CoolThink@JC –
Computational Thinking Education Programme

Course Code & Title

CF3E Computational Thinking Development - Level 3

Unit Code & Topic

L3U8.1-8.2

Hong Kong Tour Guide

香港教育大學
The Education University of Hong Kong
Massachusetts Institute of Technology
Funded by
香港賽馬會慈善信託基金
The Hong Kong Jockey Club Charities Trust
同心同步同進 RIDING HIGH TOGETHER

These teaching and learning materials were co-created by The Education University of Hong Kong, and Massachusetts Institute of Technology, and funded by The Hong Kong Jockey Club Charities Trust. (Revised August 2019)

Table of Contents

1. Synopsis

In this unit, students will learn to use the Map component in MIT App Inventor to create a tour guide app that showcases sites of interest in their community. Students will choose several places of interest from a predefined list of 15 sites in order to create an app that can be used by visitors to locate and learn about the sites. Students will also use the TinyDB component, learning how to store data persistently in an app. The focus of Lessons 1 and 2 is naming, as students will have to appropriately name components in order to keep track of them. The focus of Lessons 3 and 4 is data manipulation through lists and TinyDB. Students will continue to develop their knowledge of computational thinking concepts, such as sequences, events, naming and lists in creating this app. The computational thinking practices of abstracting and modularizing, being incremental and iterative, and testing and debugging will also be necessary for designing and making the app.

2.Unit Intended Learning Outcomes (UILOs)

After completing this unit, students will be able to:

UILO$_1$: design and code a location-based app, using the Maps component in MIT App Inventor;

UILO$_2$: create an app that uses multiple screens and passes values;

UILO$_3$: use the TinyDB component to store data persistently on a device;

UILO$_4$: demonstrate their understanding of parallel list data structures, utilizing multiple lists to manage data;

UILO$_5$: apply the computational thinking practices of abstracting and modularizing, being incremental and iterative, testing and debugging, and using algorithmic thinking;

UILO$_6$: work collaboratively with a partner to create a mobile app; and

UILO$_7$: increase their positive perception of coding by creating an app to show others around their city.

3. Mapping with the Computational Thinking Framework

The following tables show the alignment of this unit with the intended learning outcomes of the computational thinking framework. The entries indicate the expected relevance of this unit to each outcome:

✓✓✓ : High relevance
✓✓ : Some relevance
✓ : Low relevance

Computational Thinking Concepts

L3U8.1-8.2: Hong Kong Tour Guide		
1. Sequences	✓	Screen initialization uses a sequence to update location information displayed.
2. Events	✓✓✓	New events: Button.LongClick, Screen.Initialize, Camera.AfterPicture
3. Repetition		
4. Conditionals		
5. Parallelism		
6. Naming	Lessons 1 and 2 Focus	It is necessary to name variables and lists to access information on locations.
7. Operators		
8. Manipulation of data and elementary data structures	Lessons 3 and 4 Focus	Students use multiple parallel lists to store information about locations. They will also learn how to store data locally on a device via TinyDB.

Computational Thinking Practices

L3U8.1-8.2: Hong Kong Tour Guide		
1. Reusing and remixing		
2. Being incremental and iterative	✓✓✓	Students will add features to their apps incrementally in four lessons.
3. Abstracting and modularizing	✓✓	Location names will trigger indexing into lists to pull out information.
4. Testing and debugging	✓✓✓	Students will test each new feature added to their apps incrementally.
5. Algorithmic thinking	✓✓	Indexing into multiple parallel lists is used to access related information about sites.

Computational Thinking Perspectives

L3U8.1-8.2: Hong Kong Tour Guide		
1. Expressing	✓✓✓	Students will decide which sites to showcase in their apps.
2. Connecting	✓✓✓	The app connects students to their community by identifying and learning about different locations in Hong Kong.
3. Questioning	✓✓✓	Students learn new technology to create a location-based map app that uses the camera.
4. Computational identity	✓✓	Students create an app they can share with friends about their community.
5. Digital empowerment	✓✓	Students expand their knowledge of components they can use in apps.

4. Learning Prerequisites

Students should have completed Level 1 and 2 of the CoolThink@JC Computational Thinking Education Programme curriculum.

5. Pedagogical Approach

The suggested pedagogical approaches of this unit are as follows:

- Class discussion;
- Demonstration;
- Guided instruction;
- Pair Programming; and
- Testing and debugging.

6. Lesson Plan

This unit consists of 4 lessons of 35 minutes, with post-lesson activities after each period.

Lesson 1

Time	Activity
10 mins	**Introduction** 1. Ask students the following questions: (1) Have you used any maps-based apps? (2) What did you use them for? (3) Have you used an app to help you find out about different places to see while you were travelling? 2. Explain to students that they will work using the Pair Programming model to create an app that will highlight four different places in Hong Kong. They will make an app that shows the location of the places with markers on a map. Long-clicking on a location will display information about that location. A third screen allows users to take a picture, store and display it. 3. Demonstrate an example of the finished app.
5 mins	**Geographic Coordinate System Explanation** 1. Explain longitude and latitude. 2. Demonstrate with OpenStreetMap (http://www.openstreetmap.org) which is what the Map component uses.

5 mins	**Tour Guide Planning** Ask students to work with their partner to choose four landmarks or places that they would like to show people in the app from the *Hong Kong Landmarks Guide*. They will then note the corresponding information in their *Hong Kong Landmarks Worksheet*.
10 mins	**Maps and Markers Demonstration** Demonstrate the Maps and Markers components. Students will work using the Pair Programming model and follow along with teacher at the board to add the components needed for the app, using the *Student Guide: Lesson 1.* 1. Show students how to add a Map component. 2. Demonstrate how to add Marker components. 3. Show properties of Map and Marker components.
5 mins	**Wrap-up** Check with student groups to make sure they have all the necessary information for their chosen locations.
Post-Lesson Activities	(Required) Complete adding all four markers and all properties for each marker.

