Unit 4:

Introduction to Programming

EXPLORING COMPUTER SCIENCE

Introduction

Programming is one of the creative processes that can transform ideas into reality. The intention of this unit is to highlight what artifacts can be created by using programming as a tool. As with the previous unit, students will create projects that reflect the diversity of interests in the classroom and that are personal to individual students.

Scratch provides an environment that lends itself to "tinkering". The drag and drop nature of the blocks moves the focus away from messy syntax and allows for making modifications quickly. As students work through the unit, they should apply the problem solving processes introduced in Unit 2 and reinforced in Unit 3 and reflect on the processes they use to determine the appropriate Scratch code for their projects. They should engage in discussions of why a particular set of instructions didn't work the way they thought they would and in discussions of "what if" scenarios. It is through these discussions that you can help students connect mathematics and logic to computation in programs and highlight the various abstractions they are using in creating their projects. It is also through these conversations that programming constructs such as conditionals, iteration and looping can be highlighted.

The projects listed are examples only and represent a minimal set of activities. When students complete the assigned projects, they should expand their work by adding features, collaborating with other students, and adding more personalization. There are many more projects on the Scratch website (http://scratch.mit.edu). The website is also a vibrant community. Students should be encouraged to become part of the community where they can collaborate beyond their classroom and get additional ideas for projects. There is also a Scratch Ed community for teachers (http://scratched@media.mit.edu).

Scratch 2.0 is accessed via a web browser and represents an update to the downloadable v1.4. Teachers should choose the version that best fits with school hardware and internet capabilities. There are additional features in v2.0 that were not available in v1.4, including an option for "custom" blocks. These features are not addressed in Unit 4, but can serve as possible extensions to projects where appropriate, based on student needs and interests.

Specific topics for each instructional day are listed in the overview chart on the next page.

Daily Overview Chart	
Instructional Day	**Topic**
1	Introduce the Scratch programming language, including the basic terms utilized in the language.
2-3	Practice using the basic features of Scratch in the context of creating a simple program.
4	Create a dialogue between two sprites.
5-6	Introduce the methods of moving sprites in Scratch.
7-8	Practice the concept of event driven programming through the creation of an alphabet game.
9	Introduce the concept of broadcasting via role play.
10-13	Write Scratch stories and present them to the class. Peer reviews are conducted.
14	Introduce the concept of variable.
15	Introduce the concept of conditionals.
16-17	Introduce And, Or and randomness.
18	Apply knowledge of conditionals to develop a Rock Paper Scissors program in Scratch.
19	Build on previous programming concepts to create a timer.
20-23	Create a timing game in Scratch and present it to the class. Conduct peer reviews.
24	Investigate two types of games that may provide ideas for the final project.
25	Explain final project and the rubric for the final project.
26-28	Work on final projects. Conduct peer reviews.
29	Complete final projects.
30	Presentations of final projects

Daily Lesson Plans

Instructional Day: 1

Topic Description: This lesson introduces the Scratch programming language, including the basic terms utilized in the language.

Objectives:

The students will be able to:

- Name the basic terms used in Scratch
- Create the beginning of a simple program in Scratch

Outline of the Lesson:

- Journal Entry (5 minutes)
- KWL chart about programming/Scratch (15 minutes)
- Investigate features of Scratch (35 minutes)

Student Activities:

- Complete journal entry.
- Complete KWL chart about programming/Scratch.
- Groups take turns sharing out their K's and W's.
- Pairs investigate features of Scratch and start name assignment.

Teaching/Learning Strategies

- Journal Entry: How do you think programs like Microsoft Word, Internet Explorer and Windows are made?
    - Share responses with elbow partner.
    - Discuss what it means to program a computer. Remind students that in the previous unit they used a markup language to provide instructions to the computer on the layout and content of web pages. Programming languages are used to translate algorithms into a language that a computer can execute.
- KWL chart
    - Students meet with groups and each group completes a KWL chart. (Know, Want to Learn, Learned)
    - Groups take turns sharing out their K's and W's orally. Encourage them not to repeat anything that has already been said.
    - Put KWL charts up in the classroom; tell students that they will finish the L towards the end of the unit.
- Investigate features of Scratch
    - Address how sound will be handled in the classroom.
        - Scratch lends itself to playing sounds so it can get noisy. Headsets with microphones are one possible way to address it.
    - Assign students to pairs. Remind students that in pair programming one person is the "driver" and does the clicking and typing. The other person is the "navigator" and describes to the driver

what to do at each step. Students should trade roles every 5-10 minutes. Keep track of the time and announce that students should switch at even frequencies. Make sure students trade and that both students are contributing equally. This strategy will be used often throughout the unit.

- Display the Scratch interface and show students how to open name.sb.
- Have students explore and try to determine what the various blocks do.
  - Encourage students to experiment. They can't break the computer by dragging the wrong block.
  - Show students where they can access ScratchGettingStarted.pdf. and/or provide a printed copy for each pair of students
- Stop after about 20 minutes and do a quick debrief of what they have discovered so far. Ask questions that get students to comment on the following features:
  - Every character in Scratch is called a Sprite.
  - How to choose a Sprite from a file
  - How to paint your own sprite
  - Each sprite has its own scripts.
  - You can right click any block and select help to get more information on how to use it.
  - How to change the language in Scratch (for your English Learners)
  - How to go to full screen mode and back
  - How to switch back and forth between sprites by clicking on them
  - X and Y coordinates on the screen are shown on the bottom right below the stage
  - How to save in the proper location (the default is to save in the Scratch Projects folder (C:\\Program Files\Scratch\Projects))
- Have student pairs start creating a storyboard for a "name" that includes at least 3 "letters" from their combined names.
  - Show them the rubric for name.sb
  - Draw the letters; identify colors, and actions.

