Programming with Scratch
7th Grade Alg-Geo

Day 1: Intro to Programming	

Objectives:
• What is computer programming?
• Why should students program?
• Introduce Scratch
 	
Activity: What and Why
• Lecture/discussion on what and why; Intro to Programming PPT
 	
Activity: Getting started with Scratch
• Introduce Scratch Environment
• Guided Practice:
  o Start Moving
  o Add a Sound
  o Start Dancing
  o Again and Again
  o Green Flag
  o Change Color
  o Key Press
  o Add a Sprite
  o Explore and Explore More
	Materials:
	• Intro to Programming PPT
	• Getting Started with Scratch PDF
	
	Homework:
	• Finish the Getting Started Tutorial
	
	Notes:
	• Add a portal link to the Getting Started Guide and PPT notes

Day 2: Scratch Syntax: Part 1	
Objectives: • Recognize, understand, and apply statements, expressions, conditions, and loops in Scratch **Activity: The Scratch "Language"** • Lecture/discussion on statements, expressions, conditions and loops; Scratch Language 1 PPT **Activity: Follow Me** • Guided Practice: Follow Me • Highlight Scratch structures in the program	**Materials:** • Scratch Syntax: Part 1 PPT • Cat Walk Project handout • Scratch Reference Guide • Programming Concepts Handout • Cat Walk Project Grading Rubric **Homework**: • HW 2: Cat Walk Project **Notes:** • Add Cat Walk Project, Reference Guide, Programming Concepts, Rubric, and PPT to portal • Homework should be submitted via Interact

Day 3: Scratch Syntax: Part 2	
Objectives: • Recognize, understand, and apply variables, events, and threads. **Activity: Review Scratch** • Collect and discuss the CatWalk Project **Activity: The Scratch "Language"** • Lecture/discussion on variables, events, and threads; Scratch PPT 2 **Activity: Pac Man and Pong** • Guided Practice: Demo Pac Man and Pong programs	**Materials**: • Scratch Syntax: Part 2 PPT • Pac Man program; Pong program • Game Tweak Handout • Game Tweak Grading Rubric **Homework**: • HW 3: Tweak a Game **Notes:** • Add Game Tweak, PPT, and Rubric to portal • Homework to be submitted via Interact
Day 4: Create Your First Game	
Objectives: • Assimilate what students have learned about Scratch into simple game of their own design **Activity: Review Scratch** • Collect and discuss the Tweak a Game Project **Activity: Adding Levels** • Guided Practice: Copter Game and Levels Code using Your First Game PPT **Activity: Your First Game** • Students will begin crafting a simple game of their own design that includes multiple stages (backgrounds) in a scrolling style	**Materials**: • Your First Game PPT • Your First Game Handout • Copter Game Handout • Levels Code • First Game Grading Rubric **Homework**: • HW 4: Your First Game **Notes:** • Add project, PPT, and rubric to portal • Homework to be submitted via Interact

Day 5: Final Project	
Objectives: • Assimilate what students have learned about Scratch into a project of their own design. **Activity: Review Your First Game** • Collect and discuss the First Game Project. **Activity: Create Your Own Project** • Provide and discuss project requirements • Allow students to review the Scratch project gallery • Begin drafting ideas for the final project	**Materials**: • Final Project Handout • Grading Rubric **Homework**: • Review the Scratch Project Gallery for ideas • HW 5: Final Project **Notes:** • Add project and rubric to portal • Homework to be submitted via Interact
Day 6: Final Project Sharing	
Objectives: • Present and explain final projects **Activity: Present and Share** • Students will present their final project and explain the programming used	**Materials**: • Scratch Gallery account **Homework:** • Upload final projects to the Scratch Gallery

Programming with Scratch
Assignment 2: The Cat Walk Project
(Project adapted from Trevon Blunn)

In this exercise, you will use utilize the basic feature of Scratch to control a Sprite using the keyboard. Follow the step-by-step instructions carefully and note that only the final program needs to be turned in.

Your completed project must be submitted via Interact by the due date and will be graded using the rubric shown below. Save your final program as username_catwalk and include "catwalk" in the assignment description.

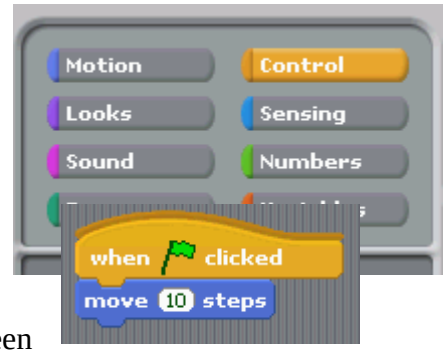
Good luck and have fun!

Grading Rubric

CATEGORY	4	3	2	1
Keyboard Control	Sprite is able to fluidly and properly move up, down, left, and right using the arrow keys.	Sprite is able to move in all four directions but movement may not be fluid and/or key controls may not work properly.	Sprite is unable to move in all four directions and/or movement is not fluid or properly controlled from the keyboard.	Sprint is unable to move.
Sprite Costumes	Sprite changes costumes (appearance) when moving left, right, up, and down; Sprite faces the direction it is moving (left or right).	Sprite changes costumes (appearance) when moving left, right, up, and down; Sprite does not face the direction it is moving (left or right).	Sprite does not change costumes (appearance) when moving left, right, up, and down; Sprite faces the direction it is moving (left or right).	Sprite does not change costumes (appearance) when moving left, right, up, and down; Sprite does not face the direction it is moving (left or right).
Use of Scripts	Final project comprised of one or two scripts that incorporate all necessary commands.	Final project comprised of three or four scripts that incorporate all necessary commands.	Final project comprised of more than four scripts that incorporate all necessary commands.	Final project comprised of more than four scripts that do not incorporate all necessary commands.

