

VITAL INFORMATION

Subject(s):	Careers, Computer Fundamentals 1-2
Topic or Unit of Study:	Software Development
Grade/Level:	9-12
Objective:	Objective: At the conclusion of this lesson students will be able to: <ol style="list-style-type: none">1. Follow a screencast tutorial, pausing and rewinding when necessary.2. Import sprites and backgrounds and save Scratch projects.3. Code by example a simple static splash screen.4. Code by example a dynamic credit screen.5. Modify code and/or graphics to refine both kinds of screens.
Summary:	Students follow a tutorial which first explains the what and why of these two screen types and then guides them through implementation. Students demonstrate understanding by making their own modifications.

IMPLEMENTATION

Learning Context:	Students are beginning work on a Scratch animation, which can be nicely framed between an introduction (splash screen) and conclusion (credit screen) to the production. Some students are in need of concrete examples to get them started and from which they can build their animations on, and I believe this is a realistic way to incorporate those examples. In the process students are encouraged by the program scaffold to match the good coding practices exemplified and to use stepwise refinement. The exercise should bridge the gap between the previous fill in the blank exercise and the future blank pages.
Procedure:	<ol style="list-style-type: none">1. Provide students with the location of the screencast tutorial and the premade sprites and backgrounds that are used in the tutorial to speed up the exercise.2. Have students follow the tutorial and build the example.3. For a reasonable result, and one without their teacher's picture in it, they will need and want to make modifications. Require that they do.

The modifications demonstrate understanding.

Differentiated Instruction:

Since the screencast tempo can be changed, students can differentiate at least this aspect on their own. This first production is unlikely to have a voice track, but future productions will. Students can then follow with two senses, one (or both) of which might be impaired.

Sample Student Products:

The screencast includes a recording of the unmodified running program and there is a Scratch project to match. Student modifications will be available after the lesson.

Collaboration:

Students will work individually.

Time Allotment:

1 class period. 55 Min. per class.

Author's Comments & Reflections:

Re. collaboration: Students may collaborate informally as far as I am concerned, but they will need to turn in/demonstrate their own deliverables. I will be reading their code. Customizations should be unique.

Re. time allotment: I expect this lesson to take one day only and this plan is a test of that expectation.

MATERIALS AND RESOURCES

Instructional Materials:

Screencast tutorial

Resources:

- Technology resources:
 - Scratch software
 - Flash player (likely already installed as an IE plugin)
- 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 :** *3.0 DEVELOP APPROPRIATE WORK HABITS FOR SUCCESSFUL EMPLOYMENT IN INFORMATION TECHNOLOGY
 - **Indicator :** 3.4 Complete tasks with minimal supervision
 - **Competency :** *9.0 UTILIZE TECHNOLOGY REQUIRED IN AN INFORMATION TECHNOLOGY WORKPLACE
 - **Indicator :** 9.5 Apply folder and directory management techniques
 - **Competency :** 16.0 PARTICIPATE IN INFORMATION TECHNOLOGY WORK-BASED LEARNING EXPERIENCES
 - **Indicator :** 16.1 Use technology appropriate for a job in information technology
 - **Competency :** 27.C DEMONSTRATE PROGRAM ANALYSIS AND DESIGN
 - **Indicator :** 27.6c Use stepwise refinement to improve design
 - **Competency :** 28.C USE SOFTWARE TO CREATE PROGRAMS

- **Indicator** : 28.2c Compile and execute programs
- **Indicator** : 28.5c Use recognized conventions for naming identifiers and formatting code
- **Indicator** : 28.1c Enter and modify code using a program editor
- **Competency** : 29.C TEST AND DEBUG TO VERIFY PROGRAM OPERATION
 - **Indicator** : 29.1c Test individual program modules

Assessment/Rubrics:

Rubrics

- | |
|---|
| 1. Splash and Credit Screen Rubrik |
|---|