Resources:

- KWL Graphic Organizer Chart
- ScratchGettingStarted.pdf (scratch.mit.edu)
- name.sb
- Name Sample Rubric
- http://scratch.mit.edu

Name Sample Rubric

Name: _____

Do you have?	Points Possible	Yes	No	Points Earned
Have a separate sprite for each letter of your name.	5			
Have at least 3 different interesting behaviors for the letters in your name.	5			
All the letters have a behavior	4			
Use the "when green flag clicked" block	3			
Use the "forever" block	3			
Extension				
Have your name reinitialize itself when the green flag is clicked. In other words, all the letters will start off in the right location facing the correct way.	2			
TOTAL:	20			

Instructional Days: 2-3

Topic Description: This lesson provides students an opportunity to practice using the features of Scratch outlined on Day 1 in the context of creating a simple program.

Objectives:

The students will be able to:

- Complete a simple Scratch program
- Utilize the green flag feature

Outline of the Lesson:

- Journal Entry (5 minutes)
- Class discussion of journal entry (15 minutes)
- Name programs (90 minutes)

Student Activities:

- Complete journal entry.
- Share journal entry responses with the entire class.
- Write programs based on their own names.

Teaching/Learning Strategies:

- Journal Entry: What do you remember about Scratch from yesterday? What do some of the blocks do?
- Class discussion of journal entry
    - Allow students to share their responses.
    - In the process, make sure to review concepts needed to finish the name project.
    - Review rubric for name project.
    - Tell students that they will do a gallery walk of the projects.
- Name programs
    - Provide time for students to complete their storyboards.
    - Student pairs write programs based on their own names.
    - Teacher circulates room checking progress and answering questions. Remind students to switch roles frequently.
    - Remind students to save their work.

Resources:

- ScratchGettingStarted.pdf (scratch.mit.edu)
- name.sb
- Name Sample Rubric
- http://scratch.mit.edu

Instructional Day: 4

Topic Description: This lesson describes how to create a dialogue between two sprites by first creating a written dialogue.

Objectives:

The students will be able to:

- Develop a dialogue between two or more Scratch sprites
- Explain the reasoning behind how their dialogue works

Outline of the Lesson:

- Gallery walk of name projects (15 minutes)
- Develop dialogue (30 minutes)
- Student presentations (10 minutes)

Student Activities:

- Participate in a gallery walk of name projects.
- Develop a dialogue.
- Present dialogues.

Teaching/Learning Strategies:

- Gallery walk of name projects
  - Facilitate gallery walk.
    - Provide an order in which students should walk.
    - Have students write questions or comments about projects as they walk...
- Develop a Dialogue
  - Explain that they will be making a dialogue between two or more sprites.
  - Have a sample dialogue with a student as an example.
  - Make a sample program using only "say _ for _ secs" blocks.
    - Ask questions that get students to suggest that you need to add
    - the "wait _ sec" block.
    - Add a few so students can see the sprites taking turns.
  - Show students Dialogue Sample Rubric and have them create their own dialogue.
  - Circulate room and ask questions to guide students as needed.
- Student presentations
  - Have a few students volunteer to present their dialogues for the entire class.

Resources:

- Dialogue Sample Rubric

Dialogue Sample Rubric

Name: _____

Do you have?	Points Possible	Yes	No	Points Earned
Have 2 or more sprites talking in dialogue.	4			
Have 3 or more sprites talking in dialogue.	5			
All the sprites are polite and they take turns talking	4			
Each sprite says at least 3 things.	3			
The conversation starts “when green flag clicked”	4			
Extension				
Have 4 or more sprites talking in dialogue	2			
TOTAL:	20			

Instructional Days: 5-6

Topic Description: This lesson describes the methods of moving Sprites in Scratch.

Objectives:

The students will be able to:

- Explain the 3 major ways to move sprites
- Choose the appropriate method of moving to make a cat circle the bases

Outline of the Lesson:

- Journal Entry (5 minutes)
- moving.sb (20 minutes)
- Discussion of responses to questions (15 minutes)
- baseball.sb (70 minutes)

Student Activities:

- Complete journal entry.
- Pairs examine moving.sb.
- Discuss responses to questions.
- Pairs complete baseball.sb.

Teaching/Learning Strategies:

- Journal Entry: Describe in your own words how you would create a graph for a line in your Algebra class.
  - Have students work individually and then share with their elbow partners.
- moving.sb
  - Assign students to pairs. For this lesson they will use the rules of pair programming as they respond to questions and complete moving.sb
  - Circulate the room and help students respond to the questions.
- Discussion of responses to questions
  - Emphasize that the "repeat" block will do whatever is inside it n times. This behavior can be called iteration or looping.
  - Point out that iteration is a construct that is used in other programming languages.
  - Emphasize the differences between the 3 ways to move.
  - Emphasize how the sprites will reinitialize themselves when the green flag is clicked.
- baseball.sb
  - Circulate the room and help student pairs finish baseball.sb. Remind them to switch roles.
  - After a pair can get the cat around the bases, encourage them to use the "point in direction" block to get the cat to turn the correct way when running.
  - If students need a hint for the extension, show them the "next costume" and "switch to costume" blocks under the "Looks" tab.

Resources:

- Moving Project
- moving.sb
- baseball.sb

- baseball solution.sb

Moving Project

There are basically 3 ways to move sprites in Scratch. Try the file moving.sb and answer the questions below:

1. Click the green flag. What do the three animals do?
2. Look at the scripts for each of the 3 sprites. What 3 blocks do all three sprites use?
3. What blocks does the cat use to move?
4. What block does the dog use to move?
5. What block does the monkey use to move?
6. Describe in your own words how the move block works. 

7. Describe in your own words how the go to xy block works. 

8. Describe in your own words how the glide block works. 

9. Some of the blocks require x: and y: coordinates. Place the mouse over the white window and look at the mouse x: and mouse y: numbers underneath the bottom. How are the x: and y: coordinates determined in Scratch?
10. Use what you've learned about moving to get the cat to run the bases (as realistically as possible – bases are run counter clockwise) in baseball.sb. Make sure that when you click the green flag, the cat starts at home plate again.
11. Extension: Make the cat change costumes so that it looks like it is running as it circles the bases.

Instructional Days: 7-8

Topic Description: This lesson introduces the concept of event driven programming and provides practice through the creation of an alphabet learning game.