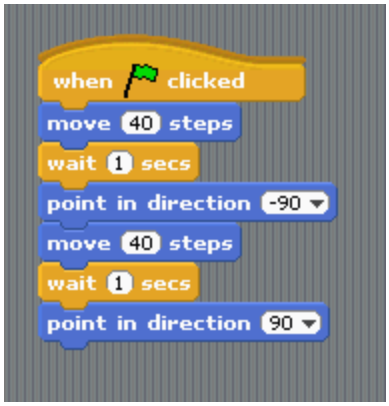
Step 1: Simple movement

In Scratch the figure is called a Sprite. One of the easiest programs to make is one that moves a sprite on the screen.

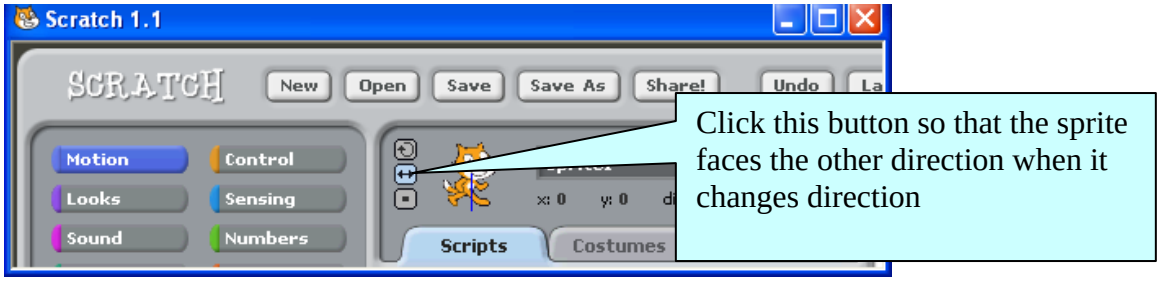


1. Click on the yellow **Control** button in the topic box
2. A series of control instructions will appear
3. Select the first one **when flag is clicked** and drag it to the Scripts area of the screen.
4. Next click the blue **Motion** button to the left of the Control button to evoke the motions instruction set
5. Drag the **move 10 steps** button across and lock it underneath the earlier command. Your program should look like the one on the right.
6. Click the **green flag** near the top right of the Scratch screen to run your program.

Step 2: Amending the program

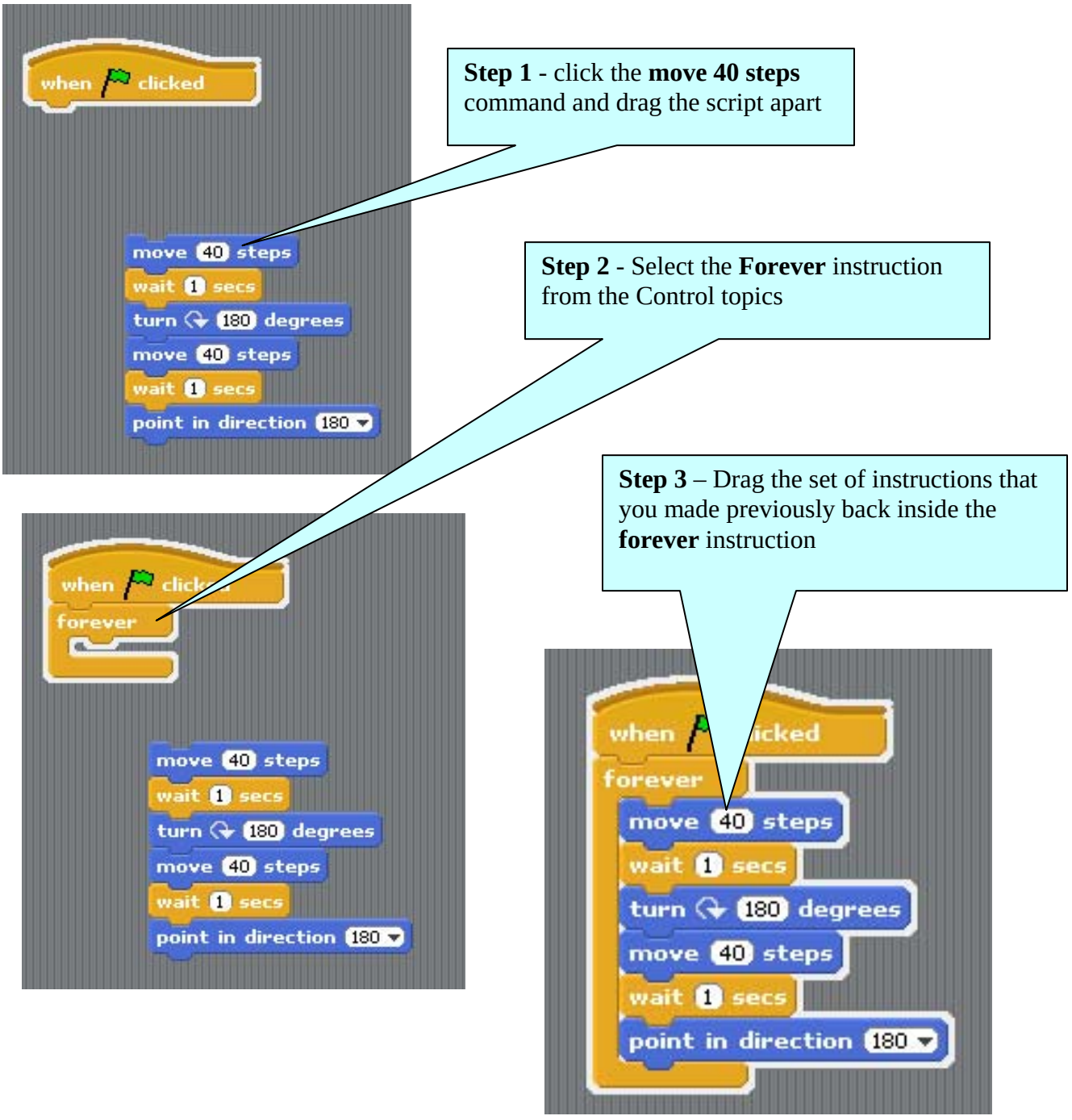


1. You can change the amount of movement by altering the value in the movement command. Change the movement value from 10 to 40.
2. Click the Control topic button again and select the wait 1 secs command
3. Click the Motion topic button and add the point in direction 90 command below it.
4. Click on the arrow beside the 90 and select -90 from the list.
5. Add another movement command and change the value to match the first
6. Add another wait command, and then another 'point' command.
7. Finally, at the top of the screen where there is a small image of the sprite, click the middle button. This makes the sprite change direction to match his movement.



