Make a Game

1. Synopsis

Students will create their own game apps using components and blocks they learned in previous units, Find the Gold and Food Chase. Students are given more time to experience the full process; from ideation, to design, to abstracting and modularizing. When applying the design in their code, students will reuse and remix what they have learned and code their blocks accordingly. Finally, they will receive users' feedback to direct modifications, along with testing and debugging.

2. Learning Objectives

After completing this unit, students will be able to:

1. Design a new app based on Drawing and Animation components in App Inventor.
2. Reuse and remix code used previously to make a new app.
3. Design and code an app incrementally, by developing a plan for step-by-step implementation of a design.
4. Provide feedback and act on suggestions for improvement.

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

Unit 6: Make a Game		
1. Sequences	✔✔	Sequences are required to perform actions in the correct order.
2. Events	✔✔✔	User interaction will be driven by events.
3. Repetition		
4. Conditionals	✔✔	Conditionals are used to test for alternate actions based on certain conditions.
5. Parallelism	✔✔	Animated sprites can move simultaneously.
6. Naming	✔✔	Naming of components is necessary for organizing a complex app.
7. Operators	✔✔	Possible use of logical operators in conditionals.

8. Manipulation of data and elementary data structures	✔✔	Variables may be used to hold values such as games scores.

Computational Thinking Practices

Unit 6: Make a Game		
1. Reusing and remixing	✔✔✔	Students may reuse concepts and code from previous units.
2. Being incremental and iterative	✔✔✔	Students make a To-do list to plan and execute tasks for building their game.
3. Abstracting and modularizing	✔✔	Students may use procedures in their app.
4. Testing and debugging	✔✔✔	Students test app at different stages to make sure it works.
5. Algorithmic thinking	✔✔✔	Students use algorithms to enact different aspects of their app.

Computational Thinking Perspectives

Unit 6: Make a Game		
1. Expressing	✔✔✔	Students express creativity in building their own game app.
2. Connecting	✔✔✔	Students can connect to games they currently play.
3. Questioning	✔✔	Students can figure out how other mobile game apps work.
4. Computational identity	✔✔	Students feel accomplished after building their own game.
5. Digital empowerment	✔✔✔	Students feel empowered to create something they designed.

4. Mapping with the CSTA Standards

This table shows the alignment of this unit with the intended learning outcomes to the CSTA CS Standards. The entries in the tables indicate the expected relevance of the unit to each outcome:

2-AP-11	Create clearly named variables that represent different data types and perform operations on their values. [C] AP: Variables [P] Creating (5.1, 5.2)	Students may use variables to store information in their game.
2-AP-13	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. [C] AP: Modularity [P] Computational Problems (3.2)	Students will decompose their app into parts and steps for implementation.
2-AP-14	Create procedures with parameters to organize code and make it easier to reuse. [C] AP: Modularity [P] Abstraction (4.1, 4.3)	Students may use procedures.
2-AP-16	Seek and incorporate feedback from team members and users to refine a solution that meets user needs. [C] AP: Program Development [P] Collaborating (2.3), Inclusion (1.1)	Students work with other groups to give and receive feedback during design and development.
2-AP-17	Incorporate existing code, media, and libraries into original programs, and give attribution. [C] AP: Program Development [P] Abstraction (4.2), Creating (5.2), Communicating (7.3)	Students will reuse and remix code from previous units.
2-AP-18	Systematically test and refine programs using a range of test cases. [C] AP: Program Development [P] Testing (6.1)	Students test and debug in stages during development.

5. Learning Prerequisites

Students should have completed “Find the Gold” and “Food Chase” units in this curriculum. They should be comfortable using Canvas, Ball, and ImageSprite components in App Inventor.

6. Lesson Plan (45 minutes x 5)

Lesson 1

Time	Activity
10 min	**Introduction to the Project** 1. Ask students: a. What did you like about the Food Chase and Find the Gold apps? b. Given your new knowledge about making games with App Inventor, what other games might you develop yourselves? 2. Explain to students that they will work with a partner to make their own game app, using at leaset two of the following components: · Accelerometer · TextToSpeech · Notifier · Sound or Player And at least two of the following features: · Multiple levels · Scoring · Timer · Animated Sprites
20 min	**App Design** Students, working in the Pair Programming model, design their own apps on paper: 1. Give students the Game App Design Worksheet. 2. Ask students to complete Parts 1-4 of the Worksheet: to describe their apps, draw the layout of their apps, list the components they will need, and describe how the components work together. 3. Remind students that their apps should contain any two of the stated components and features respectively.

10 min	**Peer Feedback** Ask student groups to present and share their worksheets with another group. Groups should write down their feedback and the corresponding changes.
5 min	**Wrap-up** 1. Check in that all groups have feasible plans. 2. Ask students to modify their Design Worksheet, based on feedback from their partner group. 3. Demonstrate how to fill out the "To-do Checklist" in the final part of the Game App Design Worksheet and ask students to complete it for homework.