Objectives:

The students will be able to:

- Explain event driven programming
- Write a program that responds to user created events from the mouse and keyboard

Outline of the Lesson:

- Presentation of solution for baseball extension (10 minutes)
- Journal Entry (10 minutes)
- Description of Alphabet Learning Game (20 minutes)
- Alphabet Learning Game (60 minutes)
- Student presentations (10 minutes)

Student Activities:

- Present solution for baseball extra credit.
- Complete journal entry.
- Develop an Alphabet Learning game.
- Volunteers complete presentations.

Teaching/Learning Strategies:

- Presentation of solution for baseball extension
    - A student may present while others watch. If no student completed the extension, teacher presents. See baseball solution.sb from previous lesson.
- Journal entry: How do the programs on the computer know what the user wants to do next? In other words, if you are surfing the web, how does the computer know what page to go to next?
- Description of Alphabet Learning Game
    - Allow some students to share journal entry with class. Steer them towards the idea of user events (clicks, typing) driving the program and causing it to respond.
    - Scratch provides some blocks that allow you to write programs that respond to user events relatively easily.
        - When green flag clicked (we've already seen this)
        - When Sprite clicked
        - When _ key pressed
    - Share Alphabet Sample Rubric with the students.
    - Create the first letter with students. See alphabet learning.sb.
        - One strategy is to have students take turns being the drivers while you navigate, pausing and asking other students to predict what will happen.
        - Show how to create new costumes.
            - Explain that students may bring in pictures from the internet.
                - Download a .gif or .jpg.
                - Use import or paint to make it the second costume for your letter.

        - Show how to change costumes.
            - Use a “switch to costume _” block.
        - Show students how to output in talk bubbles.
            - Use a “say _ for _ sec” block.
    - Remind students that they may pick the theme of alphabet game (animals, food, etc).
- Alphabet Learning Game
    - Circulate room and answer questions as students complete the alphabet learning game.
- Voluntary student presentations
    - Facilitate two or three students in presenting.

Resources:

- alphabet learning.sb
- Alphabet Sample Rubric

Alphabet Sample Rubric

Name: _____

Do you have?	Points Possible	Yes	No	Points Earned
Have at least 10 different letters.	4			
Have a theme for your letter game (i.e. animals, food, etc.)	3			
Sprites change costume when clicked on.	4			
Sprites change costume when letter is typed on keyboard	4			
Use the "say _ for _ sec" to output what the letter stands for (i.e. "E is for Elephant")	3			
Sprites all turn to letters when the "when green flag clicked"	2			
Extension				
Use a microphone to record sounds for all the letters and play the sound when the letter is clicked or typed (i.e. "E is for Elephant")	2			
TOTAL:	20			

Instructional Day: 9

Topic Description: This lesson introduces the concept of broadcasting through role play and then provides students an opportunity to complete a broadcast event in Scratch.

Objectives:

The students will be able to:

- Broadcast events
- Listen to and respond to events they create
- Change the background of the stage

Outline of the Lesson:

- Journal Entry (5 minutes)
- Discussion of journal entry (2 minutes)
- Role Play (20 minutes)
- Scratch Summer Story (28 minutes)

Student Activities:

- Complete journal entry.
- Participate in discussion of journal entry.
- Participate in role play.
- Create a Scratch summer story.

Teaching/Learning Strategies:

- Journal Entry: What does it mean to broadcast something (example the radio station is broadcasting music right now)? If a radio or television station is broadcasting something, does that mean that everyone is listening to it?
- Discussion of journal entry
    - Have a few students share their responses.
    - Stress that even though a lot of things are being broadcast, not everyone is listening to every thing that is being broadcast.
- Role Play
    - Solicit Volunteers to be the various characters.
    - Give the performers a paper with ONLY their part. See Scratch Broadcast Role Play.
    - Pass out the chart that shows all the parts to students that are not performing. See Scratch Broadcast Role Play Interwoven.
    - The students can think of it as a three act play where the scenes change. The difference here is that there are no curtains so they will see everything change.
    - The teacher will be the director and will make sure everything and everyone is in place during each scene. The teacher can yell action before the scene starts to signify that everything checks out.
    - Each performer's paper is broken into scripts for the various scenes.
    - One performer will be in charge of setting the stage. They can do this by erasing and drawing pictures on the white board behind the stage.
    - The Cat's first two scripts end with broadcasts. The cat will tell the director (teacher) that it is time to go on to the next scene.

    - You might want to have different students perform the role play a second time. This time the teacher will only yell out when the green flag is clicked. The students can check themselves to make sure that everything is okay.
    - Interesting Questions
        - Why do The Crab and the Date have only two scripts?
        - Instead of using broadcast, do you think you could just use "wait _ secs" blocks?
        - What might be an advantage to using broadcast instead?
    - Discussion
        - In Scratch, any sprite can broadcast its own event.
        - One reason why The Cat is doing the broadcasts is because he is the last one to act in the first two scenes. Therefore he knows when the scene is over.
        - Other sprites (including the one that broadcasts the event) can receive the event and perform a script
- Scratch Summer Story
    - Show students
        - Directions: Summer Story Project
        - File to edit: summer.sb
        - Rubric: Summer Story Project Sample Rubric
    - Circulate the room and answer questions.

Resources:

- Scratch Broadcast Role Play
- Scratch Broadcast Role Play Interwoven
- Summer Story Project
- summer.sb
- Summer Story Project Sample Rubric

Scratch Broadcast Role Play

This is meant to be performed in front of a white board. This can also be done using more elaborate props. Each character's parts are broken down by events that are broadcast out to everyone. Select characters and give them their parts of the scripts. There is also a script so that observers can see the flow of the entire program.

Useful props: sunglasses, a basketball, and a bag of popcorn or chips

Characters:

The Cat: our main character

The Crab:

The Opponent:

The Date:

Stage: in charge of drawing the background of the scene on the board

Scripts for the individual actors:

The Cat

when GREEN FLAG clicked:

switch to costume: sunglasses

say: Hello!

say: I'm going to tell you about my summer.

say: I spent some time at the beach.

broadcast BASKETBALL SCENE (tell everyone it's time for the next scene)

when I receive BASKETBALL SCENE:

switch to costume: basketball

say: I played lots of ball.

broadcast MOVIE SCENE (tell everyone it's time for the next scene)

when I receive MOVIE SCENE:

switch to costume: bag of popcorn or chips

say: I went on a date. We went to the movies.

