

Name of Project: Design your own Game

Grade: 7

Date: Nov/Dec 2015

Big Idea: Cause and Effect

DESIRED RESULTS

Established Goals (Standards, Performance Indicators, Learning Goals):

<http://scratched.gse.harvard.edu/ct/defining.html>

1. Computational concepts
 - **sequence:** identifying a series of steps for a task
 - **loops:** running the same sequence multiple times
 - **parallelism:** making things happen at the same time
 - **events:** one thing causing another thing to happen
 - **conditionals:** making decisions based on conditions
 - **data:** storing, retrieving, and updating values
2. Computational practice
 - **experimenting and iterating:** developing a little bit, then trying it out, then developing more
 - **reusing and remixing:** making something by building on existing projects or ideas
3. Computational perspective
 - **expressing:** realizing that computation is a medium of creation, "I can create."
 - **connecting:** recognizing the power of creating with and for others, "I can do different things when I have access to others."

Understandings:

A game requires a logical interaction of different parts

Essential Question:

How can I design and code a game so that the interactions are exciting?

Backward Stages: 1. Identify desired results. 2. Determine acceptable evidence. 3. Plan learning experiences and instruction.

Adapted from Wiggins & McTighe (2005) *Understanding by Design (UbD)*

Center for Technology and School Change <http://ctsc.tc.columbia.edu/>

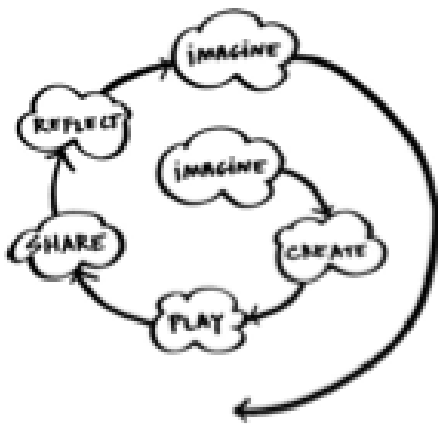
<p>Students will know:</p> <ul style="list-style-type: none"> - basic Scratch script categories and components - game design elements - tasks can occur in sequence or in parallel - tasks in sequence can occur over a number of times - tasks can be activated by events or decisions can be made based on conditions - data variables can be used to assign numerical values to games such as points, timers 	<p>Students will be able to do:</p> <ul style="list-style-type: none"> - create their own characters (sprites) and allocating various attributes to them (motion, looks) - reuse and remix other games - storyboard their games - develop games employing various design elements - reiterate games based on user feedback

<p>EVIDENCE/ASSESSMENTS:</p>
<p>Performance Task:</p> <p>Goal: Create your own interactive game Role: Game designers Audience: Fellow pioneer students Situation: Product: A game to teach other students something Standards of Criteria:</p> <ol style="list-style-type: none"> 1. Game design elements that all games should have: <ol style="list-style-type: none"> A. <u>Object</u> – What is the goal of the game? (ie collect coins, get through a maze, score points, etc.) B. <u>Operation</u> – How do you play the game? (ie Do you use keyboard keys to move? Press the spacebar to jump or shoot? Move the mouse around? etc.) C. <u>Obstacles</u> – What's the challenge that you have to overcome? (ie bad guys, a timer counting down, avoiding a danger zone, life meter, etc.) D. <u>Outcome</u> – What happens so you know you won or lost? (ie “You Won!”

screen, "Sorry, you lost" screen, celebration dance, etc.)

2. Use of at least one sprite, at least one background that make a compelling game
3. Two extensions that include data
4. Incorporation of feedback from user testing

Design Process:



Evidence/Assessments:

- Rubric to assess product
- Design Journal to assess process
- Interviews with students on three elements of computational thinking:
 - Concepts, practices and perspectives
- Projects in studio to assess first product and iterated final products

LEARNING PLAN**Learning Activities:****Session 1 Basics of Scratch (2 hours)**

Session description

In this session, students are introduced to computational creation with the Scratch programming environment by viewing a collection of sample projects and engaging in an exploratory, hands-on experience.