Lesson 2

Time	Activity
10 mins	**Review and Introduction of Multiple Screens and Parallel Lists** 1. Ensure students have completed all tasks from Lesson 1. 2. Explain the use of multiple screens in the app and demonstrate how to open a new screen with a start value. 3. Explain and demonstrate using parallel lists for related information. (1) This is used in LocationScreen in Lesson 3.
20 mins	**Coding of Screen1 of App** Ask students to work using the Pair Programming model and follow the *Student Guide: Lesson 2*. 1. Complete LongClick events for each marker. 2. Test and debug that the short-click on a marker displays a popup with the title and description of the marker that was set in the Designer. 3. Initialize three lists in LocationScreen.
5 mins	**Wrap-up and Review** 1. Check in with student groups to ensure they have completed the tasks for the *Student Guide: Lesson 2*. 2. Review the computational thinking concepts learnt in this lesson, according to the table in the Student Guide.
Post-Lesson Activities	(Required) Complete tasks for the *Student Guide: Lesson 2*.

Lesson 3

Time	Activity
5 mins	**Introduction & Review** 1. Check in with students to make sure they have completed the coding tasks from Lesson 2. 2. Demonstrate LocationScreen that they will be building in Lesson 3. 3. Review the use of parallel lists to access related information.
25 mins	**Implementation of LocationScreen in App** 1. Students work using the Pair Programming model and follow the Student Guide to code LocationScreen that displays a picture and description of a site when the marker is long-clicked. 2. Test and debug LocationScreen.
5 mins	**Wrap-up and Review** 1. Check in with student groups to ensure they have completed all tasks in the *Student Guide: Lesson 3*. 2. Review the computational thinking concepts learnt in this lesson, according to the table in the Student Guide.
Post-Lesson Activities	(Required) Complete tasks from the *Student Guide: Lesson 3*.

Lesson 4

Time	Activity
10 mins	**Introduction** 1. Review the app created so far. 2. Demonstrate the Gallery page of the finished app. 3. Demonstrate how to use the Camera component in an app. 4. Explain the use of TinyDB to store data persistently in an app. 5. Explain that they will be using these two new components, Camera and TinyDB, in their apps.
20 mins	**Implementation of Gallery Screen** Ask students to work using the Pair Programming model and follow the *Student Guide: Lesson 4* to implement GalleryScreen.
5 mins	**Wrap-up and Review** Review the computational thinking concepts learnt in this unit, following the tables in the four Student Guides. Also review the computational thinking practices used in this unit.
Post-Lesson Activities	1. (Required) Complete the multiple choice questions. 2. (Required) Complete the survey of learning attitudes. 3. (Required) Complete the open-ended questions on collaboration skills. 4. (Required) Complete all app features. 5. (Optional) Add more features to the app.

7. Instructional Details

The teaching and learning materials of the four lessons are presented in the appendices below. Please note the inclusion of the template and checkpoint aia files. These MIT App Inventor project files are included as backup resources for teachers. Each checkpoint aia file includes the project and code blocks that are expected to be completed during that lesson. If necessary, teachers can use these as reference or even a starter project file for students who miss a lesson due to illness.

Lesson 1

1. Teacher's Guide: Lesson 1
2. Hong Kong Landmarks Worksheet
3. Hong Kong Landmarks Guide
4. Student Guide Lesson 1
5. HongKongTourGuide_complete.aia
6. HongKongTourGuide_template.aia
   (1) This template has all three screens completed in the Designer, and some starter code is already implemented.
   (2) For LocationScreen, the strings already exist so that students do not have to spend time typing up the descriptions for locations. The photos are also uploaded to the template.

Lesson 2

1. Teacher's Guide: Lesson 2
2. Student Guide: Lesson 2
3. HongKongTourGuide_checkpoint1.aia

Lesson 3

1. Teacher's Guide: Lesson 3
2. Student Guide: Lesson 3
3. HongKongTourGuide_checkpoint2.aia

Lesson 4

1. Teacher's Guide: Lesson 4
2. Student Guide: Lesson 4
3. HongKongTourGuide_checkpoint3.aia

8. Assessment

The following table maps the assessment modes and corresponding Unit Intended Learning Outcomes (UILOs) in Section 2.

Assessment Modes	UILO
1. Teacher assessment	$UILO_{1,2,3}$
2. Self-assessment: (1) Multiple-choice questions (2) Survey of learning attitudes (3) Open-ended question on collaboration skills	$UILO_{2,3,4,6,7}$

Teacher Assessment

This rubric assesses the logical design of the student's project (e.g. Does the student code follow the correct logic which results in a working app?).

Level	5	4	3	2	1
Programming logic	Code blocks are appropriate, logical, efficient, and correct for the proposed solution.	One or two instances where code blocks do not work, are inappropriate, or could be more efficient.	Three to five instances where code blocks do not work, are inappropriate, or could be more efficient.	More than five instances where code blocks do not work, are inappropriate, or could be more efficient.	Code blocks are incorrect or incomplete and do not solve the problem.

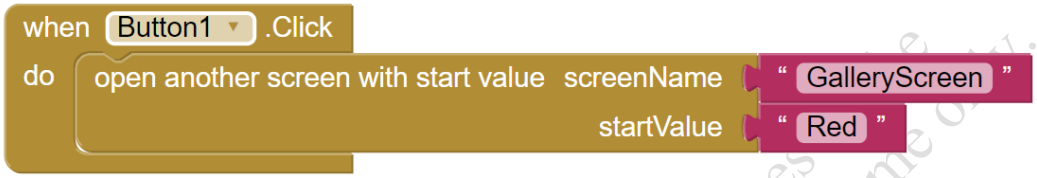
Remark: The above rubric for problem solving skills is adapted from the Rubrics for Generic Intended Learning Outcomes (GILOs) of The Education University of Hong Kong.

Self-assessment

Self-assessment allows students to reflect on their learning performance.

Multiple-choice Questions

1. A student makes an app with two screens, Screen1 and GalleryScreen. He has the code in Screen1 to pass a start value to GalleryScreen.