The Crab

when GREEN FLAG clicked:

show: (Go up on stage. You might want to pose like a crab by making your hands into claws.)

when I receive BASKETBALL SCENE:

hide: (Disappear from the stage)

The Opponent

when GREEN FLAG clicked:

    hide: (Disappear from the stage)

when I receive BASKETBALL SCENE:

show: (Go up on stage. You might want to pose like a basketball player.)

when I receive MOVIE SCENE:

    hide: (Disappear from the stage)

The Date

when GREEN FLAG clicked:

    hide: (Disappear from the stage)

when I receive MOVIE SCENE:

show: (Go up on stage.)

wait 2 secs:

say: I can't wait to see Get Smart. Would you believe I spent my entire check from work on these tickets? No? Would you believe that it cost me $15 for both using my student discount? No? How about a stick of gum and a nickel I found on the floor?

Stage

when GREEN FLAG clicked:

Switch to background BEACH: (Draw a picture of the beach on the white board. A sun in one corner and a wavy line for sand is fine.)

when I receive BASKETBALL SCENE:

Switch to background BASKETBALL COURT: (Draw a picture of a basketball court. Drawing the backboard and rim should be fine.)

when I receive MOVIE SCENE:

Switch to background MOVIES: (Draw a picture of a movie theater. Drawing a sign that says movies should be ok.)

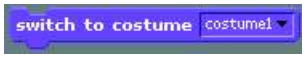
Scratch Broadcast Role Play Interwoven

	when GREEN FLAG clicked:	when I receive BASKETBALL SCENE:	When I receive MOVIE SCENE:
The Cat	switch to costume: sunglasses say: Hello! say: I'm going to tell you about my summer. say: I spent some time at the beach. broadcast BASKETBALL SCENE	switch to costume: basketball say: I played lots of ball. broadcast MOVIE SCENE	switch to costume: bag of popcorn or chips say: I went on a date. We went to the movies.
The Crab	show: (Go up on stage. Pose like a crab.)	hide: (Disappear from the stage)	
The Opponent	hide: (Disappear from the stage)	show: (Go up on stage. Pose like a basketball player.)	hide: (Disappear from the stage)
The Date	hide: (Disappear from the stage)		show: (Go up on stage.) wait 2 secs: say: I can't wait to see Get Smart. Would you believe I spent my entire check from work on these tickets? No? Would you believe that it cost me $15 for both using my student discount? No? How about a stick of gum and a nickel I found on the floor?
Stage	Switch to background BEACH: (Draw the beach.)	Switch to background BASKETBALL COURT: (Draw a basketball court.)	Switch to background MOVIES: (Draw a movie theater.)

Summer Story Project

Finish a story about what the cat did over summer. Respond to questions 1, 2, and 6 on paper.

1. Open the file summer.sb. Click the flag. What does it do so far?
2. Click on the cat and look at his script. What does the cat broadcast in the last block?
3. We'll make a basketball scene (a second script)
    a. Drag a 

block into the script section.
    b. Click the empty box and choose "basketball scene".
    c. Under looks, drag a 

block into your script
    d. Change costume1 to costume3
    e. Give the cat something to say about playing basketball over summer.
    f. Drag a 

block to the end of this second script.
    g. Click the empty box.
    h. Choose new
    i. Type in movie scene and hit ok.
4. We'll make the background change as well.
    a. Click on the stage
    b. Choose scripts
    c. Drag a 

block into the script section.
    d. Click the empty box and choose "basketball scene".
    e. Under looks, drag a 

block into your script
    f. Change background1 to basketball-court
5. Now add a third scene about going to the movies.
6. Summarize how you can use broadcast to change scenes in a story. Get your work checked off.
7. Now add in another character into each scene like in the role play (i.e. The Crab, The Opponent and The Date). These characters should show and hide.
8. Feel free to add in additional scenes.

Summer Story Project Sample Rubric

Name: _____

Do you have?	Points Possible	Yes	No	Points Earned
Answer question 1, 2 and 6	5			
Add in the second scene (basketball)	5			
Add in the third scene (movies)	5			
Add in additional characters into each scene that show and hide	5			
Extension				
Add in additional scenes	1			
TOTAL:	20			

Instructional Days: 10-13

Topic Description: Students will review how to broadcast events by developing a Scratch story and presenting it to the class.

Objectives:

The students will be able to:

- Broadcast events
- Complete a Scratch story
- Develop a Scratch story project
- Assess their peers to help them gauge their progress
- Complete a rubric
- Prepare and make a presentation of a Scratch story to the class

Outline of the Lesson:

- Introduction of project (5 minutes)
- Journal Entry (5 minutes)
- Review of brainstorming (10 minutes)
- Scratch story (35 minutes)
- Scratch story project (55 minutes)
- Peer review and discussion (30 minutes)
- Completion of Scratch story project (50 minutes)
- Presentation of stories (30 minutes)

Student Activities:

- Complete journal entry.
- Participate in discussion of brainstorming.
- Develop a Scratch story.
- Develop Scratch story project.
- Participate in peer review and discussion.
- Complete Scratch story project.
- Present stories.

Teaching/Learning Strategies:

- Introduction of project
    - Show rubric: Story Project Sample Rubric.
        - Emphasize that they will make a small presentation along with showing their story.
        - Emphasize that there is extra credit for the best stories.
    - Show example: cat story.sb
- Journal Entry: Brainstorm some ideas for your story.
- Review of brainstorming
    - Split students into groups of three.
    - Have students rotate so that each student will share brainstorms and receive feedback/suggestions from the other students.