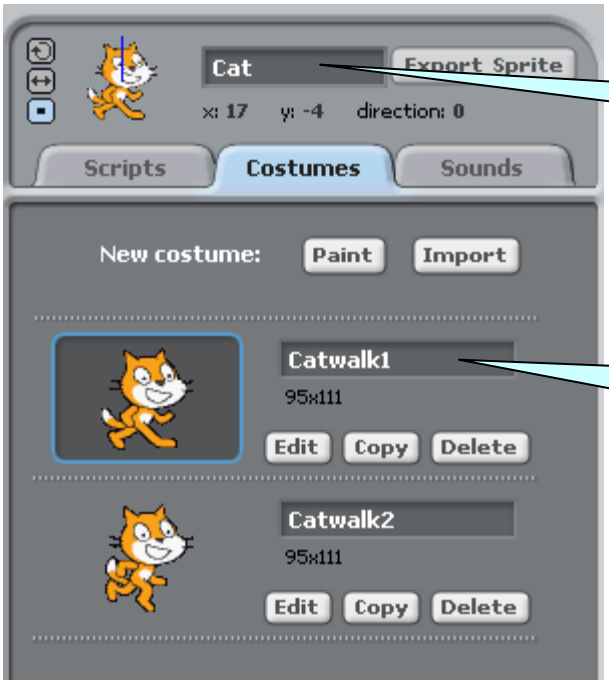
Step 3: Repeated movement (using forever)

Often we want the sprite to be animated for a period of time. In order to get the script that we have written to loop use the Forever instruction.



Step 4: Getting the sprite to walk (using costumes)

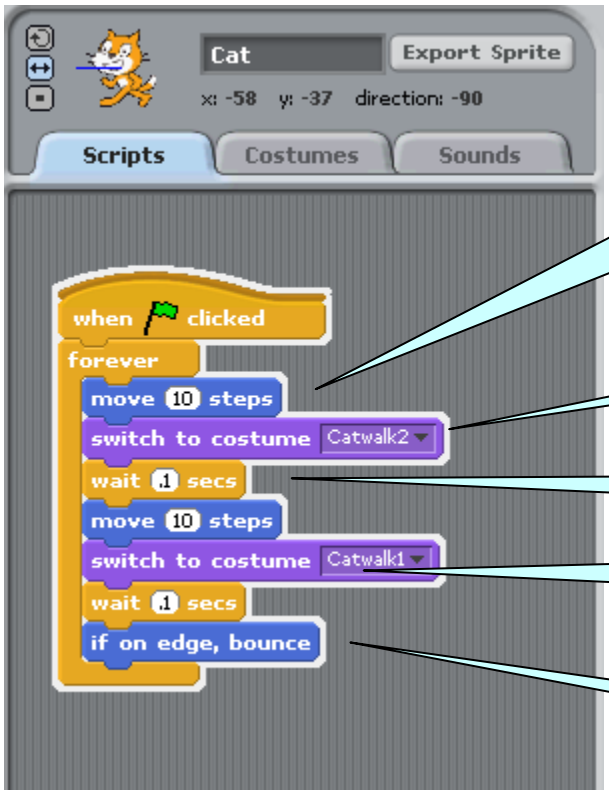
Click on the costumes tab – note that the cat sprite comes with two costumes



Rename the sprite as **Cat**

Rename the costumes **Catwalk1** and **Catwalk2**

Before completing this exercise dump your existing script into the left had side of the screen (this will delete it)



Create a new script and drag in the **move 10 steps** instruction

Drag in the **switch to costume** and select **Catwalk2** from the drop down list

Drag in a **wait** and change the value from 1 sec to 0.1 secs

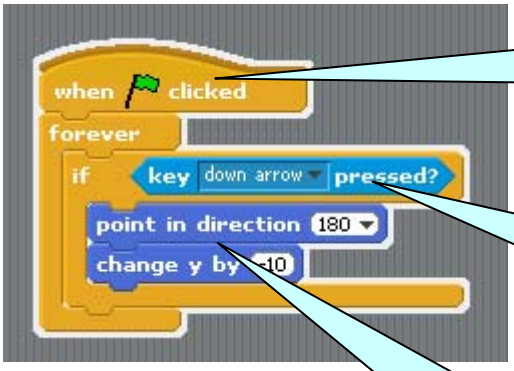
Repeat the **move**, **switch costume** and **wait** steps to go back to **Catwalk1**

Add an **if on edge, bounce** instruction

Step 5: Using the keyboard to control your sprite (using if condition instruction)

It is possible to control your sprite using the mouse keys. There is a small pointer next to the Sprite symbol at the top of the screen which can be set to any angle thereby determining the natural direction of movement of the sprite. In this exercise we want the Sprite to move exactly up and down vertically or side to side horizontally so we will use the **change x by** and **change y by** instructions to order to ensure the correct movement of the Sprite

Before completing this exercise dump your existing script into the left had side of the screen (this will delete it)

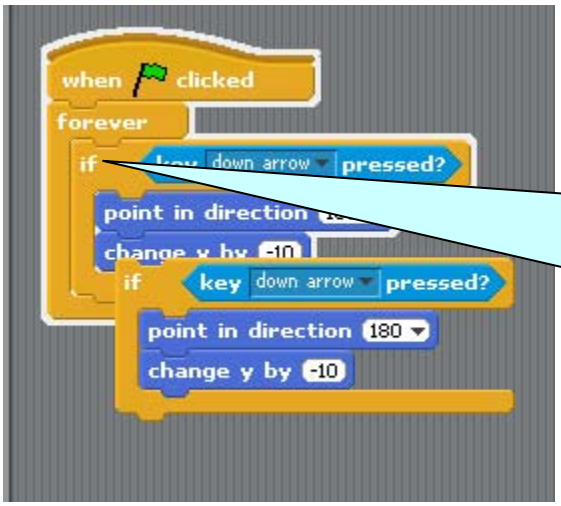


Start a new script with the **when green flag** pressed and **forever** instructions

