Design Document

Topic: Tech Girls Scratch Code Camp.

Audience: Middle-School students attending a university summer camp.

Purpose: Teach programming concepts in an engaging way, so that interest in STEM is elevated in an under-represented population.

Form: Articulate Rise Course.

Short Narrative Description
My client is the director of technology for the distance education program at Lamar University. Each year a summer camp is hosted to teach STEM concepts such as robotics and programing to middle-school children. This year the program was designed for female students in an attempt to increase interest in the field by introducing the concepts to an under-represented population. I have proposed a course development strategy that utilizes Articulate Rise to guide the students through a variety of programing projects in a lab environment. The format of the course will have to be interactive, engaging, and straightforward in order to capture the students' attention and assist them in successfully completing the coding projects.

Problem
Traditionally the camp was taught utilizing paper handouts which quickly degraded, offered no interactivity, and required intervention from instructors and counselors. The documents were very thick packets because they needed a large selection of color screen shots to demonstrate processes that would be performed on a computer. This quickly became unwieldy because of how resource intensive the process was. There were also typos and sometimes outright incorrect instructions that had to be edited and reprinted.

Goal
Develop a seamless online experience Articulate Rise that will aid students in the completion of a one-week Scratch course. This experience should also be easily reproduced, so that any instructor can facilitate the lessons. The resulting course should be multi-faceted in that it can be taught via a presentation mode, or shared to each individual student through an LMS depending on the age and skill level of the students.

Learning Expectations

Project Goals and Objectives
1. The learner will (TLW) comprehend Sequencing by programming musical instruments.

1.1 TLW become familiar with the Scratch 2.0 interface.

1.2 TLW utilize color coding to recognize basic scripting blocks.
1.3 TLW recognize the importance of script ordering.
1.4 TLW select sprites, backgrounds, and sounds.
1.5 TLW discover the costume feature and implement costumes.

2. TLW identify screen position and practice repetition with animation.
2.1 TLW animate sprites using coordinates.
2.2 TLW produce loops to automate sprite movement.
2.3 TLW use scripts to manipulate sprite sizes.
2.4 TLW customize sprite costumes using the paint tools.
2.5 TLW differentiate between the repeat block and the forever block.

3. TLW employ variables to track, store, and manipulate data for a game.
3.1 TLW randomize sprite locations using operators.
3.2 TLW adjust sprite visibility with the *hide* block.
3.3 TLW track scores with variables and operators.
3.4 TLW develop a timer using variable and operators.
3.5 TLW tailor difficulty by customizing variable values.

4. TLW demonstrate selection by developing a chat robot.
4.1 TLW use sensing scripts to accept user input.
4.2 TLW join phrases using input and operators.
4.3 TLW store user input with variables.
4.4 TLW practice decision making with selection blocks.
4.5 TLW differentiate between *If Then* and *If Then Else* blocks.

Learning Activities

The course is divided into programing objectives that relate directly to the course goals. Each area is project based with a single project tied to a programing concept. Throughout the step-by-step projects students will be introduced to practical applications for these concepts. The learner will follow instructions for a project, create the project in Scratch visual programing, complete a quiz, and write a journal entry each day of the week. Below is a list of activities that directly relate to the course goals and objectives.

Program a musical band using Scratch (Goal 1)

- Review the interactive interface graphic (G1, O1.1)
- Build a basic music script (G1, O1.2)
- Compile the music script so it plays in the right order (G1, O1.3)
- Add sprites to the project (G1, O1.4)
- Add a background to the project (G1, O1.4)
- Switch a sprite's costume (G1, O1.5)

Create a space animation using Scratch (Goal 2)

- Enter coordinates into a move block (G2, O2.1)
- Add code blocks to a repeat loop (G2, O2.2)

- Add a looks block to a sprite to change its size (G2, O2.3)
- Draw directly on a sprite with a variety of drawing tools (G2, O2.4)
- Use the repeat block and the forever block on 2 different sprites (G2, O2.5)

Develop a ghost catching game using Scratch (Goal 3)

