don't deliver, deliver. Adjusting the ratio between these will represent the quality of your messengers. More "deliver" cards means a more reliable messenger. More "delay" and "don't delivers means a less reliable network. These cards are analogous to a computer network/communication channel.

Activity Notes:

- Students should work in pairs.
- The coordinator, and or up to 4 other students, act as messengers.
- Split the pairs into two groups, and have as much distance as possible between them. The messengers will deliver messages from group 1 to their pairs in groups 2. It is crucial for the pairs to sit apart from one another where they cannot see or communicate with each other. Two rooms are ideal but sitting students on opposing sides of a classroom should suffice.
- Decide on some messages for the students to send (before the session). It's important that they're not English sentences or anything that can be put back together by their structure. Something like "1LHC255HD+RLLS" would be a suitable message, or a mobile phone number.

How to play:

1. Shuffle the Messenger Action Cards and place an even amount in a pile for each messenger.
2. Give one student of each pair a message to deliver to their partner, using only their tablet cards. The writer can only enter 6 digits, or characters, of the message on their tablet card (i.e. sending only partial messages).
3. A student from each pair should write on their tablet card, and give it to the messenger. The tablet card should at least say the name of the other person in their pair on it.
4. The messenger will collect the tablet card from the writer, pick up the top messenger action card, turn it over, read it and use the action card to decide what to do with the tablet card they have received. They may deliver it to the receiver in the pair!
5. Repeat steps 3 and 4 with each tablet.

After 5 or so minutes of chaos and frustration, your students should realise that names alone are not good enough for a protocol. Stop the class and discuss this…

What is the first issue they're having? Is it order? Perhaps it would be best to use one of those 6 slots to put a tablet number in? This means there is less room for the actual message – what does this mean in terms of the number of tablets we have to use now?

After some more time, they might notice other problems, and these should also be discussed. Possible problems could be a missing tablet, not knowing if the tablet was delivered, not knowing whether to resend a tablet.

Solutions you could suggest would be a sending back acknowledgements and waiting to hear back for these before re-sending another tablet – this means that the receiving student(s) also need blank tablets to send messages, and they will have to agree on what their 6-character responses mean before they play the game again.

You'll need at least two students for this game, but we recommend having as many as possible. If you have a large class, consider a few messengers. Once again, discuss this with your class… what happens if you have many messengers? What happens if you had one?

Activity 6.3.3: Messenger Action Cards

Deliver this tablet now	Deliver this message after the next one
Deliver this tablet now	Deliver this message after the next one
Deliver this tablet now	Deliver this message after the next one
Deliver this tablet now	Don't deliver this message
Deliver this tablet now	Don't deliver this message

Activity 6.3.3: Tablet Cards

To: From:	To: From:
To: From:	To: From:
To: From:	To: From:
To: From:	To: From:

Proudly developed by SMART with funding from Inspiring Australia

SMART

An Australian Government Initiative

Inspiring AUSTRALIA

Module 6.3 Computers and Coding – Algorithms and Apps

Lesson Plan

90 minute session

High Tech: Use PowerPoint Presentation 'M6.3 - Master Slides'.
Note: If computers and internet available, use Challenge 6.3a and slides 10-15.
If UNPLUGGED activities to be used instead, omit slides 10-15 and use slides 16 – 23.

Key Learning Area	**Topic**
Maths, Computer Science, Digital Technology	Algorithms

Timing	Running Time (hh:mm)	Procedure	Materials
2 min	00:02	**Lesson Introduction** Welcome! Recap how we program. Introduce the word 'Algorithm'.	M6.3 PowerPoint (Slides 1-2)
3 min	00:05	**Body of Lesson** Discuss usefulness of algorithms for managing data. Introduce SORTING and SEARCHING algorithms. Show video. If video unable to be played, ensure you watch prior to session and explain steps to students.	M6.3 PowerPoint (Slide 3)
10 min	00:15	Introduce the Bubble Sort Algorithm. Discuss the concept. Students may choose to write a number array and practice the concept. Undertake Activity 6.3.1.	M6.3 PowerPoint (Slide 4) Optional: pencils, paper
2 min	00:17	Algorithms are everywhere! Brainstorm where algorithms might be useful tools.	M6.3 PowerPoint (Slide 5)
(5min)	(00:22)	*Optional Video, (allow an extra 5 minutes). 'What's an Algorithm By David J Malan'. Suitable for experienced coders and/or senior primary / high school students.*	