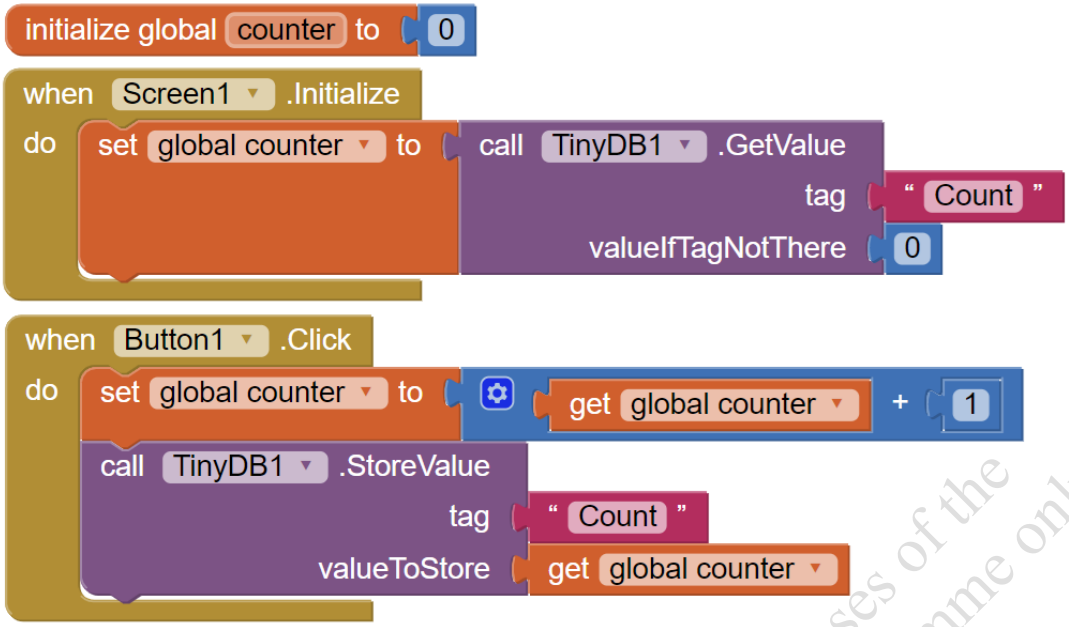


He also has an "if" block in GalleryScreen to change the canvas colour to red if the start value = "Red", but can't remember where to put the block. In which event block should it go?

A. Button1.Click
B. Canvas1.Click
C. GalleryScreen.Click
D. GalleryScreen.Initialize

(Answer: D)

For Questions 2-3, use the following code blocks:



2. When the app starts for the first time, what is the value of counter?

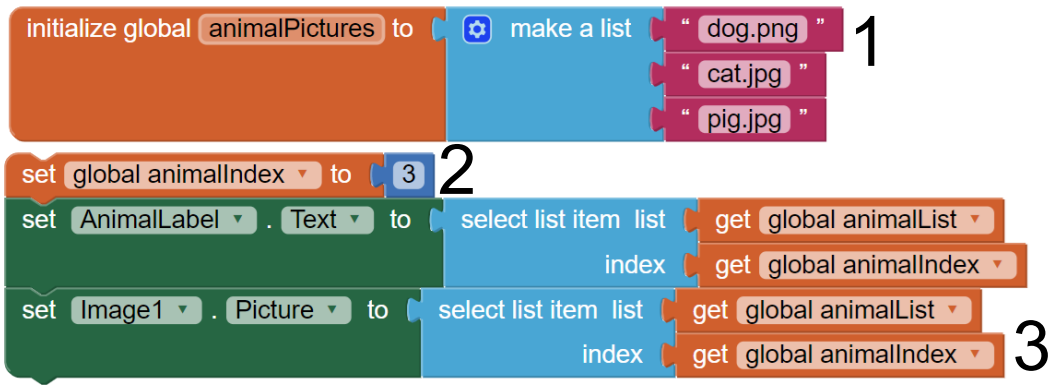
   A. 0
   B. 1
   C. 2
   D. 3
   (Answer: A)

3. The user runs the app once, and clicks Button1 three times, then closes the app. What is the value of counter when he opens the app again?

   A. 0
   B. 1
   C. 2
   D. 3

   (Answer: D)

4. Anna uses the following blocks in her app, but no picture appears. What is wrong?



A. "pig.jpg" needs to be changed to "pig.png" in the block labelled 1.

B. "3" of animalIndex must be changed to "1" in the block labelled 2.

C. "animalList" must be changed to "animalPictures" in the block labelled 3.

D. The Images all appear.

(Answer: C)

Survey of Learning Attitudes

To evaluate students' attitudes, perceptions and understandings of coding, they are required to complete the following questionnaire by putting a "✓" in the appropriate box.

After completion of this unit, I think…	Disagree	Somewhat disagree	Neutral	Somewhat agree	Agree
I feel inspired when learning more about programming.					
I feel more connected to the technology around me when I make apps.					
Programming helps me relate to people in my community.					
The learning activities of this unit are interesting and I like them.					
I am excited to share this app with friends and family.					

Collaboration Skills Questions

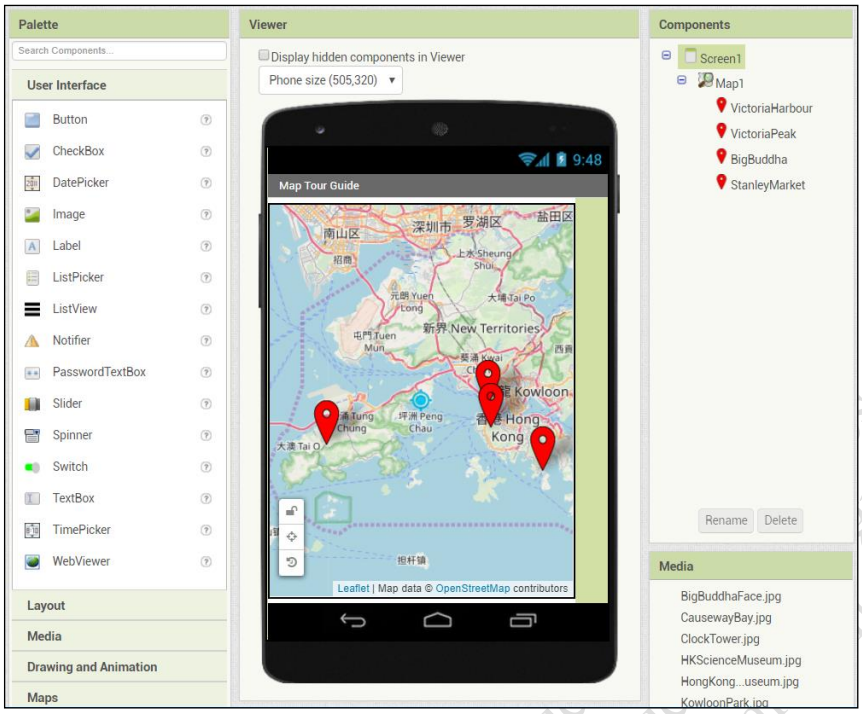
Please ask students to answer the following questions in the CoolThink@JC Learning Platform.

