

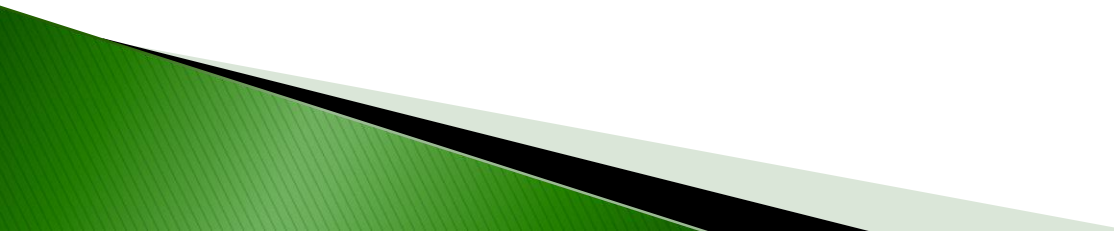
# CS0 Programming Course using Scratch

Thorna Humphries  
Software Engineering Educators' Symposium  
November 8, 2010



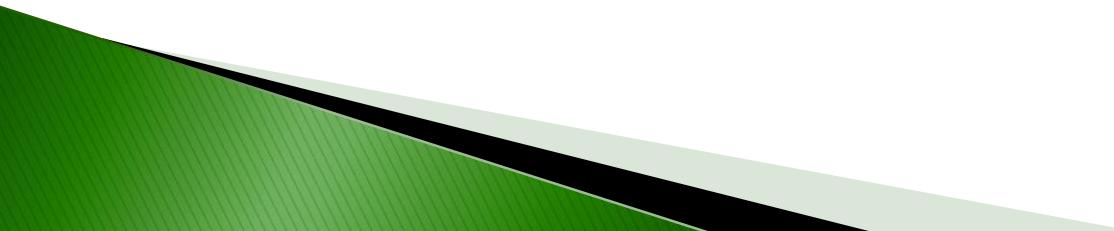
*Supported by National Science Foundation Grant DUE-0837695*

# Acknowledgements

- ▶ **Mona Rizvi**
    - Collaborator
    - Computer Science Department, Norfolk State University
  - ▶ **Debra Major, Meghan Jones, Heather Lauzun**
    - Psychology Department, Old Dominion University
- 

# The CCLI Study for a CS0 Scratch-based Course

# Motivation

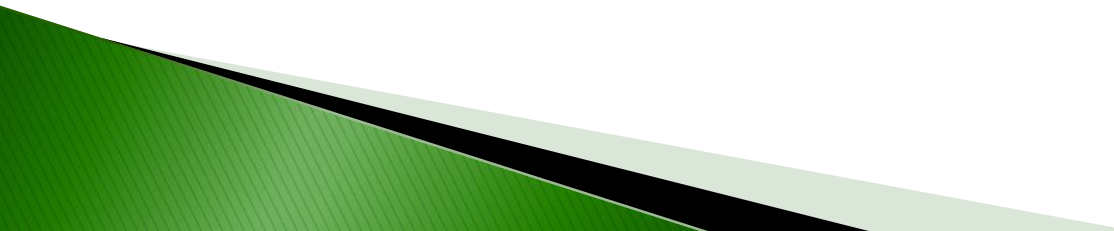
- ▶ Statistics showed that students with *weak* mathematics backgrounds had a high failure rate in CS1 and left the major within 2 years
  - ▶ Students who are classified as having weak mathematics backgrounds, either
    - Took only high school algebra
    - Took high school algebra and trigonometry
    - Entry mathematics placement exams put them in either MTH 101 Elementary Algebra and MTH 105 Intermediate Algebra
- 

# Motivation (continued)

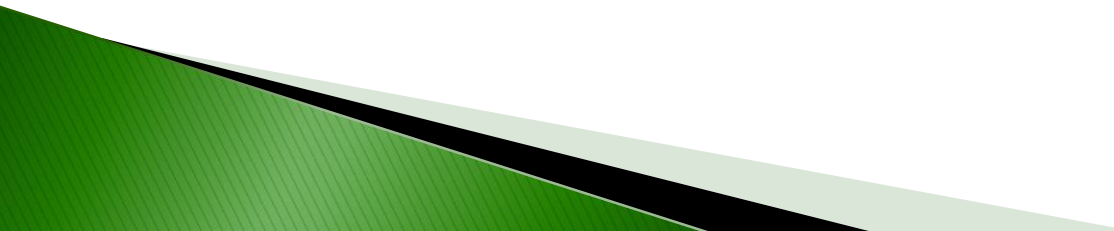
- ▶ Fall Freshman Computer Science (CS) Majors (2005–2007)