Choose the **if condition** instruction and drag in the **key pressed** instruction from the **sensing** topics. Set the **key pressed** to **down arrow**

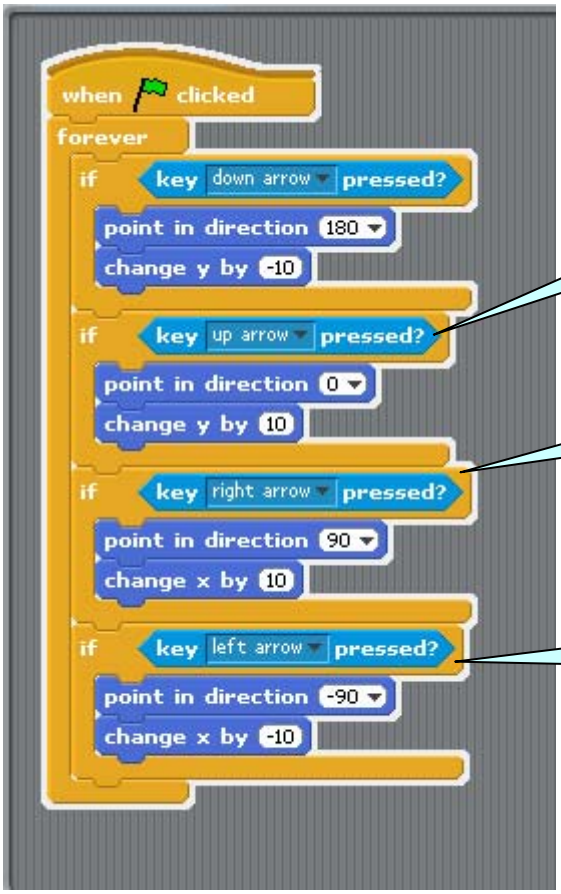
Add the **point in direction** and **change y by set** instructions **Set point in direction** to **180** and **change y by** to **-10**

We will now copy the main part of the script then edit the sub script for each movement of the mouse determined by each of the respective four arrow keys



Click on the **copy (stamp symbol) tool** then click on the word **if** in the script. This will create a copy of that part of the script. Repeat this process twice more in order to create the scripts for each different movement of the sprite

Drop the three new copied sets of instructions inside the **forever** instruction (not inside the existing **If condition** instruction)



Set the **key pressed** to **up arrow**
Set the **point in direction** to **0**
Set the **change y by** to **10**

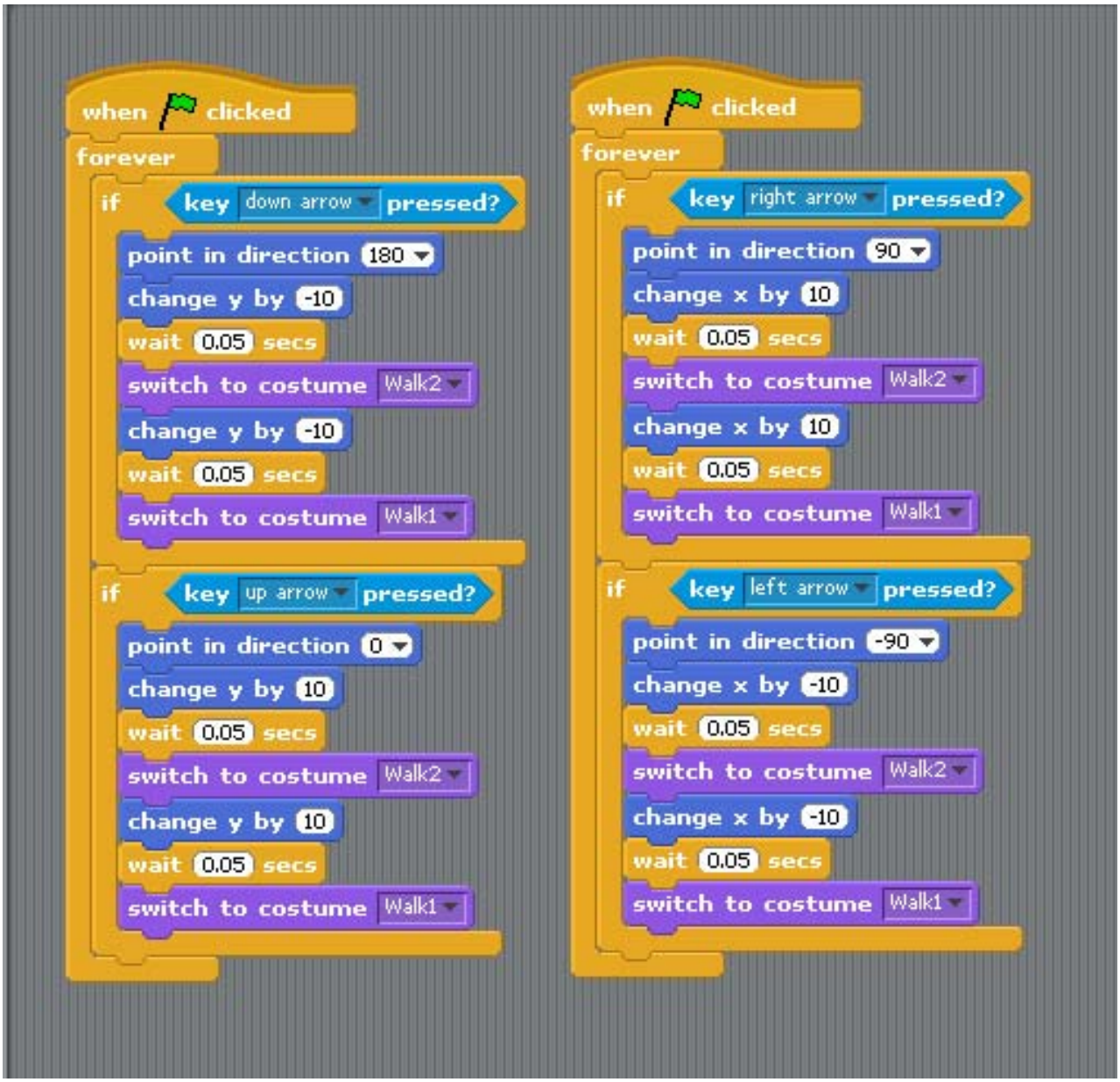
Set the **key pressed** to **right arrow**
Set the **point in direction** to **90**
Set the **change y by** to **10**