- Did you like working with another person? Do you feel you were a good partner, and respected and encouraged your partner in this project?
- What was your role as a partner in this project? What did you do and what did your partner do?

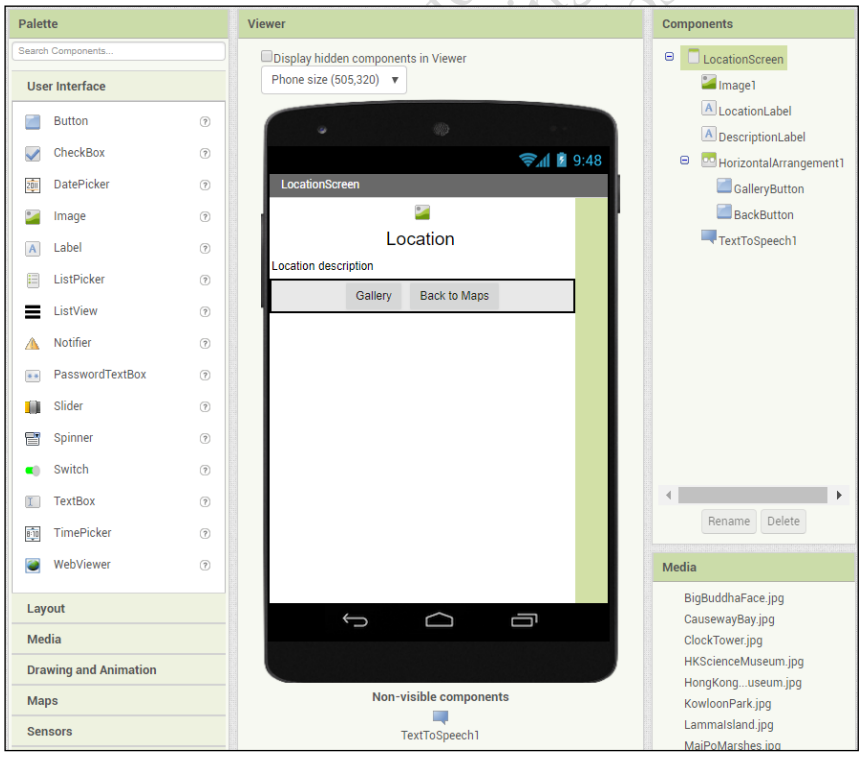
9. Screen Design and Code

Designer

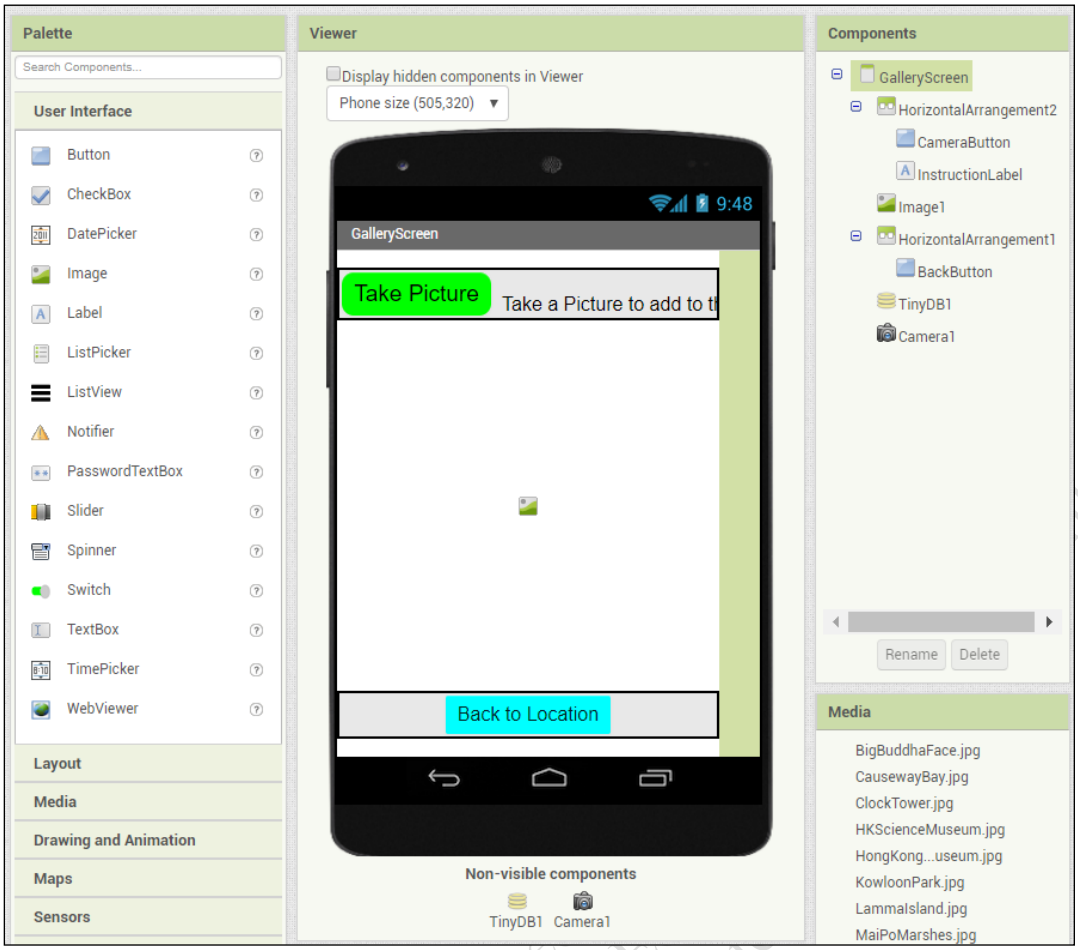
Screen1



LocationScreen

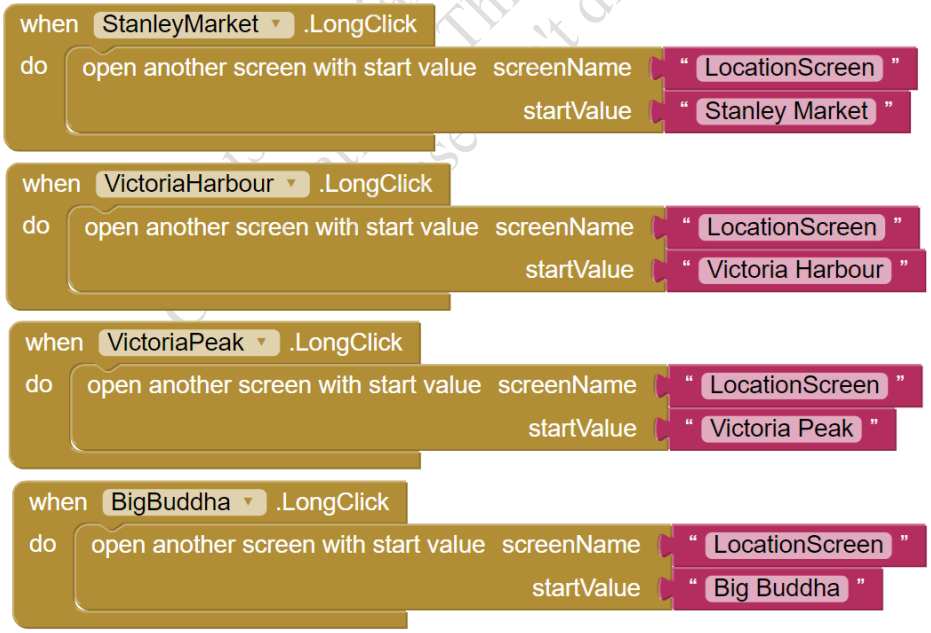


GalleryScreen

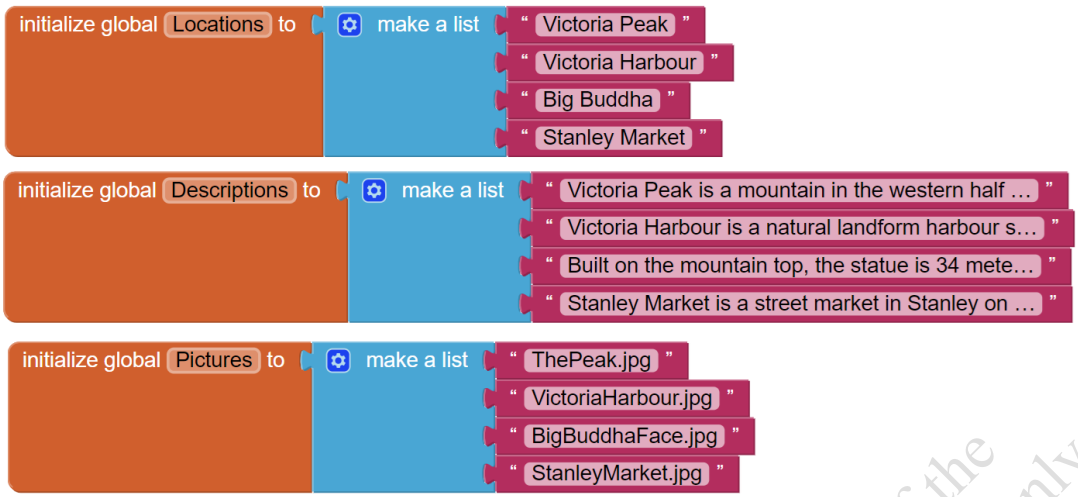


Blocks (Checkpoint 2)