Math	Enrolled In CS1	Passed CS1	Pass Rate	Enrolled in CS2	Passed CS2	Pass Rate
MTH 101 MTH 105	38	16	42%	13	5	38%
MTH 151 MTH 153	76	55	72%	53	30	57%

# Objectives of the CCLI Study

- ▶ Establish/improve students' understanding of core programming concepts, to prepare them for success in CS1
  - ▶ Establish a positive impression of programming in the students
  - ▶ Improve retention in the CS major
- 

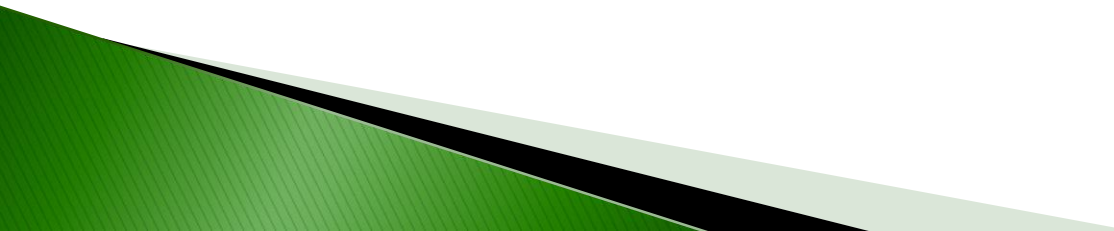
# Facts about the Study

- ▶ Funded as a proof of concept study by NSF-CCLI
  - ▶ Initiated in Fall 2009
  - ▶ Used an existing course, CSC169, and was composed of all freshmen entering CS with weak mathematics scores
  - ▶ Developed laboratory exercises, homework assignments, tests, and just recently powerpoint slides to introduce subject matter
- 

# Related Work

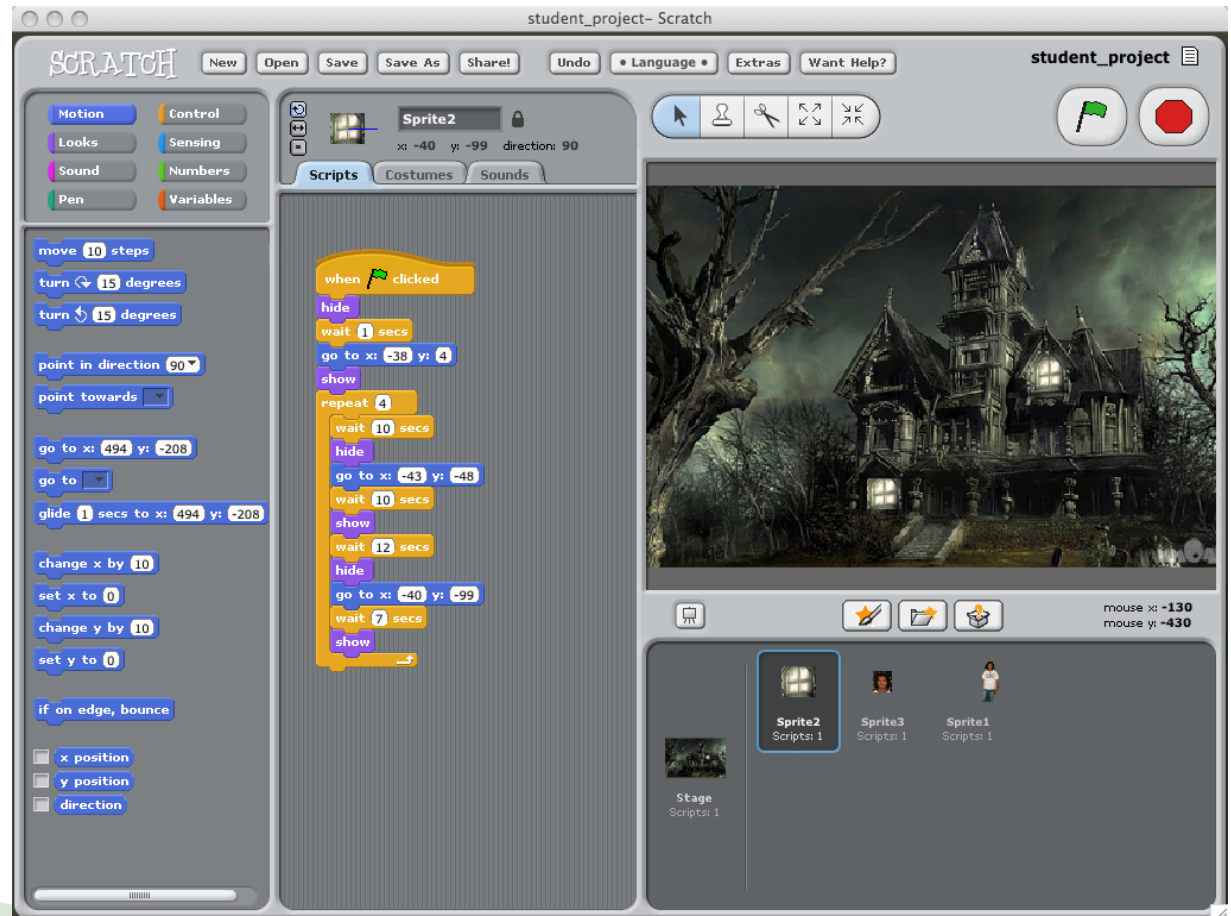
- ▶ Malan and Leitner performed some preliminary testing of Scratch as a precursor to Java at Harvard Summer School
  - Scratch taught for two weeks
  - 76% of the students surveyed at the end of the course indicated that Scratch had a positive effect on them
  - 8% rated the effect as negative
  - 16% indicated that learning Scratch had no effect on their performance in the Java programming course
- ▶ Wolz at The College of New Jersey also taught Scratch at the beginning of a CS1 /CS2 course before transitioning into Java.

