

---

## VITAL INFORMATION

---

|                                |  |
|--------------------------------|--|
| <b>Subject(s):</b>             | Careers, Computer Fundamentals 1-2   |
| <b>Topic or Unit of Study:</b> | Software Development   |
| <b>Grade/Level:</b>            | 9-12   |
| <b>Objective:</b>              | <p>At the conclusion of this lesson students will be able to:</p> <ol style="list-style-type: none"><li>1. Install Scratch on their school computers (and be able to repeat the process at home).</li><li>2. Locate the majority of Scratch documentation and resources, both installed locally and available on the internet, and use it appropriately.</li><li>3. Write/draw, refine, and execute Scratch code.</li><li>4. Test the Scratch development environment and their own code.</li><li>5. Demonstrate many standard programming constructions such as conditions, repetition, variables, and object access.</li></ol> |
| <b>Summary:</b>                | Students learn about programming by exploring Scratch as if they were evaluating it for potential use by their company.  |

---

## IMPLEMENTATION

---

|                          |   |
|--------------------------|---|
| <b>Learning Context:</b> | The students have recently conducted (job) interviews and spoken about careers and "areas of interest." The students have now just been hired to form a new department at Linden Research, the maker of Second Life. The company wants to add an interface that enables residents to program their avatars. The company is looking to purchase an existing product (or its producer) in order to jump start development. The new department's task is to thoroughly test Scratch to determine whether it meets the expected need, particularly in areas of support, reliability, and capability. A future lesson will have students in charge of creating Scratch programs for demos to venture capitalist investors, the product rollout, and publicity campaign. These will be similar to the Scratch preview video they recently viewed. |
| <b>Procedure:</b>        | <ol style="list-style-type: none"><li>1. Explain the scenario/learning context. Make use of the still shot about the Second Life millionaire from the ShiftHappens video, a screen shot from Second Life, and the first frame of the Scratch tutorial video. Explain that the final product will be an animation as</li></ol>   |

described above, but that there will be two intermediate "deliverables."

2. The customer support department is concerned about how this development will affect it and desires a comprehensive report about the resources available to help residents as they program. The list should identify the resource, describe where it is located, and state whether it is more appropriate for a hands-on person, people person, or information person. Since the students will need similar resources in order to learn how Scratch works, they should happen across most of them automatically. Show an example of an acceptable format (see instructional materials below).

3. The company needs to test the capabilities and reliability of the product. A test suite will be constructed that puts the product through its paces to explore all capabilities and ensure that they function as advertised. This suite will be consulted whenever the product is upgraded to ensure backward compatibility. It will also be "mined" for examples to be used in training sessions. Students need to write code to exercise at least two of the code blocks in each category: motion, looks, sound, pen, control, sensing, numbers, and variables. They should pick blocks so that they have at least one block in each category which does not overlap with those of their neighbors. This is an informal assurance of adequate test coverage.

4. Guide students through installation, either verbally or using the click after me approach. (I will attempt the latter.) We have the installation program available on the P: drive and will have students install it in their home directories on their network drive from where they can double click to run it. Have all students create a project with rudimentary code (e.g., when Sprite1 clicked broadcast "That tickles"), store it, exit the environment, restart, reload, and ensure that the project has survived.

5. Allow students to work and just experiment. When they start asking questions, inform them, when appropriate, that they will need to have completed most of the help deliverable before the support department can come to their aide.

6. Inform students towards the end of the first class period that they should have at least a couple of test cases complete. At the beginning of the second day show students how they can comment their code by having the sprites think before they act. Demonstrate also how the broadcast mechanism can be leveraged to achieve modularity. Lastly remind them that while they are testing they should think about how they might use the discovered capabilities in their own animation.

7. As students assert that their work is complete, have them

demonstrate. As they do, sample for understanding, and complete the rubric.

**Differentiated Instruction:**

Point out that the user interface is available in numerous languages and that the resources they find are available to themselves, particularly the ones which match the students "area of interest."

**Sample Student Products:**

I will be able to provide these after the lesson. A fairly complete list of resources can be made in advance, but students may find others. There will be no absolute answers for the testing portion, but the project files can all be saved to provide samples.

**Collaboration:**

Students will work collaboratively & individually. Students will work in groups of 2.

**Time Allotment:**

2 class periods. 55 Min. per class.

**Author's Comments & Reflections:**

This lesson is recommended only for GATE-level high school classes and small ones at that.

Re. collaboration: Students must consult with their neighbors so that their test cases don't completely overlap. Informal work in pairs and trios is perfectly reasonable, especially for people persons. Students will need to turn in their own deliverables, however.

Re. time allotment: I expect this lesson to take two to three days, but also produce results which students can use to jump start the next phase, their animations.

## **MATERIALS AND RESOURCES**

---

**Instructional Materials:**

Sample of report to customer service department, mostly so kids can get the headings on their first paper the way the teacher would like it.

**Attachments**

- |                                      |
|--------------------------------------|
| 1. <a href="#">Scratch Resources</a> |
|--------------------------------------|

**Resources:**

- Materials and resources:  
Scratch software
- Technology resources:  
Videos ShiftHappens, Scratch tutorial, and clip from Second Life
- The number of computers required is 1 per student.

## **STANDARDS & ASSESSMENT**

---

**Standards:**

 **AZ- Career and Technical Education Programs**

- **Level** : Career Preparation (Grades 10 - 12)
- **Program** : Information Technology CIP No. 15.1200
- **Option** : Software Development - Option C
- **Competency** : 28.C USE SOFTWARE TO CREATE PROGRAMS
- **Indicator** : 28.1c Enter and modify code using a program editor

- **Indicator** : 28.7c Access program and language documentation
- **Competency** : \*9.0 UTILIZE TECHNOLOGY REQUIRED IN AN INFORMATION TECHNOLOGY WORKPLACE
  - **Indicator** : 9.5 Apply folder and directory management techniques
- **Competency** : 29.C TEST AND DEBUG TO VERIFY PROGRAM OPERATION
  - **Indicator** : 29.1c Test individual program modules
- **Competency** : 32.C WRITE CODE USING CONDITIONAL STRUCTURES
  - **Indicator** : 32.1c Compare values using relational operators (=, >, <, >=, <=, not equal)
  - **Indicator** : 32.2c Evaluate Boolean expressions
  - **Indicator** : 32.4c Construct decision statements such as if/else, if, switch case
  - **Indicator** : 32.6c Implement multiple-choice decision statements such as if/else, if, switch case
- **Competency** : 36.C IDENTIFY WAYS TO INPUT AND OUTPUT INFORMATION
  - **Indicator** : 36.5c Use graphics to create images at specified locations
  - **Indicator** : 36.2c Use input/output statements in a program
- **Competency** : 38.C EMPLOY OBJECT-ORIENTED PROGRAMMING TECHNIQUES
  - **Indicator** : 38.4c Write appropriate statements to invoke an object's accessor method(s)
  - **Indicator** : 38.5c Change the state of an object by invoking a modifier method

**Assessment/Rubrics:** Results of the resource search can be graded outside of class, but for the testing portion, students should demonstrate their work to the teacher and understanding should be sampled in class.

**Rubrics**

- |   |
|---|
| <ol style="list-style-type: none"> <li>1. <a href="#"><u>Scratch Resource Rubrik</u></a></li> <li>2. <a href="#"><u>Scratch Testing Rubrik</u></a></li> </ol> |
|---|