Lesson 2

Time	Activity
5 min	**Review and Introduction to Lesson** 1. Check that all student groups have completed the Game App Design Worksheet, especially the To-do Checklist. This will guide them through creating apps over the rest of the unit. 2. Explain the target of this lesson: add components, and implement and test one element of their apps with code blocks.
35 min	**Code the first feature of the app** Student groups work on their apps based on their worksheets. 1. Code the first feature of the app, based on their To-do Checklist. 2. Test and debug the app using MIT AI2 Companion.
5 min	**Wrap-up** 1. Check with student groups to see if they were able to accomplish implementing the first element of their app, based on their To-do Checklist. 2. Explain that student groups will implement a second element in the next lesson, based on their To-do Checklist.

Lesson 3

Time	Activity
5 min	**Review and Introduction to Lesson** Explain the target of this lesson: add components and implement and test a second element of their apps with code blocks.
20 min	**Code the Second Feature of the App** Student groups work on their apps based on their worksheets. 1. Code the second feature of the app, based on their To-do Checklist. 2. Test and debug the app using MIT AI2 Companion.
15 min	**Peer Feedback** 1. Student groups meet with another group (this should be a different group than the one they met with in Lesson 1) and share their apps with each other. Groups provide feedback in the Feedback Worksheet to the other group. 2. Ask for any volunteer groups to share what improvements they will make after seeking feedback from classmates.
5 min	**Wrap-up** 1. Ask students to update their Design Worksheet based on feedback from their partner group. 2. Explain to students that they have two more class periods to implement the remaining elements in their app. Also explain that they will be peer reviewing each other's apps in the final lesson, so they should try to accomplish as much as possible in the next lesson.

Lesson 4

Time	Activity
5 min	**Review and Introduction to Lesson** Explain the target of this lesson: implement another element from To-do Checklist in app.
35 min	**Code another feature of the app** Student groups work on their apps based on their worksheets. They should pick up where they left off in the previous lesson.
5 min	**Wrap-up** Explain to students that they have half of the final class period to implement the remaining elements of their app. Also explain that they will be peer reviewing each other's apps in the final lesson, so they should try to wrap up the current feature they are working on.

Lesson 5

Time	Activity
5 min	**Introduction to Lesson** Students will have 15 minutes to finish their app, or complete an element so that what is completed works correctly.
15 min	**Complete app** Student groups work on their apps based on their worksheets. They should pick up where they left off in the previous lesson.
20 min	**Peer Feedback** Student groups meet with another group (not one they've met with previously in this unit), and show each other their apps. Each group will complete a Feedback Worksheet for the other group, and verbally report to the other group what they've written.
5 min	**Wrap-up** 1. Ask students to complete the survey of learning attitudes and self assessment on collaboration. 2. If there is time, have a short class discussion about the project.

7. Assessment

Survey of learning attitudes

In order to evaluate students' attitude, perception, and understanding towards coding, students are required to finish a 5-point scale survey below by putting a "✓" in the appropriate box.

After completion of this unit, I think…	**Disagree**	**Somewhat disagree**	**Neutral**	**Somewhat agree**	**Agree**
Learning how to make apps makes me want to learn more about coding.					
I feel more connected to the technology around me when I make apps.					
I am excited to share this app with friends and family.					

Self Assessment on Collaboration

Ask students to reflect on how well they worked with their partner, and to answer the following questions honestly.

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

8. Screen Design and Code

Games are based on students’ own designs, so there is no screen design and code blocks for this unit.

Appendix 1
Lesson 1 Teacher's Guide

Learning Objectives

1. Develop a design for a new app and plan the detailed layout and components needed.
2. Decompose a complex development project into a set of tasks that can be accomplished in sequence.
3. Work collaboratively to design a game app.
4. Give positive and constructive feedback to their peers, and accept and integrate feedback from others.

Lesson Outline

Introduction to Project (10 minutes)

Students will be building their own game apps with partners. To set the stage, ask students about the two game apps they've just built. The only constraints will be to include a certain number of components and features in the app.

1. Ask students:
    a. What did you like about the Food Chase and Find the Gold apps?
    b. Given your new knowledge about making games with App Inventor, what other games might you develop yourselves?
2. Explain to students that they will work with a partner to make their own game apps,

where they will use at least <u>two</u> of the following components:

- Accelerometer
- TextToSpeech
- Notifier
- Sound or Player

And at least <u>two</u> of the following features:

- Multiple levels
- Scoring
- Timer
- Animated Sprites

App Design (20 minutes)

Students will work using the Pair Programming model. Remind them of the rules for Pair Programming.

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 or keyboard

With their partners, students will first design their apps on paper.

1. Hand out the *Game App Design Worksheet*.
2. Ask student groups to complete Parts 1-4 of the worksheet: to describe their apps, draw the layout of their apps, list the components they will need, and describe how the

components work together.

3. Remind students that their apps should contain any two of the stated components and features respectively.

Peer Feedback (10 minutes)

Ask student groups to present and share their worksheets with another group and write down their feedback and the corresponding changes. They should write the feedback down on the last page of the Design Worksheet. If feedback prompts them to change something in their design, they should make those changes in the worksheet.