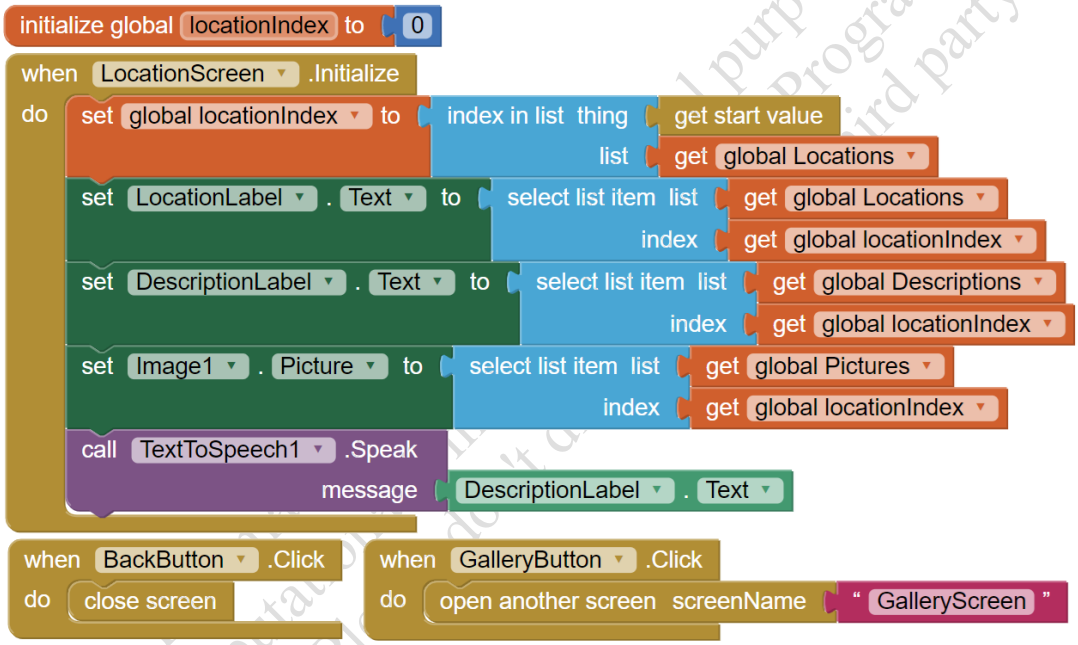
Screen1



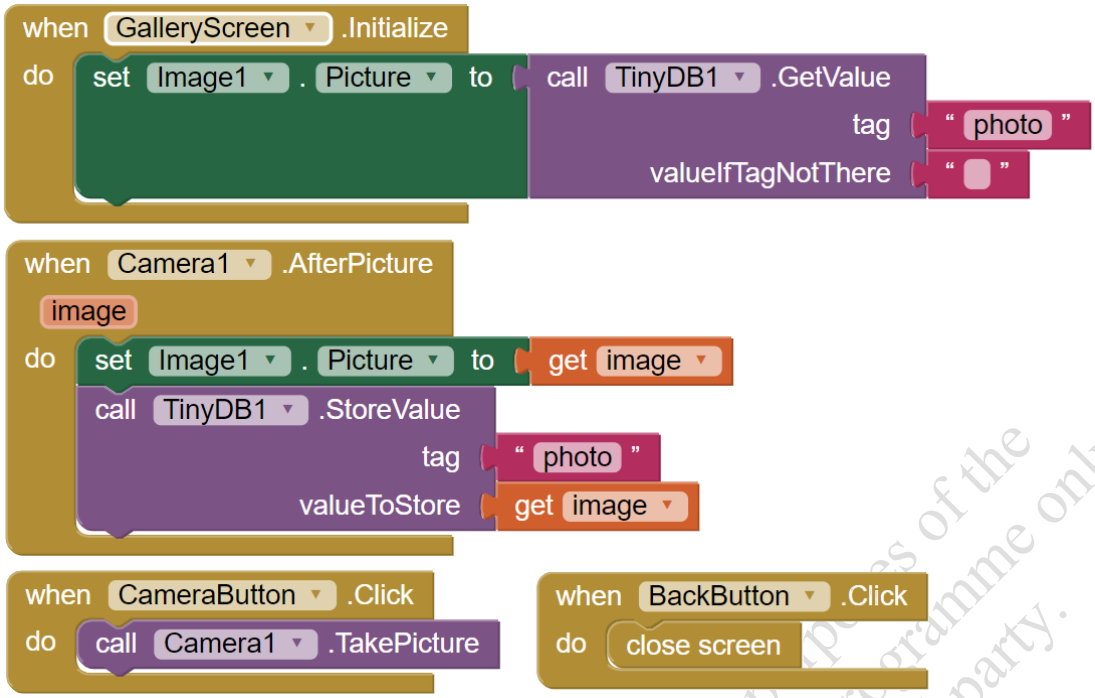
LocationScreen



Blocks (Checkpoint 3) - added to blocks above



GalleryScreen: (Completed app) added to blocks above



Appendix 1

L3U8.1-8.2

Teacher's Guide: Lesson 1

Teaching Objectives

1. Focus: Reinforce the concept of naming to make students' code easier to work with and read.
2. Teach students about the geographic coordinate system, and how to use latitude and longitude to mark a geographic location.
3. Assist students in taking steps to plan a tour guide app, gathering visual and text content in preparation.

Lesson Outline

Introduction (10 minutes)

Start by a short class discussion to relate this unit to other apps students may have used.

1. Ask students the following questions:
   (1) Have you used any maps-based apps?
   (2) What did you use them for?
   (3) Have you used an app to help you find out about different places to see while you were travelling?
2. Explain to students that they will work in pairs to create an app that will highlight four different places in Hong Kong. The students will make an app that shows the location of the places with markers on a map. Clicking on a location will display information about that location in a pop-up. Long-clicking will open another screen with more detailed information and a picture of the landmark. A third screen allows the user to take a picture for a "gallery".

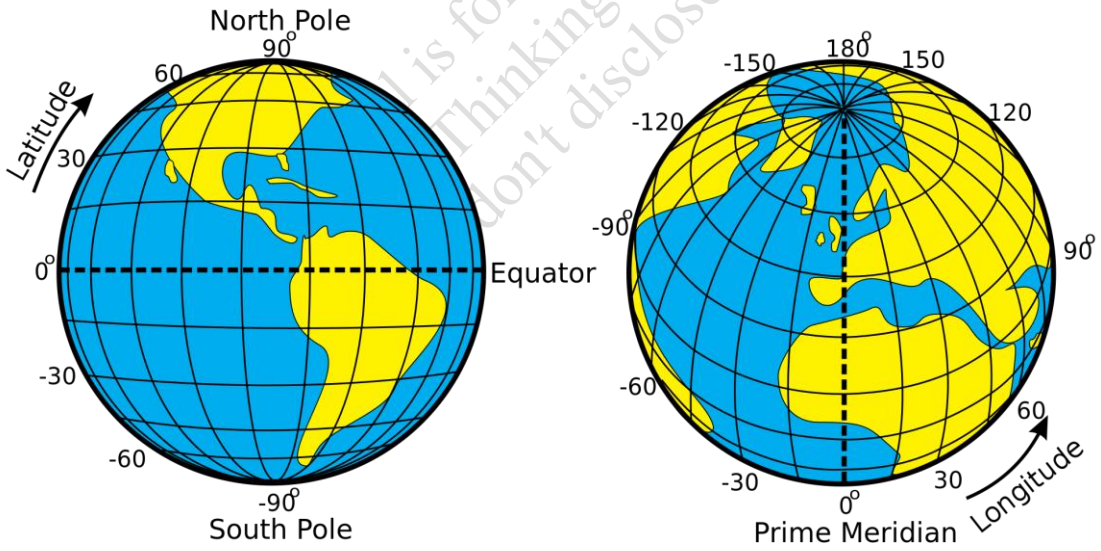
3. Remind students that they will work using the Pair Programming model in this unit. Remind them how the Pair Programming model works, and ask them if they can remember the table of rules.

