Name: ___________________________ Block: ____ Date: _________________

1.5 Project 1: Animated Storytelling

This is your first project! You and your classmates will use SNAP basics to implement an animated version of a movie, play, nursery rhyme, or other scene. You will each get a chance to put everything you've learned to work, and maybe even learn a little more in the process.

Overview

Storytelling is a great way to convey culture. But you already know that - you planned your story, know your characters and the actions they will take. This document contains the project specifications, the requirements for this project, which is how you will be graded. They are divided into two categories: Behaviour and Implementation.

Behaviour is about what the program does when someone runs it. This is how someone would normally interact with the code.

Implementation is about how you wrote your code. People would only see this if they peeked behind the curtain.

Project Specifications

Behavior

You will create a short animation in SNAP depicting a story of your choice.

- Whenever the green flag is clicked, your SNAP animation should display your chosen story line by line somewhere on the stage. (This should work correctly even if the last run was interrupted and restart.)
- The animation should advance on its own, but should do so at a pace that allows each action to complete and the viewer to read any line of text before the next line is shown or new action begins.
- In addition, the sprites must act out the story; you should not simply create a series of static backgrounds or costumes that show a stop-motion version of the story.
- When the story ends, there should be a way for the user to replay the entire animation from the beginning.
- You are free to be as creative as you like with your choice of sprites and actions.

You may choose from the costumes provided by SNAP, find images elsewhere, or create your own. (You will not be graded on your artistic skills.) You may interpret the story literally or be clever with your depiction (but don't go too far). However, all sprites, behaviors, words, and animations must be school-appropriate.

If you choose a particularly long story, you may not need to animate the whole thing. Please check with your teacher if you think your story is long enough for this.

Implementation Details

1. Fill out a Planning Page (Homework #4) for the story program. Make sure you consider all aspects of the program carefully *before* writing any code.
2. As mentioned above, your animation must display text and animate sprites. Action must be performed by sprites and must consist of more than simply changing costumes.
3. You must include the following components in your animation:
    a. At least three sprites that act in some way to contribute to the depiction of the story
    b. At least one sprite that moves
    c. At least one sprite that rotates
    d. At least one sprite that changes costume
    e. At least one sprite that is both hidden and shown at some point
4. Sprites must communicate with each other behind the scenes using broadcasts.
5. At the end of the program, the user can choose to restart the story (in a way that you programmed - clicking on the green flag again doesn’t count).

Note that multiple of these requirements may be satisfied by the same sprite (e.g. the same sprite can both move and change costume), but you must have at least three sprites, and two separate sprites must act in the animation.

Grading Scheme/Rubric

This project will be assessing you for the three learning targets on the four-point scale. Immediately below you will find the rubric. Underneath that is a list of some evidence that will demonstrate your proficiency in the learning targets.

4-Point Rubric	1 - Little or No Mastery Cannot demonstrate mastery, even with instructor assistance.	2 - Partial Mastery of Target Demonstrates partial understanding or can perform portions of the target with assistance.	3 - Meets Expectations for Target Demonstrates understanding by meeting project requirements.	4 - Advanced Demonstrates understanding exceeding the project requirements.
Learning Target #1: I can plan a program.	Plan may include some characters and actions.	Plan includes sprites needed and some actions sprites will take. Algorithm for a complex action is attempted.	Plan includes sprites needed, actions that each sprite will take, and breaks complex actions down into step-by-step algorithms.	
Learning Target #2: I can use blocks from the Motion, Looks, and Control palettes.	My program meets none or few of the implementation requirements OR I needed help to meet some of those requirements.	My program meets some of the implementation requirements OR I needed help to meet most of the implementation requirements.	My program meets all implementation requirements.	
Learning Target #3: I can write a program that tells a story.	It is unclear what is happening in my story.	Someone running my program can make a reasonable guess as to what happens in my story through animation and/or text, but it is somewhat unclear.	Someone running my program can either watch the animation OR read the lines of text and understand the entire story.	

Evidence of Learning Targets

Learning Target #1: I can plan a program.	
My plan explains the story I want to tell.	
My plan contains multiple sprites, each one representing a character in my story.	
Each of my sprites has its actions listed.	
Complex actions are identified, then split into smaller steps.	
My planning was complete *before* I started coding in SNAP.	
Learning Target #2: **I can use blocks from the Motion, Looks, and Control palettes.**	
I can make a program that runs entirely from when the 'green flag' is clicked to begin the program.	
I have sprites that move, rotate, change costumes, and disappear/appear.	
I use broadcasts to have sprites trigger actions in other sprites.	
After the story is told, the user can choose to restart the story.	
Learning Target #3: I can make a program that others can clearly view and follow along with	
I can make a program that always starts my sprites from the same location each time I run my program	
I can make a story-telling program where the animations on the canvas match the lines of text to match the story that's being told	
I can make a program that fellow classmates can run and either watch the animation OR read the lines of text and understand the entire story	