Wrap-up (5 minutes)

1. Check that all groups have feasible plans.
2. Ask students to modify their Design Worksheets based on feedback from their partner groups.
3. Demonstrate how to fill out the "To-do Checklist" in the final part of the *Game App Design Worksheet* and ask students to complete it for homework. If there is enough time, they may start it in class.

Appendix 2
Lesson 2 Teacher's Guide

Learning Objectives

1. Use the computational thinking practice of being incremental and iterative in developing an app step-by- step, following a development plan.
2. Test and debug collaboratively with a partner.

Lesson Outline

Review and Introduction to Lesson (5 minutes)

1. Check that all student groups have completed the *Game App Design Worksheet*, especially the To-do Checklist. This will guide them through creating their apps over the rest of the unit.
2. Explain the target of this lesson: add components and implement and test one element of their apps with code blocks. This should follow what they've listed in their To-do Checklist as the first set of steps to follow.

Code the first feature of the app (35 minutes)

Student groups work on their apps based on their worksheets. Students should use the To-do Checklist from their Design Worksheet to code the first feature of the app. Remind students to test and debug the app using MIT AI2 Companion, before proceeding to the next feature.

Wrap-up (5 minutes)

1. Check with student groups to see if they were able to implement the first elements of their apps, based on their To-do Checklists.
2. Explain that student groups will implement a second element in the next lesson, based on their To-do Checklist.

Appendix 3
Lesson 3 Teacher's Guide

Learning Objectives

1. Use the computational thinking practice of being incremental and iterative in developing an app step-by- step, following a development plan.
2. Give positive and constructive feedback to their peers and also to accept and integrate feedback from others.

Lesson Outline

Review and Introduction to Lesson (5 minutes)

Explain the target of this lesson: add components for and implement and test one more element of their apps with code blocks. Students should continue to follow their To-do Checklist to implement new features in their apps.

Code another feature of the app (20 minutes)

Student groups work on their apps based on their To-do Checklists. They should follow from where they left off in the previous lesson. Remind students to test and debug as they go, and not wait until the end to test.

Peer Feedback (15 minutes)

1. Student groups meet with another group (this should be a different group than the one they met with in Lesson 1) and share their apps with each other. Groups provide feedback in the *Feedback Worksheet* to the other group. They should also provide verbal feedback to the group, and discuss their feedback together.
2. Ask for any volunteer groups to share what improvements they will make after seeking feedback from classmates.

Wrap-up (5 minutes)

1. Ask students to update their *Game App Design Worksheets* based on feedback from their partner group. Students may also need to revise their To-do Checklists to reflect updates to their design.
2. Explain to students that they have two more class periods to implement the remaining elements in their app. Also explain that they will be peer reviewing each other's apps in the final lesson, so they should try to accomplish as much as possible in the next lesson.

Appendix 4
Lesson 4 Teacher's Guide

Learning Objectives

1. Use the computational thinking practice of being incremental and iterative in developing their app step-by- step, following a development plan.
2. Test and debug their apps incrementally.
3. Work collaboratively to build and test their apps.

Lesson Outline

Review and Introduction to Lesson (5 minutes)

Explain the target of this lesson: implement another element from To-do Checklist in the app. Students should make sure one feature is working and fully test it before moving on to adding a new feature. Remind students that they will have this class period and just half of the next class to finish their app. They will be peer reviewing each other's app in the final lesson of the unit.

Code another feature of the app (35 minutes)

Student groups work on their apps based on their To-do Checklists. They should follow on where they left off in the previous lesson.

Wrap-up (5 minutes)

Explain to students that they have just half of the final class to implement the remaining elements in their app. Their goal should be to implement the current feature they are working on. Also, remind students they will peer review each other's apps in the last class of the unit.

Appendix 5
Lesson 5 Teacher's Guide

Learning Objectives

1. Use the computational thinking practice of being incremental and iterative in developing their app step-by-step, following a development plan.
2. Give positive and constructive feedback to their peers, and accept and integrate feedback from others.

Lesson Outline

Introduction to Lesson (5 minutes)

Students will have 15 minutes to finish their app, or complete an element so that what is completed works correctly. Encourage student groups to make sure their app works, even if they haven't completed all features they initially planned for the app.

Complete app (15 minutes)

Student groups work on their apps based on their To-do Checklists. They should focus on having their apps in a working state.

Peer Feedback (20 minutes)

Student groups meet with another group (not one they've met with previously in this unit), and show each other their apps. Each group will complete a *Feedback Worksheet* for the other group, and verbally report to the other group what they've written.

Wrap-up (5 minutes)

1. Ask students to fill out the survey of learning attitudes, and self assessment on collaboration.
2. If there is time, have a short class discussion. Pose the following questions:
    a. How did it feel to design and build an app of your own design from scratch?
    b. Did you accomplish all you wanted to with the app?
    c. Would you do anything differently, based on your experience, or from the feedback you received?
    d. What kind of feedback was most helpful?