- Develop the story
    - Have them first develop their story on paper.
- Scratch story project
    - Remind students of the problem solving that they did in unit 2 and that they can use that process to develop their overall plan and algorithm; they can also make a storyboard.
    - Circulate room and help students with projects.
    - If students are having difficulties, ask questions that get them to recognize some of these possible strategies
        - Develop their story further on paper.
        - Make a title screen or a first scene.
        - Break the project into smaller parts and focus on the next part that they can complete.
- Peer review and discussion
    - Elbow partners review each others' projects to date, checking the rubric to see which features are complete and which need additional work.
    - Elbow partners discuss the reviews. Make notes for changes to be made.
    - The feedback forms will be submitted along with other parts of the project. Students will receive points both for the feedback provided and the responses to feedback.
- Completion of Scratch story project
    - Circulate room and respond to questions.
    - Collect projects, rubrics and peer review forms.
    - Provide guidance as students prepare their presentations.
- Presentation of stories
    - Discuss the features used in the various stories. Ask students why certain things work/don't work well.

Resources:

- cat story.sb
- Story Project Sample Rubric
- Peer Review Sheet

Peer Review Sheet

Name: _____

Do you have?	Yes	No	Notes
The Story			
Have at least 3 scenes			
Have at least 4 scenes			
Have at least 3 different sprites			
Have at least 8 say or think boxes			
Animate the movement of your characters			
Use broadcast to change scenes in your story			
Have the characters take turns speaking to each other			
Have at least one conversation between characters			
Have a title scene with your name on it			
Story initializes itself when the flag is clicked			
The entire story plays once you click the flag			

Additional Feedback

Story Project Sample Rubric

Name: _____

Do you have?	Points Possible	Yes	No	Points Earned
The Story				
Have at least 3 scenes	10			
Have at least 4 scenes	5			
Have at least 3 different sprites	10			
Have at least 8 say or think boxes	10			
Animate the movement of your characters	5			
Use broadcast to change scenes in your story	10			
Have the characters take turns speaking to each other	5			
Have at least one conversation between characters	5			
Have a title scene with your name on it	10			
Story initializes itself when the flag is clicked	4			
The entire story plays once you click the flag	4			
The Presentation				
Explain an example from your program of how events (broadcast) were used to transition from one scene to another.	10			
Peer Review Sheets	12			
TOTAL:	100			

Instructional Day: 14

Topic Description: This lesson provides an introduction to the concept of variables.

Objectives:

The students will be able to:

- Explain the concept of variables
- Create examples of variables
- Explain the concept of iteration
- Create examples of iteration

Outline of the Lesson:

- Finish Presentations (25 minutes)
- Journal Entry (5 minutes)
- Make Variable Example (15 minutes)
- Enhance Variable Example (10 minutes)

Student Activities:

- Finish Presentations.
- Complete journal entry.
- Participate in a discussion of the Make Variable example.
- Enhance the variable example.

Teaching/Learning Strategies:

- Finish Presentations
- Journal Entry: Describe in your own words what the word variable means.
    - Work individually and then share with elbow partner.
- Make Variable Example
    - Ask a few students to share their responses to the journal entry.
    - Ask questions that guide students to the notion that it is the same in a program—a variable is a name that represents a value that can be changed.

      .
    - Make the variable example with the students (variable example.sb) having the students help you and build their own (in pairs) at the same time.
        - Start by explaining that you want to make a game where you earn points for picking healthy foods and lose points for picking unhealthy ones.
        - Ask questions that get students to navigate you to
            - The variables
            - Add the sprites for the banana, cheesie poofs and text that says “Click on food to eat it”.
            - Make a variable” called Good Nutrition Points.
            - Increase by 1 every time you choose the banana
                - Explain that this is an example of iteration.
            - Decrease by one when you click on the cheesie poofs
            - What happens when the green flag is clicked?

    - Explain that this is an example of initialization.

- Enhance Variable Example
    - Have student pairs enhance the variable example by
        - Adding a food that is worth 2 points when clicked on.
        - Adding a food that is –3 points when clicked on.

Resources:

- No additional resources needed

Instructional Day: 15

Topic Description: This lesson provides an introduction to the concept of conditionals.

Objectives:

The students will be able to:

- Explain the concept of conditionals
- Enhance a variable program with conditionals

Outline of the Lesson:

- Journal Entry (5 minutes)
- Discuss conditional (15 minutes)
- Age program (10 minutes)
- Age solutions (5 minutes)
- Enhance variable example (20 minutes)

Student Activities:

- Complete journal entry.
- Participate in discussion of conditions.
- Develop an Age program.
- Review Age solutions.
- Enhance the variable example.

Teaching/Learning Strategies:

- Journal Entry: What comes to mind when you hear the word "if"? What are some ways we use the word "if" in English?
    - Share with their elbow partners.
- Conditional discussion
    - Have a few students share their responses for the "if" parts and use that as a springboard.
    - In English, if is used to state a condition where something might happen if the condition is true. Hence this topic is called conditionals. Point out that this is a common computer science construct.
    - An example from computing is when a program like Microsoft Word asks you if you want to save your work when you hit close. If you click yes, it saves your changes. If you click no, it discards your changes.
    - if (some condition)
        - then do this
    - Show students "if" block in Srcatch.
        - Notice that only hexagon shaped blocks can fit within it.
        - Notice that if the condition is true, it will do anything that is enclosed within the top and bottom of the "if" block.
    - Show the students age.doc and age.sb.
        - Remind students that since we are using integers (whole numbers) > 15 it means people that are over 15 not including 15.
        - Show them how to use the slider to change the age.
- Age Program
    - For solution, see age solution.sb.

- Age solutions
    - Show a solution like age solution.sb.
    - Show an alternate solution.
        - Since numbers are integers (whole numbers) we can do “age > 2” to mean “age >= 3”.
        - To do >= in scratch, you need to use the “or” block. See age – greater-equal.sb.
- Enhance variable example
    - Instruct students to go back into their variable example about nutrition and add:
        - A message about being nutritious if the number of points becomes greater than 9.
        - A message about eating healthier food if the number of points becomes less than –4.
        - They can either have a sprite say the message or use broadcast to change the sprites/stage to convey the message.

Resources:

- Age Project
- age.sb
- age solution.sb
- age – greater-equal.sb
- variable example.sb

Age Project

You are going to finish a program that will tell you what you can do depending on your age. Use the slider to set the age.