Set the **key pressed** to **left arrow**
Set the **point in direction** to **-90**
Set the **change y by** to **-10**

Click the green flag button.
It should now be possible to drive your cat sprite around the screen using the arrow keys

Step 6: Cat walking with keyboard control

Looking back over steps 4 and 5 it should be possible to make a script combining what you have already learned that allows the Cat sprite to walk around the screen.



Note that two scripts have been created here so it would display on one page and it is possible to combine all of this into one script.

Save your finished project and submit it by the due date.

Rubric Made Using:
RubiStar (http://rubistar.4teachers.org)

The Cat Walk Project

Teacher Name: **woessner**

Student Name: _____

CATEGORY	4	3	2	1
Keyboard Control	Sprite is able to fluidly and properly move up, down, left, and right using the arrow keys.	Sprite is able to move in all four directions but movement may not be fluid and/or key controls may not work properly.	Sprite is unable to move in all four directions and/or movement is not fluid or properly controlled from the keyboard.	Sprint is unable to move.
Sprite Costumes	Sprite changes costumes (appearance) when moving left, right, up, and down; Sprite faces the direction it is moving (left or right).	Sprite changes costumes (appearance) when moving left, right, up, and down; Sprite does not face the direction it is moving (left or right).	Sprite does not change costumes (appearance) when moving left, right, up, and down; Sprite faces the direction it is moving (left or right).	Sprite does not change costumes (appearance) when moving left, right, up, and down; Sprite does not face the direction it is moving (left or right).
Use of Scripts	Final project comprised of one or two scripts that incorporate all necessary commands.	Final project comprised of three or four scripts that incorporate all necessary commands.	Final project comprised of more than four scripts that incorporate all necessary commands.	Final project comprised of more than four scripts that do not incorporate all necessary commands.

Created by Patrick Woessner
http://www.pwoessner.com

Programming with Scratch
Assignment 3: Tweaking a Game
(Project adapted from http://learnscratch.org/index.php)

Modifying or "tweaking" an existing program is a very effective way to build your programming skills. In this exercise, you will have a choice of improving one of two simple, classic games; Pac Man and Pong. Each presents a unique set of challenges and opportunities; choose the game that you feel will be most rewarding.

Your completed project must be submitted via Interact by the due date and will be graded using the rubric shown below. Save your final program as username_gametweak and include "gametweak" in the assignment description.

In addition, you must submit a brief written summary of the changes/tweaks you made that explains what you changed and why those tweaks made the game better/more interesting. This should be saved as username_tweaksummary and include "tweaksummary" in the assignment description.

Good luck and have fun!

Grading Rubric

CATEGORY	4	3	2	1
Modifications	Game includes at least three significant modifications; each modification improves the over-all game experience.	Game includes at least three significant modifications; one or more modifications do not improve the over-all game experience.	Game has fewer than three modifications; each tweak improves the over-all game experience.	Game has fewer than three modifications; one or more modifications do not improve the over-all game experience.
Functionality	All elements of the game function properly; code is clear and logic is easy to follow.	All elements of the game function properly; code is somewhat unclear and/or logic is somewhat difficult to follow.	Most elements of the game function properly; code is clear and logic is easy to follow.	Several elements of the game do not function properly; code is somewhat unclear and/or logic is somewhat difficult to follow.
Rules	Rules were written clearly enough that users could easily play	Rules were written, but one part of the game needed slightly more explanation.	Rules were written, but people had some difficulty figuring out the game.	The rules were not written.
Written Summary	Summation explained every modification that was made and why those changes improved the game; explanation was clear and concise.	Summation explained every modification that was made and why those changes improved the game; explanation was slightly unclear and/or not concise.	Summation explained most modifications that were made and why those changes improved the game; explanation was clear and concise.	Summation excluded several modifications; explanation was unclear and/or not concise.

Option 1: Pac Man

This project, available from Learnscratch.org, implements a version of the traditional Pac Man game. It provides a simple method to navigate through the maze without trespassing its limits. It is a good example for an initial game project and allows for easy expansion and the addition of new features.

Video tutorials for recreating the game and the complete game code are available here. To make this project your own, however, requires some modifications. Once you have recreated the program, add at least three additional tweaks/features to the game. You may select from the following list of suggestions or come up with something of your choosing:

- A second Pac Man (with different sets of controls.)
- Additional mazes
- Additional objects for the Pac Man to eat
- A score counter for the objects eaten
- Sound effects according to the motion of the Pac Man

Remember that your final project must include a brief written summary that explains (1) the modifications/tweaks you make and (2) how they improved the overall game experience.

Option 2: Pong

This project, available from Learnscratch.org, implements a version of the traditional pong game. It includes two moving parts: the ball, which bounces on the edges of the Stage, and the paddle, which is controlled by the player. It includes sound effects, and it is a good example of an initial interactive game project.

Video tutorials for recreating the game and the complete game code are available here. To make this project your own, however, requires some modifications. Once you have recreated the program, add at least three additional tweaks/features to the game. You may select from the following list of suggestions or come up with something of your choosing:

- An additional ball
- A score counter for the number of hits
- A second paddle controlled by the key arrows (a second player)
- Assign different score values to each ball hit
- Change the speed of the ball and/or width of the paddle

Remember that your final project must include a brief written summary that explains (1) the modifications/tweaks you make and (2) how they improved the overall game experience.