DO	DON'T
• Be respectful. • Talk to one another about the work. • Explain what you are doing. • Think ahead and make suggestions. • Switch roles often.	• Be a bossy navigator. • Grab the driver's mouse/keyboard.

4. Demonstrate the finished app, showing all three screens. (HongKongTourGuide_complete.aia)

Geographic Coordinate System Explanation (5 minutes)

1. Explain to students that they will be using the geographic coordinate system, where latitude is the vertical marker and longitude is the horizontal marker for a location on the globe. Latitude is measured with 0º at the equator, the North Pole 90º and the South Pole -90º. Longitude is measured with 0º at Prime Meridian, which is located in Greenwich, England. Clockwise around the globe is measured 0º to positive 180º, and counter-clockwise around the globe is measured 0º to negative 180º.



Each location on earth can then be marked by a location of latitude, longitude on the earth's surface.

2. Demonstrate using OpenStreetMaps.
    (1) Click on a location to place a marker.
    (2) Note the latitude and longitude that appears with the location at the bottom of the screen.
    (3) Click left/right to see the change in longitude.
    (4) Click up/down to see the change in latitude.

Tour Guide Planning (5 minutes)

Ask students to work with their partner to choose four landmarks or places they would like to show people in their travel app, based on the *Hong Kong Landmarks Guide*. They will complete the *Hong Kong Landmarks Worksheet* with the provided information.

1. Location name
2. Short description
3. Longer paragraph of information about the site
4. Picture
5. Latitude, longitude

All of this information will be in the Guide. Students just need to identify which four landmarks they will use in their apps, and fill in their worksheets appropriately.

Map and Markers Demonstration (10 minutes)

Introduce students to the Map and Markers components in MIT App Inventor by demonstrating the components at the board, and having students follow along at their computers.

1. Remind students that they will be working in pairs on this app, and to follow the guidelines for the Pair Programming model.
2. Ask students to open the HongKongTourGuide_template.aia project in MIT App Inventor.
    (1) Note that the template should have all location choices available; i.e., there should be names & descriptions as text blocks in the app already, as well as photos and strings for their names, that students can use.

(2) Latitude/Longitude values will need to be entered in the Designer by students. Those values can be found in the *Hong Kong Landmarks Guide*.

3. Drag a Map component onto the screen and change its width and height properties to "Fill Parent".
4. Drag out four Marker components onto the Map component.
5. Show students how to find the latitude, longitude for Hong Kong (22.396428, 114.109497) using http://www.openstreetmap.org.
6. Set the "CenterFromString" property of the Map component to Hong Kong's latitude, longitude.
7. Show students how changing the "ZoomLevel" property for the Map component will change how zoomed in or out the map is. Setting to 8 will zoom out enough to see all of Hong Kong, and students can adjust the number up to zoom in more.
8. Rename one marker to a location in Hong Kong. Remind students it is a good idea to name components to describe what they are being used for. For example, "VictoriaPeak".
9. Show students how to change the latitude and longitude properties for a marker. For example, Victoria Peak's latitude is 22.275883, and its longitude is 114.145532.
10. Show students how to change the Title and Description properties for a marker.
11. Explain that checking "EnableInfoBox" for a marker will pop up an information box for the marker when it is clicked.
12. Ask students to update the properties for their 4 markers: Title, Description, Latitude, Longitude, according to the information on the Landmarks Worksheet. They may follow these directions in the *Student Guide: Lesson 1*. They should also make sure EnableInfoBox is checked.

Wrap-up (5 minutes)

1. Check with student groups to make sure they have added all 4 markers and completed setting the properties for them.
2. Review the computational thinking concepts learnt in this lesson, according to the table in the *Student Guide: Lesson 1*.

Post-Lesson Activities

(Required) Complete adding all four markers and all properties for each marker.

Appendix 2

L3U8.1-8.2

Teacher's Guide: Lesson 2

Teaching Objectives

1. Focus: Continue reinforcing logical naming as an important practice for writing better code.
2. Teach students how to open another screen in an app and to pass a start value to that screen.
3. Demonstrate the use of parallel lists to organize related information in an easily accessible way.
4. Encourage students to work collaboratively using the Pair Programming model.

Lesson Outline

Review and Introduction of Multiple Screens and Parallel Lists (10 minutes)

1. Ensure students completed the tasks from Lesson 1: Screen1 of the app has a map with four markers, and those four markers all have their Title, Description, and Latitude and Longitude properties correctly set.
2. Explain how to add and access a second screen.
   (1) Explain that if you need to pass a small piece of information between screens, you can use the "open another screen with start value" block. Then the "start value" is a variable that can be accessed in the second screen.
   (2) Explain that this method will be used. The app will pass the location name of the clicked marker between screens as a start value.

3. Explain and demonstrate using parallel lists for related information (using the table below). Students will use three lists in LocationScreen - Locations, Descriptions and Pictures. Each list will contain the information for the four markers. The important thing is that they are all in the same order, e.g. Victoria Peak information are all the first element in each list. So when the start value is passed to LocationScreen, it will tell the app which information to pull out of each list, based on the same index.

Locations		Descriptions	Pictures
1	Victoria Peak	You can get to the Peak by taking a 120-year-old funicular tram.	VictoriaPeak.jpg
2	Victoria Harbour	Victoria Harbour is very popular with tourists, and is the main location for fireworks in Hong Kong.	VictoriaHarbour.jpg
3	Stanley Market	Stanley Market is an open-air market full of shops that is very popular with tourists.	StanleyMarket.jpg
4	Big Buddha	Also known as the Tian Tan Buddha, you have to walk up 268 steps to reach the Buddha!	BigBuddaFace.jpg

Coding of Screen1 of App (20 minutes)

Ask students to work using the Pair Programming model and follow the *Student Guide: Lesson 2* to do the following

1. Complete LongClick events for each marker. When each marker is long-clicked, LocationScreen will open with the given location name as the start value.
2. Test that the four markers appear when the app is run, and that a short-click will cause a pop-up with the Title and Description. Since LocationScreen is not yet coded, students will not test that screen in this lesson.