1. Currently, it only does the first condition. Your task is to finish the program so that the cat will tell you the rest:

   If you are older than 2 "you don't need diapers"

   If you are older than 15 "you can drive"

   If you are older than 16 "you can see an R rated movie"

   If you are older than 17 "you can vote"

   If you are older than 20 "you can gamble"

   If you are older than 24 "you can rent a car"

   If you are older than 49 "you can retire"

2. If the age is less than 3, make the code tell you:
   "Sorry, you are not old enough for anything yet"

3. Feel free to add more conditions.

Instructional Days: 16-17

Topic Description: This lesson introduces And, Or and randomness. Students have an opportunity to practice utilizing these features in the context of programs.

Objectives:

The students will be able to:

- Use conditionals with And and Or to write a grade program
- Use a random number generator to write a dice program

Outline of the Lesson:

- Journal Entry (5 minutes)
- And/Or discussion (15 minutes)
- Grades program (35 minutes)
- Random discussion (20 minutes)
- Dice (35 minutes)

Student Activities:

- Complete journal entry.
- Participate in discussion of And/Or.
- Develop Grades program.
- Participate in discussion of Random.
- Complete Dice program.

Teaching/Learning Strategies:

- Journal Entry: What's the difference between And and Or? What does the word random mean in English?
  - Students should complete individually and then share with their elbow partners.
- And/Or Discussion
  - Start with a few journal entries about And and Or.
  - Kinesthetic And/Or Activity (Following is a possible set of conditions.)
    - Tell the students to stand up if the condition is true.
    - Say: If (you are a girl AND you are wearing blue) stand up.
      - Find a girl that is not wearing blue and is sitting. Ask her why she is sitting if she's a girl?
      - Ask: How many parts of the condition must be true for you to stand up if it is an AND?
    - Say: If (you are a boy OR you are wearing blue) stand up.
      - Find a boy that is standing but is not wearing blue. Ask: Why are you standing if you are NOT wearing blue?
      - Ask: How many parts of the condition must be true for you to stand up if it is an OR?
      - Ask: If both parts of the condition are true for an OR, is it ok to stand?
  - Show the students the "and" and "or" blocks in Scratch.
    - Emphasize how they are hexagon shaped and take two other hexagons.
  - Show the students Grades Project.
- Grades
  - Circulate and help students with projects.

  - If many students are stuck, build the "B" part of the code together as a class.
  - In the last minute, have students share their solutions with their elbow partners.
- Random discussion
  - Have a few students share their journal entries about what random means.
  - Ask: if I roll a pair of dice, will the numbers come out in order (2, then 3, then 4 the next roll, etc.)
  - Roll a pair of dice a few times to prove it.
  - This unpredictability is called randomness.
  - Randomness can make games more exciting.
    - For example, how many spaces will I get to move this turn?
  - Randomness is also used in computer science for simulations and in scientific/statistical experiments.
  - Walk students through dice.sb showing them the "pick random _ to _" block.
    - Explain that the numbers create the range that the random integer can fall under. The block works inclusively. Therefore 1 to 6 will produce numbers 1,2,3,4,5,6.
- Dice
  - Instruct students to finish dice.sb so that it creates a pair of dice. They can create their own look for the dice.
  - Circulate and help students with projects.
  - In the last minute, have students share their solutions with their elbow partners.

Resources:

- Grades Project
- grades solution.sb
- dice.sb
- dice solution.sb

Grades Project

Your task is to make a Scratch program that will tell you the letter grade based on the percentage.

1. Create a variable grade.
2. Double click grade to display the scroll bar.
3. When the green flag is clicked, the program should look at the value of grade and the sprite should respond with a letter as follows:

   A: greater than 90

   B: greater than 79 and less than 90

   C: greater than 69 and less than 80

   D: greater than 59 and less than 70

   F: less than 60

At Crazy High School, students only qualify for tutoring if they have a B OR a D. After it says the grade, make your program say "You qualify for tutoring" if the grade is a B or D.

Instructional Day: 18

Topic Description: This lesson requires students to apply their knowledge of conditionals to develop a Rock Paper Scissors program in Scratch.

Objectives:

The students will be able to:

- Apply knowledge of conditionals to complete a Rock Paper Scissors program

Outline of the Lesson:

- Review of Rock Paper Scissors rules (5 minutes)
- Rock Paper Scissors discussion (10 minutes)
- Rock Paper Scissors project (40 minutes)

Student Activities:

- Review Rock Paper Scissors rules.
- Participate in Rock Paper Scissors discussion.
- Complete Rock Paper Scissors project

Teaching/Learning Strategies:

- Review of Rock Paper Scissors rules
    - Lead a class discussion—students volunteer to share the rules for Rock, Paper Scissors.
- Rock Paper Scissors discussion
    - Give students a tour of rps starter.sb.
        - Show them the variables for ROCK, PAPER and SCISSORS.
        - Show students the variables for player and computer.
            - Explain how the else part works if the condition of the if is false.
        - Instruct students that they will only need to change the script that starts with "When I receive determine winner" under the computer sprite. (They may change more features if they have time.)
    - Have students work in pairs to create an outline (pseudo code) to handle all the cases for the computer choosing ROCK. Remind students that this is an algorithm.
    - Show students a working example in presentation mode (so they can't see the blocks).
- Rock Paper Scissors project
    - Have student pairs use their outline to get started. Remind them to switch roles.
    - Circulate room and ask questions to guide them as necessary.
    - Allow students to try various approaches to solving the problem.
    - If students finish, suggest they add an extension for keeping score of the wins for the computer and player.

Resources:

- rps starter.sb
- rps solution.sb
- rps solution b.sb

Instructional Day: 19

Topic Description: This lesson builds on previous concepts to create a timer.

Objectives:

The students will be able to:

- Create a timer

Outline of the Lesson:

- Review of Rock Paper Scissors solutions (10 minutes)
- Creation of a timer (15 minutes)
- Review of Timer solutions (5 minutes)
- Introduction of Timing Game (15 minutes)
- Timing Game theme (10 minutes)

Student Activities:

- Review Rock Paper Scissors solutions.
- Create a timer.
- Review Timer solutions.
- Choose Timing Game theme.