Rubric Made Using:
RubiStar (http://rubistar.4teachers.org)

Making A Game : Tweaking a Game

Teacher Name: **woessner**

Student Name: _____

CATEGORY	4	3	2	1
Modifications	Game includes at least three significant modifications; each modification improves the over-all game experience.	Game includes at least three significant modifications; one or more modifications do not improve the over-all game experience.	Game has fewer than three modifications; each tweak improves the over-all game experience.	Game has fewer than three modifications; one or more modifications do not improve the over-all game experience.
Functionality	All elements of the game function properly; code is clear and logic is easy to follow.	All elements of the game function properly; code is somewhat unclear and/or logic is somewhat difficult to follow.	Most elements of the game function properly; code is clear and logic is easy to follow.	Several elements of the game do not function properly; code is somewhat unclear and/or logic is somewhat difficult to follow.
Rules	Rules were written clearly enough that users could easily play	Rules were written, but one part of the game needed slightly more explanation.	Rules were written, but people had some difficulty figuring out the game.	The rules were not written.
Written Summary	Summation explained every modification that was made and why those changes improved the game; explanation was clear and concise.	Summation explained every modification that was made and why those changes improved the game; explanation was slightly unclear and/or not concise.	Summation explained most modifications that were made and why those changes improved the game; explanation was clear and concise.	Summation excluded several modifications; explanation was unclear and/or not concise.

Created by Patrick Woessner
http://www.pwoessner.com

Programming with Scratch
Assignment 4: Your First Game

In this exercise, you will use your creativity and knowledge of Scratch to design a simple game. The theme/purpose of the game is up to you, but it must contain the following elements:

- **The game starts/resets when the Green Flag is clicked**
- **Main character can be controlled from the keyboard**
- **Minimum of two levels**
- **Minimum of one sound effect**
- **Ability to win or lose**

Your may wish to refer to the Copter Game code if you need help controlling your main character (sprite) with the keyboard. The Levels program provides a simple example of how to add a second background/level to your game. Remember that this game is not simple a reproduction of the Copter Game; be original and produce something that utilizes your creativity and knowledge of Scratch.

Your completed project must be submitted via Interact by the due date and will be graded using the rubric on the next page. Save your final program as username_firstgame and include "firstgame" in the assignment description.

Good luck and have fun!

Grading Rubric

CATEGORY	4	3	2	1
Start of Game	Green Flag starts/resets game to correct player position and background.	Green Flag starts/resets game to correct player position but background incorrect.	Green Flag starts but does not reset either player position or background.	No Green Flag in program.
Player Control	Main character controlled from keyboard using simple/logical keys and controls work correctly.	Main character controlled from keyboard but keys not simple/logical OR controls work incorrectly.	Main character controlled from keyboard but keys not simple/logical AND controls work incorrectly.	No keyboard control of main character.
Levels	Contrasting colors were used to give each of the two levels visual appeal; levels uncluttered and easy to navigate.	Contrasting colors were used to give each of the two levels visual appeal; levels uncluttered and easy to navigate.	Contrasting colors and "borrowed" graphics were used to give the cards and gameboard visual appeal.	Little or no color or fewer than 3 graphics were included.
Sound Effect	Sound effect used effectively with Sprite to enhance game-play.	Sound effect used with Sprite but does not enhance game play.	Sound effect used with Sprite detracts from game-play.	No sound effect used in the game.
Win/Lose	Player able to win and lose the game; win-lose evident to player when event occurs.	Player able to win and lose the game; win-lose not evident to player when event occurs.	Player able to win OR lose the game.	Player unable to either win or lose the game.
Creativity	Considerable thought went into making the game interesting and fun to play as shown by creative sprite(s), sound effect(s), levels, and strategy.	Thought went into making the game interesting and fun to play, but some of the game elements (sprites, sounds. levels, strategy) lacked creativity.	Tried to make the game interesting and fun, but several of the game elements made it harder to understand/enjoy the game.	Game was not interesting or fun to play and game elements lacked creativity.

Rubric Made Using:
RubiStar (http://rubistar.4teachers.org)