10 min	00:27 (00:32)	Algorithms and Magic. Introduce the 21 card trick. Demonstrate Activity 6.3.2. Support students to try out the card trick if time permits.	M6.3 PowerPoint (Slides 6-7) Playing cards, minimum 1 partial deck of 21 cards.
2 min	00:29 (00:34)	Discuss Networks, and the need for algorithms to assist in network design.	M6.3 PowerPoint (Slide 8)
3min	00:32 (00:37)	Discuss Apps. What is an App? Types of Apps? Discuss what Algorithms may be useful for App design.	M6.3 PowerPoint (Slide 9)
20 min	00:52 (00:57)	Introduce the MIT App Inventor Challenge* 6.3a Announce Challenge. Show the Video Tutorial. If video unable to be played, ensure you watch prior to session and explain steps to students. Step through the Designer and Block Editors. Support students to access computers/internet and navigate to the MIT App Inventor website. Students will need an email address to log-in to MIT App Inventor. Support students to work through steps to complete the Ball Bounce activity.	M6.3 PowerPoint (Slides 10-14) Computers, Internet access, Email Accounts, Ball Bounce Tutorial Sheets Android devices (phones, tablets) are optional to support this activity. ** Note: If computers are not available, introduce and start the Island Connection Challenge 6.3b, discuss Minimal Spanning Trees.*
3 min	00:55 (01:00)	Recap the Ball Bounce Activity** Who was successful? What challenged were experienced? How were they overcome? Would anyone like to share their App / coding with the group?	PowerPoint M6.3 (Slide 15) ***Note: Alternately, ask students how they are progressing with the Island Connection Challenge.*

25 min	01:25	Mini Golf App Design*** Support students who have completed the Ball Bounce activity to work through steps to complete the Mini Golf activity. Encourage students to try out each other's Mini Golf games. ****Note: Alternately, undertake unplugged activity 6.3.3 "Tablets of Stone"*	PowerPoint M6.3 (Slide 10) Computers, Internet access, Email Accounts, Mini Golf Tutorial Sheets Android devices (phones, tablets) are optional to support this activity.
5 min	01:30	**Lesson Conclusion** Pack up and discussion, relevant to activities undertaken. Example: Did everyone's Mini-Golf game work in the same way? Did everyone's coding look the same, or were there differences?	Ensure all students have saved their work and logged out of their MIT App Inventor accounts, emails and computers.

Proudly developed by SMART with funding from Inspiring Australia SMART An Australian Government Initiative Inspiring AUSTRALIA

Module 6.3 Computers and Coding – Algorithms and Apps

Lesson Plan

75 minute session

High Tech: Use PowerPoint Presentation ‘M6.3 - Master Slides’. Hide slides: 6, 7, 8 (note, include slide 8 if choosing to run with the unplugged Challenge 6.3b)
Note: If computers and internet available, use Challenge 6.3a and slides 10-15.
If UNPLUGGED activities to be used instead, omit slides 10-15 and use slides 16 – 23.

Key Learning Area	**Topic**
Maths, Computer Science, Digital Technology	Algorithms

Timing	Running Time (hh:mm)	Procedure	Materials
		Lesson Introduction	
2 min	00:02	Welcome! Recap how we program. Introduce the word ‘Algorithm’.	M6.3 PowerPoint (Slides 1-2)
		Body of Lesson	
3 min	00:05	Discuss usefulness of algorithms for managing data. Introduce SORTING and SEARCHING algorithms. Show video. If video unable to be played, ensure you watch prior to session and explain steps to students.	M6.3 PowerPoint (Slide 3)
10 min	00:15	Introduce the Bubble Sort Algorithm. Discuss the concept. Students may choose to write a number array and practice the concept. Undertake Activity 6.3.1.	M6.3 PowerPoint (Slide 4) Optional: pencils, paper
2 min	00:17	Algorithms are everywhere! Brainstorm where algorithms might be useful tools.	M6.3 PowerPoint (Slide 5)
3 min	00:20	Discuss Apps. What is an App? Types of Apps? Discuss what Algorithms may be useful for App design.	M6.3 PowerPoint (Slide 9)