# Relate Work (continued)

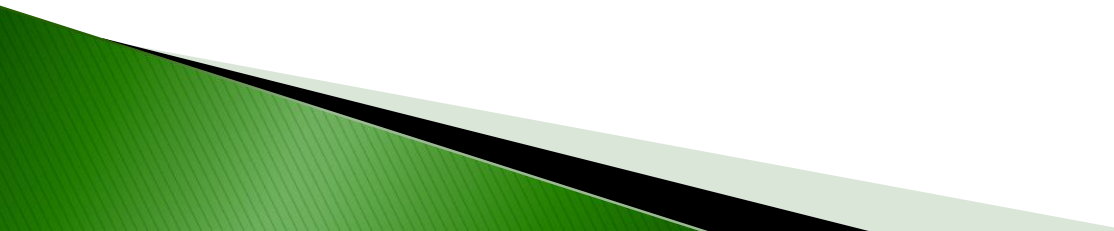
- ▶ Team (Brian Harvey and Dan Garcia) at UC Berkeley developed a CS course with Scratch for students who have no exposure to programming in high school
    - Covers computing concepts
    - Covers diverse topics such as social implications of computer science and computer games
    - 50% of course programming
- 

# Scratch

- ▶ Located at [www.scratch.mit.edu](http://www.scratch.mit.edu)



# Why Scratch

- ▶ Simple and intuitive
    - Blocks that snap together
    - No worry with syntax errors
  - ▶ Media-rich
  - ▶ Support of common programming constructs (e.g. iteration, selection, and sequence)
  - ▶ Implicit notion of objects through sprites and scripts that manipulate sprites
  - ▶ Community
    - Encourages and rewards reuse
    - Has over 1, 393,381 projects to learn by example
- 