Your First Game with Scratch

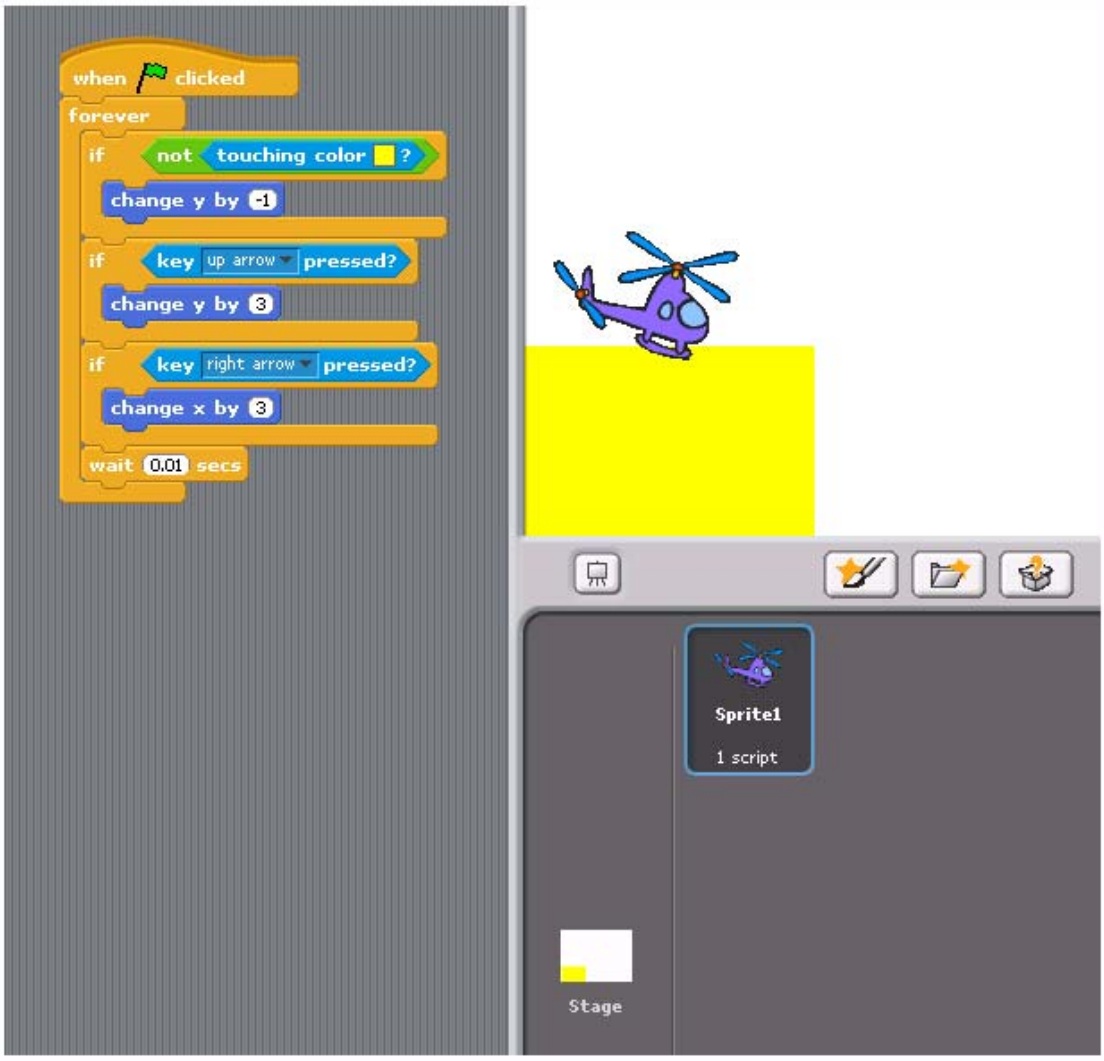
Teacher Name: **woessner**

Student Name: _____

CATEGORY	4	3	2	1
Start of Game	Green Flag starts/resets game to correct player position and background.	Green Flag starts/resets game to correct player position but background incorrect.	Green Flag starts but does not reset either player position or background.	No Green Flag in program.
Player Control	Main character controlled from keyboard using simple/logical keys and controls work correctly.	Main character controlled from keyboard but keys not simple/logical OR controls work incorrectly.	Main character controlled from keyboard but keys not simple/logical AND controls work incorrectly.	No keyboard control of main character.
Levels	Contrasting colors were used to give each of the two levels visual appeal; levels uncluttered and easy to navigate.	Contrasting colors were used to give each of the two levels visual appeal; levels uncluttered and easy to navigate.	Contrasting colors and "borrowed" graphics were used to give the cards and gameboard visual appeal.	Little or no color or fewer than 3 graphics were included.
Sound Effect	Sound effect used effectively with Sprite to enhance game-play.	Sound effect used with Sprite but does not enhance game play.	Sound effect used with Sprite detracts from game-play.	No sound effect used in the game.
Win/Lose	Player able to win and lose the game; win-lose evident to player when event occurs.	Player able to win and lose the game; win-lose not evident to player when event occurs.	Player able to win OR lose the game.	Player unable to either win or lose the game.
Creativity	Considerable thought went into making the game interesting and fun to play as shown by creative sprite(s), sound effect(s), levels, and strategy.	Thought went into making the game interesting and fun to play, but some of the game elements (sprites, sounds. levels, strategy) lacked creativity.	Tried to make the game interesting and fun, but several of the game elements made it harder to understand/enjoy the game.	Game was not interesting or fun to play and game elements lacked creativity.

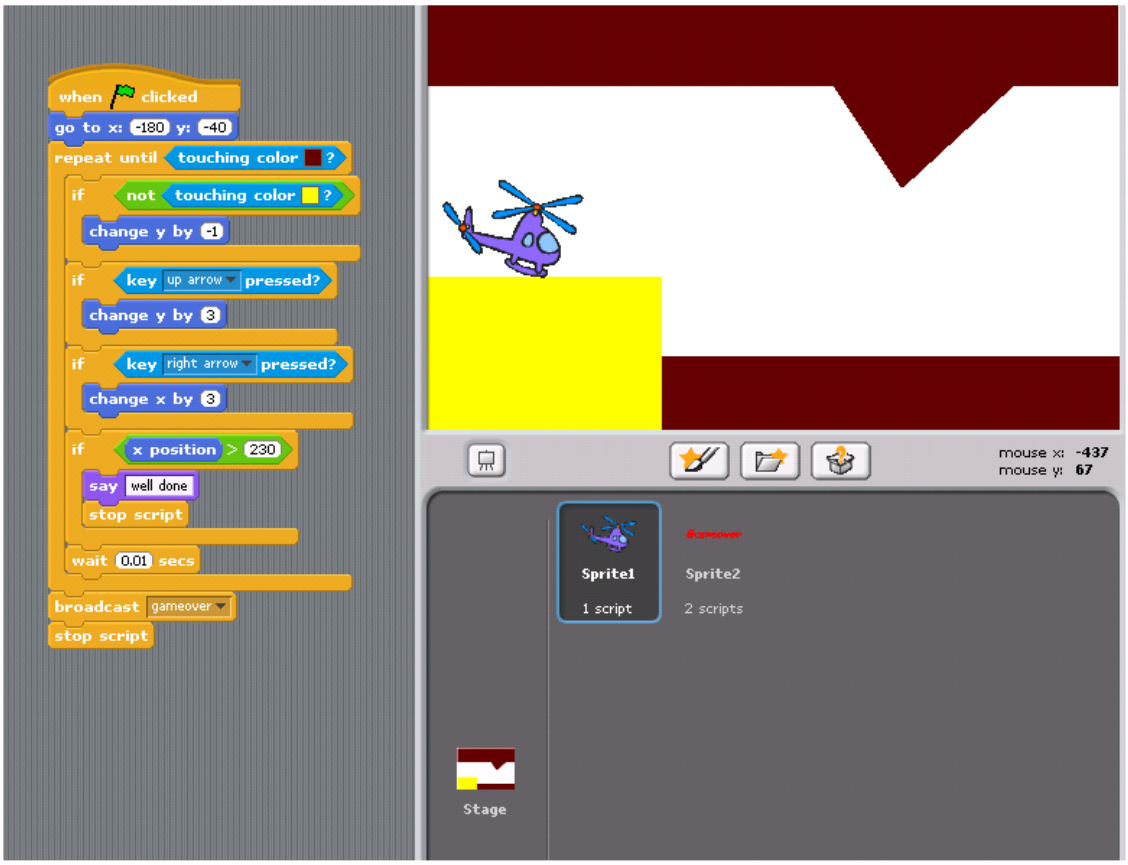
Programming with Scratch
Copter Game
Alg-Geo