20 min	00:40	Introduce the MIT App Inventor Challenge* 6.3a Announce Challenge. Show the Video Tutorial. If video unable to be played, ensure you watch prior to session and explain steps to students. Step through the Designer and Block Editors. Support students to access computers/internet and navigate to the MIT App Inventor website. Students will need an email address to log-in to MIT App Inventor. Support students to work through steps to complete the Ball Bounce activity.	M6.3 PowerPoint (Slide 10-14) Computers, Internet access, Email Accounts, Ball Bounce Tutorial Sheets Android devices (phones, tablets) are optional to support this activity. ** Note: If computers are not available, introduce and start the Island Connection Challenge 6.3b, discuss Minimal Spanning Trees.*
3 min **25 min**	00:43 01:08	Recap the Ball Bounce Activity** Who was successful? What challenged were experienced? How were they overcome? Would anyone like to share their App / coding with the group? Mini Golf App Design*** Support students who have completed the Ball Bounce activity to work through steps to complete the Mini Golf activity. Encourage students to try out each other's Mini Golf games. ****Note: Alternately, undertake unplugged activity 6.3.3 "Tablets of Stone"*	PowerPoint M6.3 (Slide 15) ***Note: Alternately, ask students how they are progressing with the Island Connection Challenge.* PowerPoint M6.3 (Slide 10) Computers, Internet access, Email Accounts, Mini Golf Tutorial Sheets Android devices (phones, tablets) are optional to support this activity.

7 min	01:15	**Lesson Conclusion** Pack up and discussion, relevant to activities undertaken. Example: Did everyone's Mini-Golf game work in the same way? Did everyone's coding look the same, or were there differences?	Ensure all students have saved their work and logged out of their MIT App Inventor accounts, emails and computers.

Module 6.3 Computers and Coding – Algorithms and Apps

Lesson Plan

45 minute session

High Tech: Use PowerPoint Presentation 'M6.3 - Master Slides'. Hide slides: 5, 6, 7, 8
Note: If computers and internet available, use Challenge 6.3a and slides 10-15.
If UNPLUGGED activities to be used instead, omit slides 10-15 and use slides 16 – 23.

Key Learning Area	**Topic**
Maths, Computer Science, Digital Technology	Algorithms

Timing	Running Time (hh:mm)	Procedure	Materials
		Lesson Introduction	
2 min	00:02	Welcome! Recap how we program. Introduce the word 'Algorithm'.	M6.3 PowerPoint (Slides 1-2)
		Body of Lesson	
3 min	00:05	Discuss usefulness of algorithms for managing data. Introduce SORTING and SEARCHING algorithms. Show video. If video unable to be played, ensure you watch prior to session and explain steps to students.	M6.3 PowerPoint (Slide 3)
8 min	00:13	Introduce the Bubble Sort Algorithm. Discuss the concept. Undertake Activity 6.3.1.	M6.3 PowerPoint (Slide 4) Optional: pencils, paper
2 min	00:14	Discuss Apps. What is an App? Types of Apps? Discuss what Algorithms may be useful for App design.	M6.3 PowerPoint (Slide 9)
25 min	00:42	Introduce the MIT App Inventor Challenge* 6.3a Show the Video Tutorial. If video unable to be played, ensure you watch prior to session and explain steps to students.	M6.3 PowerPoint (Slide 10-14) Computers, Internet access, Email Accounts, Ball Bounce and Mini Golf Tutorial Sheets. Android devices (phones, tablets) are optional to support this activity.