# Results to Date–Performance Passing Rates in CS1

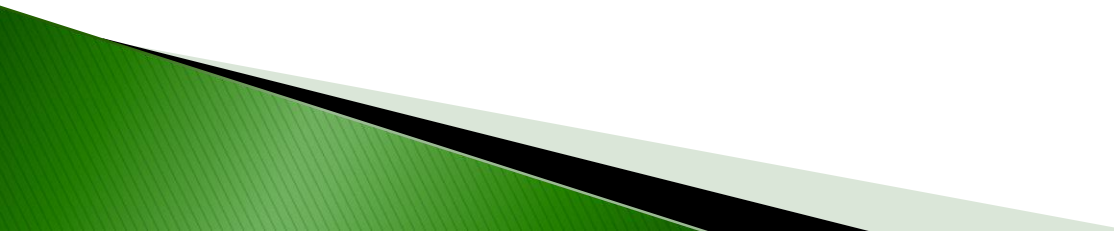
## ▶ Fall 2009

- 33 freshman CS majors took CS0
- 40 freshman CS majors took CS1 (sufficient math)
  - 4 dropped the course
  - **36% Passing Rate**

## ▶ Spring 2010

- 24 CS majors who took CS0 in Fall 2009 took CS1
  - 1 dropped
  - **74% Passing Rate**

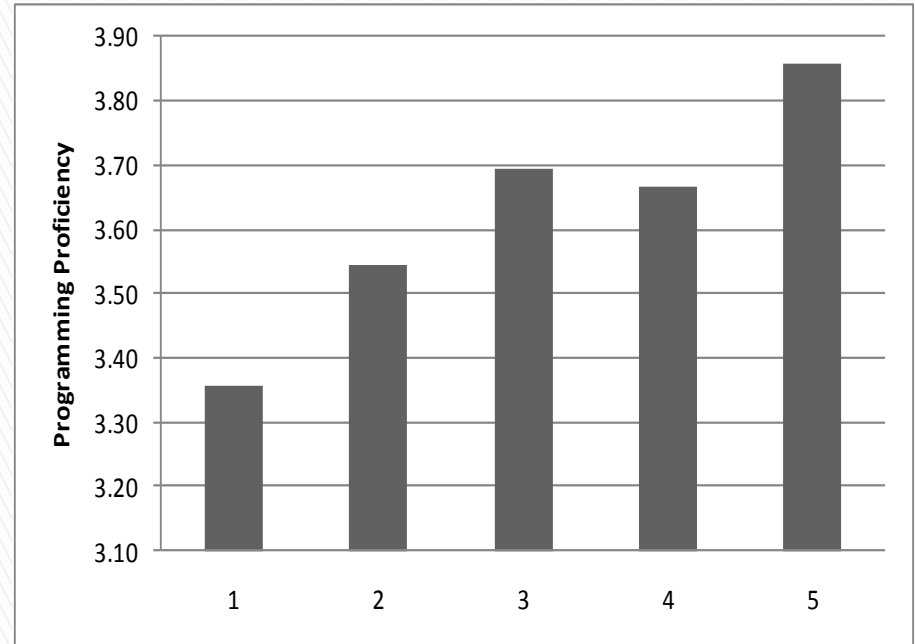
# Results to Date– Retention

- ▶ Measured by students who return the next year to take CS courses
  - ▶ Target Group: **57%**  
(Fall 2009–Fall 2010)
  - ▶ Target Group without intervention: **33%**  
(Fall 2008–Fall 2009)
- 

# Programming Proficiency

- I can't start a programming assignment without assistance
- I can begin a programming assignment but need assistance identify problems
- I am sometimes able to independently identify problems but always need assistance to find solutions
- I am able to independently identify problems in my programs but sometimes need assistance to find solutions
- I am able to independently identify problems in my programs and find solutions on my own