3. Initialize three lists in LocationScreen - Locations, Descriptions and Pictures. All text for filling out these lists will be included in the template. Students just have to locate the correct text blocks and snap them into each list in the correct position.
4. Test and debug that Screen1 works as expected.

   (1) Clicking on each marker should make the corresponding title and description appear in the info box.
   (2) Long-clicking each marker should open LocationScreen, but nothing should happen yet, as that screen has not yet been coded.

Wrap-up and Review (5 minutes)

1. Check in with student groups to make sure they have completed the *Student Guide: Lesson 2* and have tested to see that the correct information pops up for each marker when it is short-clicked. Also, long-clicking on a marker should open LocationScreen.
2. Review the computational thinking concepts learnt in this lesson, according to the table in the *Student Guide: Lesson 2*.
3. Explain that students will code LocationScreen in the next lesson, so users will see more information about each location when the marker is long-clicked.

Post-Lesson Activities

(Required) Complete the *Student Guide: Lesson 2* tasks.

Appendix 3

L3U8.1-8.2

Teacher's Guide: Lesson 3

Teaching Objectives

1. Focus: Guide students to expand their knowledge of lists.
2. Demonstrate the computational thinking practice of abstracting through the use of multiple parallel lists, where a single index is used to access related information.
3. Reinforce ways to store and access information by indexing into a list.
4. Encourage being incremental and iterative by adding new features to an app step-by-step.
5. Demonstrate how to use the TextToSpeech component in an app.
6. Guide students to test and debug their apps thoroughly.

Lesson Outline

Introduction and Review (5 minutes)

1. Check in with the student groups to make sure all groups have added the map and markers to their apps, and that long clicking on any of the markers brings up the LocationScreen.
2. Demonstrate the LocationScreen of the Hong Kong Tour Guide app. Show students how long-clicking on a marker in Screen1 will open LocationScreen, and display the Title, Description, and a picture of the landmark. It will also "speak" the description, using the TextToSpeech component.
3. Review the use of parallel lists to access related information. Explain that students will write code to set an index, based on the start value passed from Screen1. That index will be used to access the correct location name, description and picture for the selected marker.

Implementation of LocationScreen in App (25 minutes)

Ask students to work using the Pair Programming model and follow the *Student Guide: Lesson 3* to modify a second screen that displays a picture and description of a site when the marker is long-clicked. Circulate among the groups to make sure students are on track. Encourage groups to fully test their apps by long-clicking on each marker and checking that the correct information and picture appear.

Wrap-up and Review (5 minutes)

1. Check in with student groups to make sure all teams have completed the Student Guide and have a working app with Screen1 and LocationScreen working as expected.
2. Review the computational thinking concepts learnt in this lesson, according to the table in the *Student Guide: Lesson 3*.

Post-Lesson Activities

(Required) Complete tasks from the *Student Guide: Lesson 3*.

Appendix 4

L3U8.1-8.2

Teacher's Guide: Lesson 4

Teaching Objectives

1. Focus: Teach students more about data manipulation in their apps by adding persistent data storage using TinyDB.
2. Teach students how to use the Camera component in an app.
3. Encourage students to test and debug their apps to make sure they continue to work as they add more features.
4. Encourage students to work collaboratively using the Pair Programming model.

Lesson Outline

Introduction (10 minutes)

1. Review the app created so far.
2. Demonstrate the Gallery screen of the finished app. Show students how the new screen will allow users to take a picture and display it as a gallery.
3. Demonstrate how to use the Camera component. In the Blocks Editor, show TakePicture and AfterPicture blocks.
   (1) Explain the TakePicture block calls up the regular camera feature of a smartphone or tablet.
   (2) Explain the AfterPicture block is an event that is triggered in the app when the camera finishes taking a picture.
4. Explain the ability to store data persistently using TinyDB. In the Blocks Editor, show the StoreValue and GetValue blocks.
   (1) Explain that TinyDB stores values even when the app is closed.
   (2) Explain the concept of storing via tags, and value if not exists. Explain that a tag for TinyDB is analogous to a variable name, and that the value in TinyDB is the same as the value of a variable.

(3) Explain the parameter "value if not exists" allows you to set a default value if the user has not yet saved anything in TinyDB.

(4) Tell students that in their Tour Guide app, they will be storing the name of the picture that a user takes, so when the app is re-opened, the picture will still display in the Gallery Screen.

Implementation of Gallery Screen (20 minutes)

Ask students to work using the Pair Programming model and follow the *Student Guide: Lesson 4* to implement GalleryScreen. To test, ask students to take a picture in the Gallery, then reconnect the MIT AI2 Companion app to see that the picture taken in a previous session still appears.

Wrap-up and Review (5 minutes)

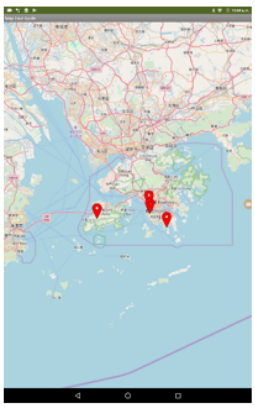
Review the computational thinking concepts learnt in this unit, following the tables in the four student guides. Also review the computational thinking practices used in this unit.

Post-Lesson Activities

1. (Required) Complete the multiple choice questions.
2. (Required) Complete the survey of learning attitudes.
3. (Required) Complete the open-ended questions on collaboration skills.
4. (Required) Complete all app features included in the Student Guides.
5. (Optional) Add more features to the app.

Reference for Multiple Screens

This flowchart is a complete version of what has been provided in the Student Guides and has been provided as a teacher resource. It shows how the three screens interact with each other, and the full functionality of the completed app.



Screen 1

LocationScreen

GalleryScreen