		Step through the Designer and Block Editors. Support students to access computers/internet and navigate to the MIT App Inventor website. Students will need an email address to log-in to MIT App Inventor. Support students to work through steps to complete the Ball Bounce activity and Mini Golf activity if time permits.	** Note: If computers are not available, introduce and start the Island Connection Challenge 6.3b, discuss Minimal Spanning Trees.*
3 min	00:45	**Lesson Conclusion** Pack up and discussion, relevant to activities undertaken. Example: Did everyone's Mini-Golf game work in the same way? Did everyone's coding look the same, or were there differences?	Ensure all students have saved their work and logged out of their MIT App Inventor accounts, emails and computers.

Module 6.3 - References

CSUnplugged.org

CS Education Research Group at the University of Canterbury, NZ, Creative Commons BY-NC-SA 4.0 licence.

Code.org

© Code.org, Used as per copyright agreement.

MIT App Inventor

This session uses MIT's App Inventor as well as links to some of its resources.

http://appinventor.mit.edu/explore/

MIT's App Inventor and resources are licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License

© 2012-2017 Massachusetts Institute of Technology

21 Card Trick http://kidsentertainerhub.com/the-21-card-trick/

Algorithms

https://www.theguardian.com/science/2013/jul/01/how-algorithms-rule-world-nsa

http://www.geeksforgeeks.org/greedy-algorithms-set-5-prims-minimum-spanning-tree-mst-2/

https://en.wikipedia.org/wiki/Bubble_sort

Module 6.3- Required Materials

- Pens, pencils and writing paper are generally required every session.
- Students may like to bring a note pad to record their observations and ideas.
- A group usually refers to 2 - 4 students.

Activity	Material	Amount	Where can I find it?
All sessions	PowerPoint Slides* (digital, or printed)	1 per coordinator	Coordinator Package
All sessions	Printed PowerPoint* Slide Notes	1 per coordinator	Coordinator Package
All sessions	Printed Lesson Plan	1 per coordinator	Coordinator Package
All sessions	Printed Module 1 Risk Assessment	1	Coordinator Package
All sessions	Computer, Data Projector, Screen	1	Venue
Activity 6.3.1 Bubble Sort All Sessions	Paper, pens (Optional)	1 per student	Stationary shop, recycled.
Activity 6.3.2 21 Card Trick 120 minute, 2 x 60 minute, 90, 75 minute sessions	Deck of playing cards	Min 1 x partial deck of 21 cards	Variety store, newsagency, games shop
Activity 6.3.3 Tablets of Stone 120 minute, 2 x 60 minute, 90, 75 minute sessions (UNPLUGGED SESSIONS)	Printed (and pre-cut out): 'Tablet Cards' and 'Message Action Cards'	2 to 4 sheets of 'tablet cards' per student. 1 sheet 'message action cards' per messenger (min 1).	Coordinator Package
	Pens / pencils	1 per student	Stationary shop, recycled.
	Timer	1 per coordinator	Venue / room clock. Phone application. Variety or sports store.

* PowerPoint Slides have been provided as a Master Slide Set for a 120 minute (or 2 x 60 minute) session duration. Hide/ omit slides as noted in lesson plans for delivery of shorter session durations.

Activity	Material	Amount	Where can I find it?
Challenge 6.3a App Inventor All Sessions Ball Bounce and Mini Golf Activities	Computer, tablets, laptops, with internet connection	1 per student group	Venue, BYO Device
	Email Address (school or gmail)	1 per student group	BYO or create online https://accounts.google.com/SignUp
	Android Devices or Android emulator	1 per student group	BYO Device or download and install on computer
Challenge 6.3b Island Connection (ALL UNPLUGGED SESSIONS)	Island Maps and Score Sheets	1 per group (encourage pairs)	Coordinator Package
	Island Cards	1 or 2 sets, depending on group size	Coordinator Package
	Pens/Pencils	1-2 per student	Stationary shop, recycled.

Proudly developed by SMART with funding from Inspiring Australia
SMART An Australian Government Initiative Inspiring AUSTRALIA