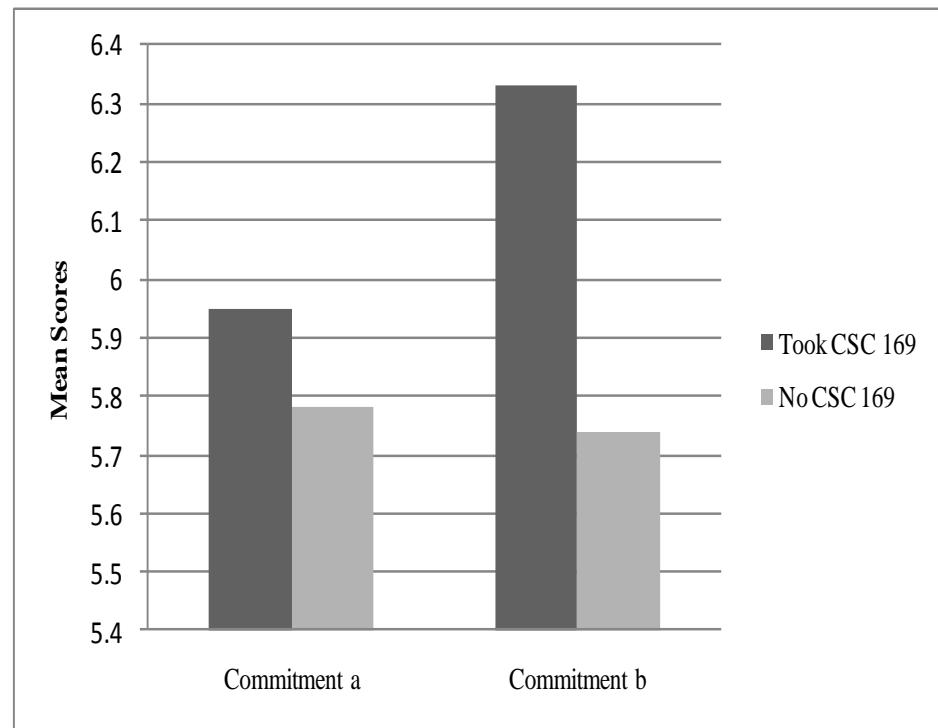
Perceived Proficiency  
using Likert Scale



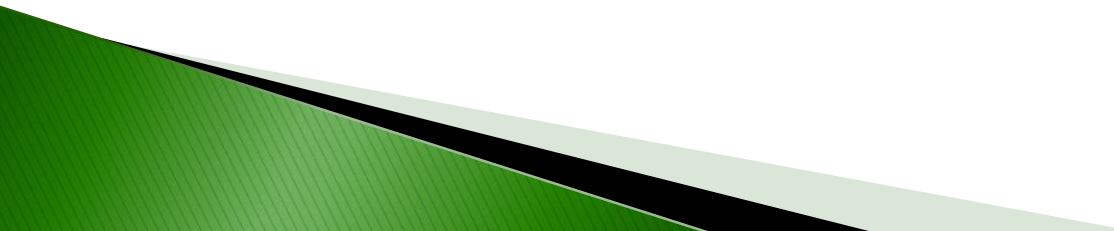
Change in Mean Programming  
Proficiency over Time

# Results to Date – Attitudes

- ▶ Commitment for CSC169 Students Compared to Non-CSC169 Students in CS1 (Spring 2010)



# Observations – Why Successful

- ▶ Time provided for freshmen to mature
  - ▶ A “I can do” attitude developed by students that enables them to struggle through the difficult times
  - ▶ Control constructs of programming taught in CS0 and then revisited in CS1
- 

# Anecdotal Data from Focus Group October 2010

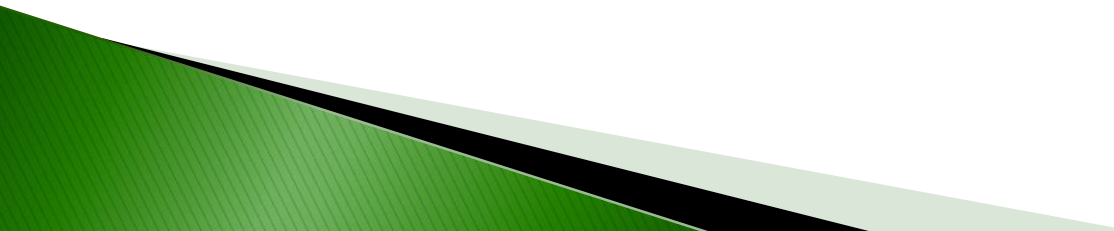
- ▶ Question: In what ways has learning how to program in Scratch helped you in your programming classes?
- ▶ Responses
  - Using loops, it helps in C++.
  - Helps to learn how to trouble problems
  - Gave us a programming mindset, doing things step-by-step
  - Good foundation to start with
  - Teaches you how to plan instead of just jumping into it
  - Steps and logic are similar in Scratch and C++