Teaching/Learning Strategies:

- Review of Rock Paper Scissors solutions
    - Review rps solution.sb and rps solution b.sb.
    - Allow students to share their own unique solutions.
- Creation of a timer
    - Explain to students that they will make a timer that will count down from 10 to 0.
    - Show students Timer Project.
    - Pairs complete Timer Project.
- Review of Timer solutions
    - Allow students to share their own unique solutions.
    - Review timer solution a.sb and timer solution b.sb.
- Introduction of Timing Game
    - Have students help build an example. (See timing.sb.)
    - Review Timing Game Sample Rubric.
- Timing Game theme
    - Circulate room and help students pick the theme of their timing game.

Resources:

- rps solution.sb (modified version of Jesse Moya's solution)
- rps solution b.sb (modified version of Jesse Moya's solution)
- Timer Project
- Timing Game Sample Rubric
- timer solution a.sb

- timer solution b.sb
- timing.sb

Timer Project

How to make a timer in Scratch:

1. Create a variable called timer.
2. When the flag is clicked, initialize the timer to 10.
3. Continually, wait 1 second and check if the timer = 0
    a. output the current time either with a sprite or just show the variable
    b. If the timer = 0 make either the background or a huge sprite say “Time’s Up”
4. When the flag is clicked, everything should start over.
5. Be creative as to what you want your program to look like.
6. Make sure the timer stops at 0 and does not continue into negatives.

Timing Game Sample Rubric

Name: _____

Do you have?	Points Possible	Yes	No	Points Earned
The Game				
Have 3 or more “timed” sprites	10			
Have 4 or more “timed” sprites	5			
Use a timer for your game	5			
Keep score (points)	10			
Give the user feedback as to how well they timed their button pressing	10			
Have a help screen with directions	5			
Does the game reset when the flag is clicked	10			
Does the game stop when it is over	5			
Does the game notify the user when it is over	10			
Does the game keep track of how many “perfects” in a row	5			
Does the game get harder as you keep playing	5			
Peer Review Sheet	20			
TOTAL:	100			

Instructional Day: 20-23

Topic Description: Students create a timing game in Scratch and participate in an Arcade Day during which they display their games.

Objectives:

The students will be able to:

- Create a timing game
- Assess their peers to help them gauge progress
- Complete their rubrics and submit their timing games
- Prepare a presentation of a Scratch program
- Evaluate their peers' timing games

Outline of the Lesson:

- Timing game (95 minutes)
- Peer Review and discussion (15 minutes)
- Completion of timing game (70 minutes)
- Arcade walk (40 minutes)

Student Activities:

- Work on timing game.
- Participate in peer review and discussion.
- Continue working on and complete timing game.
- Participate in arcade walk.

Teaching/Learning Strategies:

- Work on timing game
    - Circulate room and help students with projects.
- Peer review and discussion
    - Elbow partners review each others' projects to date, checking the rubric to see which features are complete and which need additional work.
    - Elbow partners discuss the reviews. Make notes for changes to be made.
    - The feedback forms will be submitted along with other parts of the project. Students will receive points both for the feedback provided and the responses to feedback..
- Completion of timing game
    - Circulate room and help students with projects.
    - Collect projects, rubrics and peer feedback forms.
    - Provide guidance as students prepare their presentations.
- Arcade Walk
    - Have students rotate through the room playing each other's games. Use a timer to indicate the amount of time that each student has at each computer.
    - Have students vote for the top two games out of the entire class. The vote should be based on both the content and adherence to the rubric.
    - Discuss features of games and how they conform to the rubric. What types of programming strategies did students use?

Resources:

- Timing Game Sample Rubric
- Peer Review Sheet

Peer Review Sheet

Name: _____

Do you have?	Yes	No	Notes
The Game			
Have 3 or more “timed” sprites			
Have 4 or more “timed” sprites			
Use a timer for your game			
Keep score (points)			
Give the user feedback as to how well they timed their button pressing			
Have a help screen with directions			
Does the game reset when the flag is clicked			
Does the game stop when it is over			
Does the game notify the user when it is over			
Does the game keep track of how many “perfects” in a row			
Does the game get harder as you keep playing			

Additional Feedback

Instructional Day: 24

Topic Description: Investigating Games

Objectives:

The students will be able to:

- Investigate two different types of games
- Get ideas for their final projects

Outline of the Lesson:

- Monkey game (25 minutes)
- Review of answers (5 minutes)
- Pinball game (25 minutes)

Student Activities:

- Complete Monkey game.
- Review answers.
- Complete Pinball game.

Teaching/Learning Strategies:

- Monkey game
  - Have students answer the questions in Monkey Game Project.
  - Have students enhance monkey game.sb.
- Review answers
  - See Monkey Game Project Solutions and monkey game solution.sb.
- Pinball game
  - Have students answer questions in Pinball Project.
  - Have students enhance pinball.sb.

Resources:

- Monkey Game Project
- monkey game solution.sb
- monkey game.sb
- Pinball Project
- Pinball.sb (An example that comes with Scratch)

Monkey Game Project

Answer these questions on paper:

1. Play the game by using the arrow keys. What blocks make the monkey respond to the keys?
2. Does the banana always appear in the same place?
3. What blocks do you think decide what x and y the banana should change to?
4. What are the names of the orange blocks under Variables?
5. What block(s) are used to change the score?

Make these changes to the file:

6. Customize the sprites in the game (make the characters be who you want).
7. Add another sprite that gives you 2 points if you touch it.
8. Get the game to stop at 10 points or more by telling you that you win.

Pinball Project

Open pinball.sb and answer the questions below on paper:

1. Look at the scripts for the pinball. How did the author simulate gravity?
2. How does the ball know when to "bounce" off of something?
3. Does the ball always bounce the same way when it hits something?
4. How do you think the ball determines which direction to bounce?
5. What's the purpose of the purple line at the very bottom of the game?
6. Modify the game to keep track of points and get it checked off. Write down what changes you made.
7. What other features do think would make this game better?