Objectives

The students will:

- understand the concept of computational creation, in the context of Scratch
- be able to experiment with their own Scratch-based computational creation
- become familiar with resources that support their computational creation

Session activities summary

- Introduce the concept of computational creation and the Scratch environment
- Explore the Scratch interface
- Create small Scratch projects

IMAGINE (30 min):

- a. Present account details with connected studio
- b. Scratch basics - video, resource booklet (basic layout description, Scratch cards, help function, key block description) and design journal
 - o You build projects by snapping blocks together, just as you can build things in the physical world by snapping LEGO bricks together.
 - o There are more than 100 blocks in 8 different categories.
 - o Explain about resource pack, cards and 2 minute video clips of Scratch functions.
 - o Follow Scratch - Getting started

CREATE (10 min):

- c. What can you make out of 10 Scratch blocks?

PLAY (30 min)

- d. Introduce a character with various blocks: movement, animation, sound and backgrounds
- e. Now can you:
 - i. Change the backdrop
 - ii. Try adding sound
 - iii. Try adding movement to your character

SHARE (20 min)

- f. Post to studio and comment on someone else's work

REFLECT (20 min)

- g. Assessment

Task and Game exploration (1,5 hours)

IMAGINE (20 min)

- a. What do all games have in common/exploring game design elements
- b. Present performance task to students

PLAY (10min):

- c. Showcase three concepts in games (events, loops, conditionals)
 - Maze game: <https://scratch.mit.edu/projects/10128431/>
 - Scrolling game: <https://scratch.mit.edu/projects/85012526/>
 - Pong game: <https://scratch.mit.edu/projects/10000036/>

REFLECT (20min):

- d. Reflect on events, loops, conditionals in design journal (defining these terms)

CREATE (30min):

- d. Remix the games by adding different elements

SHARE (10min):

- c. Share remixed games on studio and in groups

Session 2. Data in Games and Game design (2 hours)

PLAY (20min):

- a. Collectible game with points and win page:
<https://scratch.mit.edu/projects/85014274/>

REFLECT (20min):

- b. Reflect on broadcasting with win page, points in design journal

IMAGINE (20min):

- c. Begin imagining and designing own game (MVP= Minimum Viable Product)

- d. Imagine design elements of own game
 - e. Storyboard own game
- CREATE (60min)
- f. Spend time creating game

Session 3. Create game and post (2 hours)

- CREATE (30min)
- a. Finish creating game
- SHARE (10min):
- b. Post to studio
- PLAY (20min):
- c. Play other games
- SHARE (20min):
- d. Feedback from other groups as part of a user experience
- REFLECT (20min):
- e. What iterations can you make based on comments? in design journal
- IMAGINE (20min):
- e. Explore extensions and decide which one to include (Show extension games)
(might need to skip this depending on project completion)

Session 4. Final showcase and reflection (2 hours)

- CREATE (45min):
- a. Iterate own game and add one extension
- SHARE (5min):
- b. Post to studio
- PLAY (30min)
- a. Play two other games
- REFLECT (20min)
- b. Computational concepts, challenges and successes
- SHARE (15min)
- c. Group discussion about carrying on with Scratch

Give students 10 minutes to explore the Scratch interface in an open-ended way. One prompt is: "You have 10 minutes to make something surprising happen to a sprite." Students are encouraged to work together, ask each other for help, and

share what they are figuring out during the 10 minutes.

Resources:

Scratch	https://scratch.mit.edu/
Design challenges with Scratch	http://scratched.gse.harvard.edu/sites/default/files/designstudio.pdf
Computational thinking with Scratch	http://scratched.gse.harvard.edu/ct/index.html
Framework for computational thinking assessment	http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf
Computational thinking one page summary	http://scratched.gse.harvard.edu/sites/default/files/computational%20thinking%20article.pdf
What is computational thinking defined	http://scratched.gse.harvard.edu/ct/defining.html
Blocks guide	http://scratched.gse.harvard.edu/sites/default/files/blocks_scripts_guidex.pdf
Creative computing guide - HGSE Educator Guide	http://scratched.gse.harvard.edu/guide/files/CreativeComputing20141015.pdf
Creative Computing - Learner workbook	http://scratched.gse.harvard.edu/guide/files/CreativeComputing20140820_LearnerWorkbook.pdf
Interview protocol for computational thinking	http://scratched.gse.harvard.edu/ct/files/Student_Interview_Protocol.pdf

TO DO:

Create student accounts
 Create studio
 Create studios with different games
 Create studio for students to post
 Design journal
 Design process poster
 Resource Pack with basic layout of Scratch page
 Scratch cards
 Interview questions