# CS0 Scratch-based Course

# Course Details

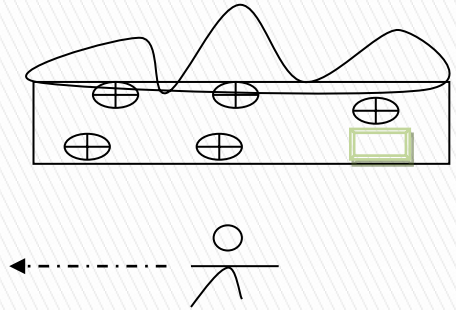
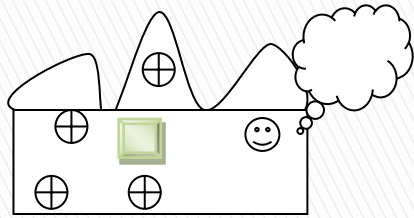
- ▶ **Laboratory based**
  - PowerPoint presentation to introduce subject matter
  - One or more laboratory exercises on the new subject
- ▶ **Midterm and final exams**
  - Written exam with sample code to test understanding of programming concepts
  - Programming exam where students complete a program within an hour
- ▶ **Final Project**
  - Includes storyboard and implemented Scratch program

# Goals of the Course

- ▶ Be able to design and implement complex multimedia Scratch projects
  - ▶ Use and understand basic programming constructs including sequence, iteration, and selection
  - ▶ Use and understand more complex programming constructs (variables, arrays, simple objects, functions, event-driven programming, and synchronization)
- 

# Course Outline

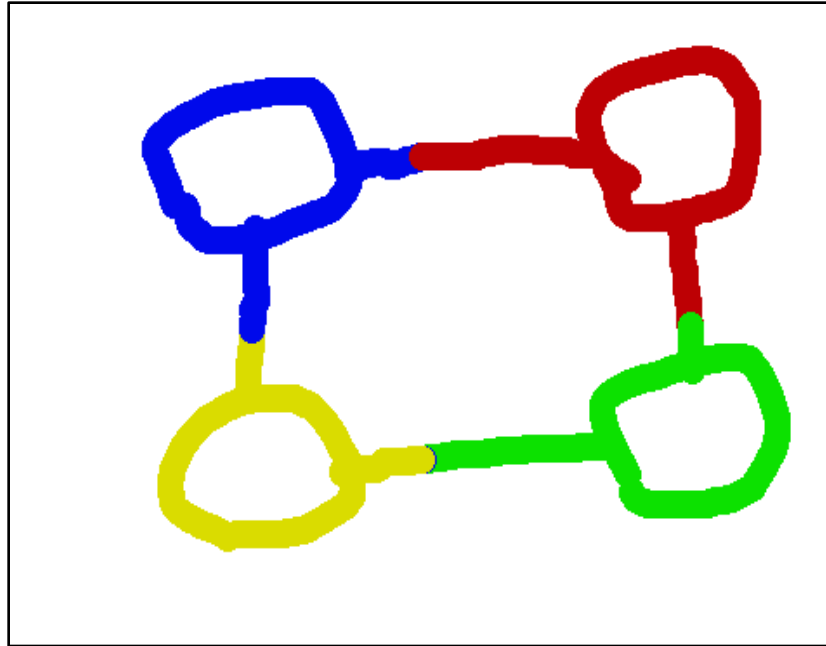
Week	Topic
1	Introduction to Programming; Scratch Overview
2-3	Motion (2D graphics, X-Y coordinate plane)
4	Planning and Designing a project; storyboards
5	Using multimedia (external graphics and sound)
6-7	Iteration, Boolean Expressions, Selection
8-9	Midterm Exam, Broadcasting
10-12	Variables
13-14	List (arrays)
15-16	Final Exam, Project Presentations

	
<p><b>Scene I – Act 1</b></p> <p>Music about “seeing faces” is playing in the background. Jerome is standing in the front yard of his haunted house, practicing his deceitful routing for Halloween. Jerome calls out to his imaginary twin brother, Frome. Jerome moves towards the left side of the stage and hides.</p> <p>Jerome: Let’s play hide and seek!</p>	<p><b>Scene I – Act 2</b></p> <p>While the music continues to play, the lights are blinking on and off. Frome continues to peek out the windows, carefully selecting where he appears.</p> <p>Frome: (thinking) What’s with the lights?</p>