Instructional Day: 25
Topic Description: Introduce the final project.

Objectives:

The students will be able to:

- Make an appropriate choice of which final project they will do

Outline of the Lesson:

- Review of responses for Pinball Project (5 minutes)
- Introduction of projects (15 minutes)
- Final projects (35 minutes)

Student Activities:

- Review responses for Pinball project.
- Participate in discussion of introduction to projects.
- Start Final Projects.

Teaching/Learning Strategies:

- Review of responses for Pinball project.
- Introduction of projects
    - Review final project and final project rubric.
- Final projects
    - Circulate the room
        - Help students decide which project to do.

Resources:

- Pinball Project
- Final Project
- Final Project Sample Rubric

Instructional Day: 26-28

Topic Description: Complete final projects.

Objectives:

The students will be able to:

- Incorporate all objectives in the unit into the final project

Outline of the Lesson:

- Work on final project (80 minutes)
- Peer review and discussion (30 minutes)
- Completion of final project (55 minutes)

Student Activities:

- Work on final project.
- Participate in peer review and discussion.
- Complete final project.

Teaching/Learning Strategies:

- Work on final project
    - Circulate room and help students with projects.
- Peer Review and discussion
    - Elbow partners review each others' projects to date, checking the rubric to see which features are complete and which need additional work.
    - Elbow partners discuss the reviews. Make notes for changes to be made.
    - The feedback forms will be submitted along with other parts of the project. Students will receive points both for the feedback provided and the responses to feedback.
    - Circulate the room and make sure students understand the rubric and what they still need to accomplish to finish their project.
- Completion of final project
    - Circulate room and help students with projects.

Resources:

- Final Project
- Final Project Sample Rubric
- Peer Review Sheet

Instructional Day: 29

Topic Description: Complete final projects.

Objectives:

The students will be able to:

- Finish their KWL charts
- Complete their rubrics and turn in their final projects
- Prepare their presentations

Outline of the Lesson:

- KWL chart (15 minutes)
- Completion of final projects (40 minutes)

Student Activities:

- Complete KWL chart.
- Groups take turns sharing their L's orally.
- Finish final projects.

Teaching/Learning Strategies:

- KWL chart
    - Refresh memory on activity.
        - They already did the K (Know) and W (Want to learn), now they must fill in the L (Learned) section of their chart.
        - They should meet in their original groups.
    - Students fill in the L portion of their chart.
- Groups take turns sharing out their L's orally. Encourage them not to repeat anything that has already been said.
- Completion of final projects
    - Collect projects, rubrics and peer feedback forms.
    - Provide guidance as students prepare their presentations.
    - Note: If time permits, the gallery walk of projects can be completed here to allow more time for presentations on the last day.

Resources:

- KWL Graphic Organizer Chart.pdf (UCLA SMP)

Instructional Day: 30

Topic Description: Complete presentations.

Objectives:

The students will be able to:

- Complete a presentation on the final project

Outline of the Lesson:

- Gallery walk (10 minutes)
- Final Project Presentations (45 minutes)

Student Activities:

- Participate in gallery walk.
- Complete final project presentations.

Teaching/Learning Strategies:

- Have students do a gallery walk of the projects.
- Presentations
    - Have students take turns presenting. They will present their product as well as explain how they created the projects.
        - Help guide students by asking questions if the student does not fully explain how they wrote their program.
    - Have students in audience complete their Peer Review Sheet (the presentation portion).

Resources:

- Peer Review Sheet
- Final Project Sample Rubric

Final Project

Your task is to create a Scratch program that does one of the following:

Creates a game
Creates a presentation
Uses Storytelling
Option of your choice (cleared by teacher)

You many choose any one of the following topics (or a topic of your choice that is teacher-approved).

1. **My Community Project**

- Decide on **at least one** positive thing that you want to highlight and **at least one** thing you want to improve about your community.
- Research and use statistics to back up your conclusions. **(cite your findings)**
- Include personal comments/recording.
- Include at least one picture.
- Include a peer observation about your topic in program.
- Use **at least 6** of the 8 block options.
- Multiple scenes (3+)
- Show/hide sprites

2. **My Career Project**

- Choose **a career or two** you are interested in.
- Research and present criteria and requirements of the career(s). **(cite your findings)**
- Include personal comments/recording.
- Include at least one picture.
- Include a peer observation about your topic in program.
- Use **at least 6** of the 8 block options.
- Multiple scenes (3+)
- Show/hide sprites

3. **My Dreams Project**

- If you had no limitations what is your biggest dream(s).
- Research and present information that relates to your topic. **(cite your findings)**
- Include personal comments/recording.
- Include at least one picture.
- Include a peer observation about your topic in program.
- Use **at least 6** of the 8 block options.
- Multiple scenes (3+)
- Show/hide sprites

Final Project Sample Rubric

Name: _____

Do you have?	**Points Possible**	**Yes**	**No**	**Points Earned**
The Content				
3 or more scenes	10			
1 or more topics discussed	5			
Information cited	5			
Statistics to back up your conclusions	5			
A personal comment/recording	5			
A personal picture	5			
A comment/recording from someone else in the class	5			
The Program				
Use at least 6 of 8 block types (emphasis on control blocks, e.g.; broadcast)	10			
Scenes only show sprites that are supposed to be in that scene	10			
Program starts and restarts when green flag is clicked	10			
The Presentation				
Show and explain the contents of each scene in your project	9			
Explain how each scene changes (how the program works)	9			
Peer Grading	12			
TOTAL:	100			

Peer Review Sheet

Name: _____

Do you have?	Yes	No	Notes
The Content			
3 or more scenes			
1 or more topics discussed			
Information cited			
Statistics to back up your conclusions			
A personal comment/recording			
A personal picture			
A comment/recording from someone else in the class			
The Program			
Use at least 6 of 8 block types (emphasis on control blocks, e.g.; broadcast)			
Scenes only show sprites that are supposed to be in that scene			
Program starts and restarts when green flag is clicked			
The Presentation			
Show and explain the contents of each scene in your project			
Explain how each scene changes (how the program works)			

Additional Feedback