(adapted from Meridian Moodle: http://www.meridianmoodle.com/mod/resource/view.php?id=756)



Copter Flight Code:

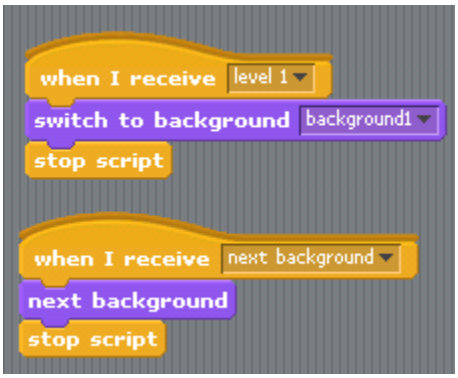
Copter Game with One Level:



Adding a Level:

Copter Script

Background Script



Programming with Scratch
Assignment 5: Final Project—Design Your Own Program

In this exercise, you will use utilize your creativity and knowledge of Scratch to create a program of your own design. You may choose to create a game, animation, or something else, but it must contain the following elements:

1. Your project must have at least two sprites, neither of which may be a cat.
2. Your project must contain at least three scripts in total (i.e., not necessarily per sprite).
3. Your project must use at least one condition and one variable.
4. Your project must use at least one sound.
5. Your project should be more complex than the simple, short examples that we have discussed in class.

Feel free to look through the projects that come with Scratch for inspiration, but your own project should not be terribly similar to any of them. Try to think of an idea on your own, and then set out to implement it. If, along the way, you find it too difficult to implement some feature, try not to fret: alter your design or work around the problem. If you set out to implement an idea you find fun, you should not find it hard to satisfy this assignment's requirements.

Your completed project must be submitted via Interact by the due date and will be graded using the rubric on the following page. Save your final program as username_finalproject and include "finalproject" in the assignment description.

Good luck and have fun!

Grading Rubric

CATEGORY	4	3	2	1
Required Elements	All required elements present in project; 2 sprites, 3 scripts, 1 condition, 1 variable, 1 sound	One required element missing	Two required elements missing	Three or more required elements missing
Clarity of Scripts	Scripts are highly logical and efficient; very easy to follow progression of logic through the program	Scripts are generally logical and efficient; fairly easy to follow progression of logic through the program	Scripts are generally logical but not efficient; somewhat difficult to follow progression of logic through the program	Scripts are neither logical nor efficient; very difficult to follow progression of logic through the program
Creativity	Considerable thought went into making the program interesting and fun to play or watch as shown by creative sprite(s), sound effect(s), and backgrounds.	Thought went into making the program interesting and fun, but some of the elements lacked creativity.	Tried to make the program interesting and fun, but several of the elements made it harder to understand/enjoy the program.	Program showed a severe lack of creativity; was neither fun nor interesting.
Visual Appeal	Color choices and graphics made the program visually appealing and showed evidence of thoughtful design.	Color choices and graphics were slightly distracting but showed evidence of attempting a thoughtful design.	Color choices and graphcis were distracting/disjointed and showed little evidence of thoughtful design.	Color choices and graphics appeared completely random and severely detracted from the program.
Knowledge of Scratch	The project demonstrated a sophisticated understanding of Scratch as evidenced by the scripts, sprites, and backgrounds.	The project demonstrated a functional understanding of Scratch as evidenced by the scripts, sprites, and backgrounds.	The project demonstrated very limited understanding of Scratch as evidenced by the scripts, sprites, and backgrounds.	The project demonstrated little to know understanding of Scratch as evidenced by the scripts, sprites, and backgrounds.

Rubric Made Using:
RubiStar (http://rubistar.4teachers.org)

Scratch Final Project

Teacher Name: **woessner**

Student Name: _____

CATEGORY	4	3	2	1
Required Elements	All required elements present in project; 2 sprites, 3 scripts, 1 condition, 1 variable, 1 sound	One required element missing	Two required elements missing	Three or more required elements missing
Clarity of Scripts	Scripts are highly logical and efficient; very easy to follow progression of logic through the program	Scripts are generally logical and efficient; fairly easy to follow progression of logic through the program	Scripts are generally logical but not efficient; somewhat difficult to follow progression of logic through the program	Scripts are neither logical nor efficient; very difficult to follow progression of logic through the program
Creativity	Considerable thought went into making the program interesting and fun to play or watch as shown by creative sprite(s), sound effect(s), and backgrounds.	Thought went into making the program interesting and fun, but some of the elements lacked creativity.	Tried to make the program interesting and fun, but several of the elements made it harder to understand/enjoy the program.	Program showed a severe lack of creativity; was neither fun nor interesting.
Visual Appeal	Color choices and graphics made the program visually appealing and showed evidence of thoughtful design.	Color choices and graphics were slightly distracting but showed evidence of attempting a thoughtful design.	Color choices and graphcis were distracting/disjointed and showed little evidence of thoughtful design.	Color choices and graphics appeared completely random and severely detracted from the program.
Knowledge of Scratch	The project demonstrated a sophisticated understanding of Scratch as evidenced by the scripts, sprites, and backgrounds.	The project demonstrated a functional understanding of Scratch as evidenced by the scripts, sprites, and backgrounds.	The project demonstrated very limited understanding of Scratch as evidenced by the scripts, sprites, and backgrounds.	The project demonstrated little to know understanding of Scratch as evidenced by the scripts, sprites, and backgrounds.