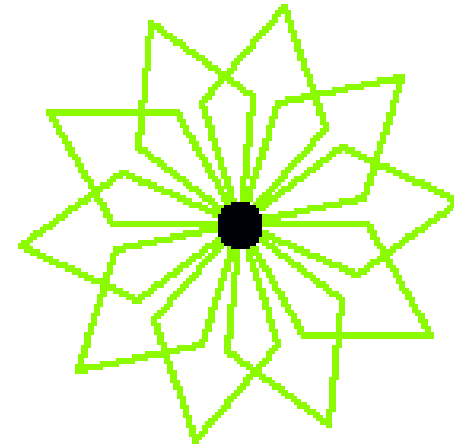
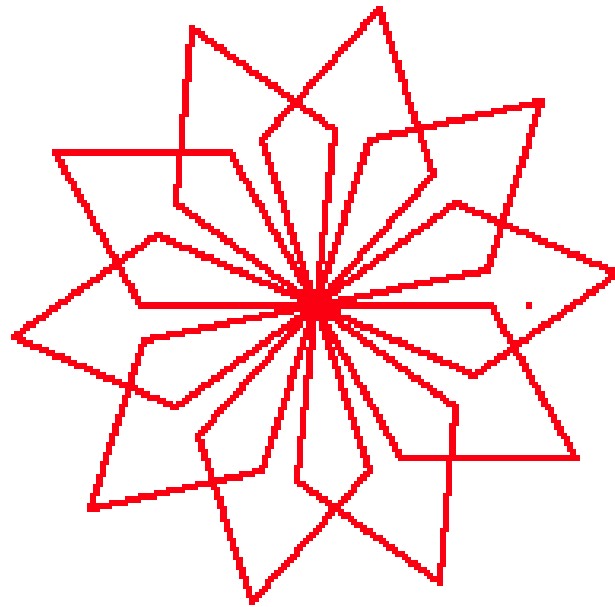
A sample student storyboard for Haunted House project



Scratch Project using Sound



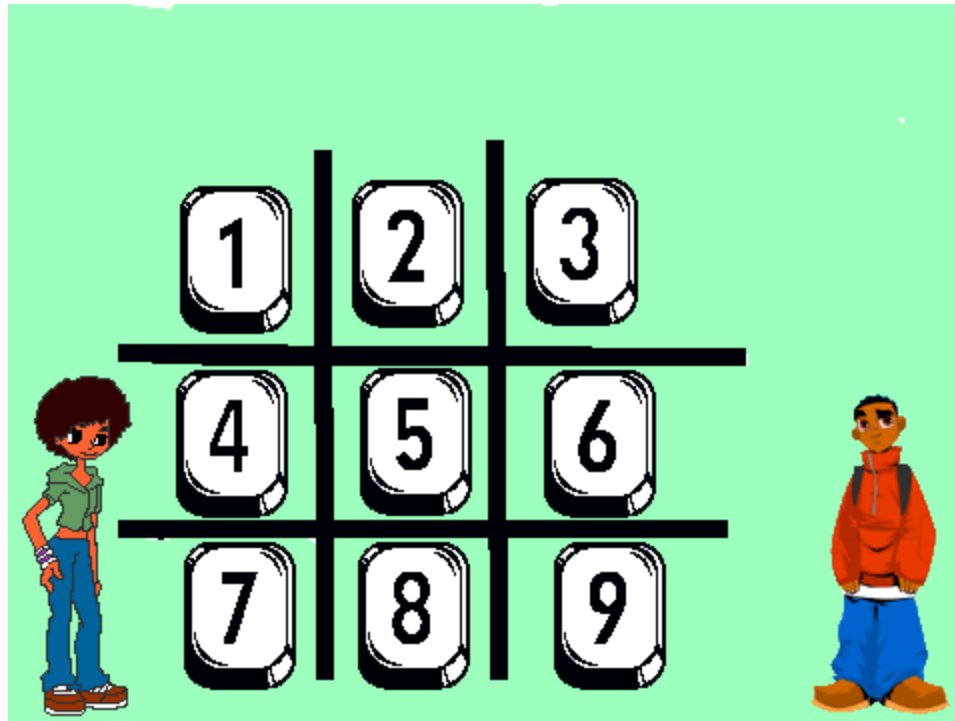
Selection Laboratory Exercise



Variables Laboratory Exercise



Broadcast Laboratory Exercise



Tic-Tac-Toe Game

# Students' Work