- Use a random number block to make a sprite appear on the screen (G3, O3.1)
- Show and hide the same sprite with *Looks* blocks (G3, O3.2)
- Add a score variable and make it change based on mouse clicks (G3, O3.3)
- Add a timer variable and make it count down (G3, O3.4)
- Adjust numbers in the operator blocks to make the game more challenging (G3, O3.5)

Design a talking chat robot using Scratch (Goal 4)

- Make the chat robot ask a question and wait for a response (G4, O4.1)
- Add spaces to each operator so the sentences are formatted correctly (G4, O4.2)
- Add a variable that can hold a text phrase (G4, O4.3)
- Have the robot respond based on an If statement (G4, O4.4)
- Have the robot respond based on an if else statement (G4, O4.4)
- Have the robot respond based on both types of statement (G4, O4.5)

Assessment

The assessment portion of the project will consist of pre-developed summative quizzes that will present the learner with questions directly related to the project they just completed. The learner does not need to reach a certain threshold to proceed to the next project, but instead will be able to take the quiz any number of times to reinforce the concepts. Once they have completed the section quiz, they will be asked to journal about their experiences with the project as a form of qualitative data collection for the camp organizers.

Environmental Resources

Learners

- Desktop in a lab environment
- Audio and Recording devices connected to desktop
- Internet access
- Scratch Offline Application
- Compatible browser (Firefox, Chrome)
- Weblink to delivery system for Articulate Rise course

Instructors

- Desktop or laptop
- Audio and recording devices connected to desktop or laptop
- Internet access
- Scratch Offline Application
- Weblink to delivery system for Articulate Rise course (with admin access)

- Projector or other large display

Evaluation

1. *How will you tell if the instructional design was successful? Who will determine this? Will there be an outside, unbiased observer? A satisfaction survey?*

The success of the implementation will be observed in two parts. Camp administrators will collect the student journals to obtain qualitative data in the form of direct student feedback. At the end of the weeklong course of study, the students will complete a google forms survey that contains Likert scale questions related to the entire program. These questions will be a form of self-assessment that allows each learner to determine their level of knowledge for each concept that was presented.

2. Does your assessment relate to your evaluation? How?

My evaluation will contain elements from the quizzes and from the journal writing prompts. This will allow the learner to better self-assess themselves in the evaluation. The repetition of programing concepts will be a key element throughout implementation and evaluation.

3. How will you determine whether your outcomes relate back to your original needs analysis and goals/objectives to determine the effectiveness of your intervention/implementation?

The effectiveness of the design will be based on a combination of the performance of the learners' quiz scores, the qualitative data observed in the journals, and the evaluation surveys. The survey results will provide valuable data on the overall design of the project, and revisions will be made based on a combination of all the previously mentioned methods.

Timeline

The one-week course will be broken up into multiple units that each directly relate to a programing concept. For example, there will be a unit based on sequencing with a project, quiz, and journal prompt that relate to a course level goal and its corresponding objectives. Each project should take approximately sixty minutes to complete, with thirty minutes remaining for the quiz and journal. The expectation is that the overall units will contain approximately an hour and thirty minutes of content. This estimate will vary greatly depending on the grade level of the learner. As stated by the client, sixth grade students may struggle to complete all of the material within the allotted time, while eighth grade students should reach the end of the course in a week.

Learner Chronology

Unit 0 – Introduction (activities = 20 minutes total)

- About Scratch (5 minutes)
- Interface Tutorial (15 minutes)

Unit 1 – Sequencing (activities = 90 minutes total)

- Project 1 – Rock Band (60 minutes)
- Unit 1 Quiz (15 minutes)
- Unit 1 Journal (15 minutes)

Unit 2 – Repetition (activities = 90 minutes total)

- Project 2 – Space Animation (60 minutes)
- Unit 2 Quiz (15 minutes)
- Unit 2 Journal (15 minutes)

Unit 3 – Variables (activities = 90 minutes total)

- Project 3 – Ghost Clicker (60 minutes)
- Unit 3 Quiz (15 minutes)
- Unit 3 Journal (15 minutes)

Unit 4 – Selection (activities = 90 minutes total)

- Project 4 – Chat Bot (60 minutes)
- Unit 4 Quiz (15 minutes)
- Unit 4 Journal (15 minutes)