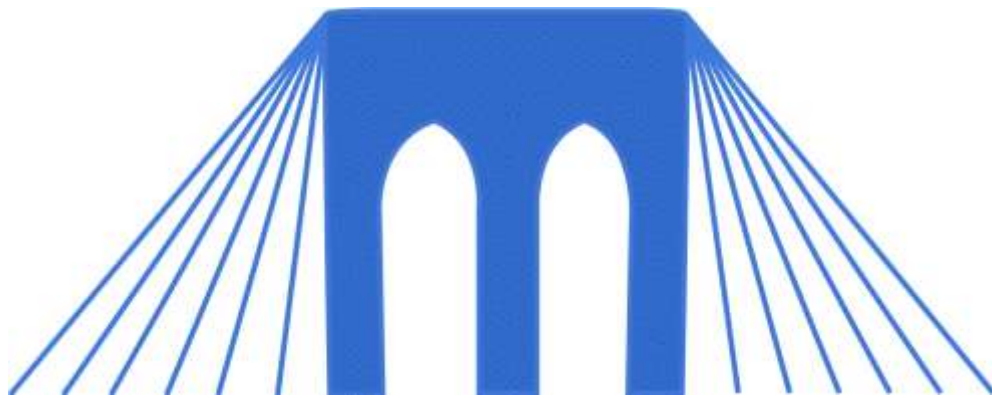


Unit 11:

Game Programming & Game Design



BRIDGES TO
COMPUTING

<http://bridges.brooklyn.cuny.edu>

Topic Descriptions and Objectives

Unit 7: Game Programming & Game Design

Sessions:

- 3 (55min) topic lectures
- 3 (55min) lab sessions
- 4 (55min) project sessions
- Total: 10 hours *

Topics to be addressed:

- Programming in Scratch
- Principles of Game Design
- Game Design methodology
- Game State & Game Mathematics

** Excluding support topic lectures which may be given as part of this unit, or separately as part of a "fundamentals of computer science" unit.*

Topic Description:

After students have been introduced to basic programming terminology as well as issues associated with program design and development, they will be introduced to programming within the Scratch IDE environment. Students will learn some basic terminology used within the field of "game studies" and about the MDA (Methods, Dynamics, Aesthetics) methodology for game development.

By creating several lab games, and their own larger "project game" in Scratch students will gain experience in: programming within the Imperative, Procedural and Object-Oriented paradigms; working with program logic including state; the program development life-cycle; debugging.

Objectives:

The student will be able to:

- Match a list of programming terms and definitions/functions.
- Implement a structured method for designing/creating a game.
- Evaluate a game based on its mechanical, dynamical and aesthetic attributes.
- Program a 2D game within the Scratch environment.

Daily Overview Chart	
Instructional Session	Topic(s)
1 (Taster)	<u>Lecture 1</u> : Learn about programming languages, and the basics of programming in the Scratch environment.
2 (Taster)	<u>Lab 1</u> : Create first Scratch game, a simple 3 sprite “catching” game.
3	<u>Lecture 2</u> : Learn about “game studies” and principles of game design. Explore what makes a game “fun”.
4	<u>Lab 2</u> : Write a review of a game, classify the game by genre and suggest methods for improving the game based on known game design principles.
5	<u>Lecture 3</u> : Explore the concepts of game state and event driven programming. Examine common mathematical problems (collision detection/response) found in 2D games.
6	<u>Lab 3</u> : Create a game that has multiple modes of play (states) that follows a simple state flow diagram.
7	<u>Lab 4</u> : Complete “final project proposal” worksheet.
8-10	Final Project – Create your own game."

Daily Lesson Plans

Instructional Session(s): 1: (Game Programming and Scratch)

Topic Description: This lesson introduces and defines programming languages, and the basics of programming in the Scratch environment..

Objectives:

The student will be able to:

- Match a list of terms and definitions related to programming languages.
- Name the basic components of the Scratch IDE.
- Create the beginning of a simple program in Scratch.

Outline of the Lesson:

- Journal Entry or Q/A session (5 minutes)
- Lecture (30 minutes)
- Scratch IDE session (10 minutes)
 - Takes place half-way through the lecture
- Quiz (10 minutes)

Student Activities:

- Journal Entry or Q/A session
 - Have the students try and answer the following questions:
 - What is a program?
 - What is a programming language and how do they work?
 - What is an IDE (integrated development environment)?
 - What makes a game fun?
 - What skills do you need to have to create a computer game?
- Scratch IDE session
 - Students will stop mid-lecture to explore the Scratch IDE and play **one** simple game in Scratch.
- Quiz
 - Students will match the following terms and definitions.

Teaching/Learning Strategies:

- Journal Entry or Q/A session
 - Have students name as many different programs as they can.
 - Have students name as many different programming languages as they can.
- Lecture
 - Clearly define each of the following terms as they appear in the lecture:

- Program
- Binary
- Syntax
- Semantics
- Variable
- Function
- Library
- IDE
- Green Flag
- Blocks
- Scripts
- Paradigm
- Imperative Programming
- Sequence
- Selection
- Repetition
- Procedural Programming
- Object-Oriented Programming
- Properties
- Sprites Scratch IDE Session
 - Emphasize:
 - Although Scratch is programming, it is not used in industry.
 - Every character in Scratch is called a Sprite.
 - Each sprite has it's own scripts, and may have it's own variables.
 - You can right click any block and select help to get more information.
 - How to change the language in Scratch (for your English Learners)
 - How to go to full screen mode and back
 - How to switch back and forth between sprites by clicking on them
 - X and Y coordinates on the screen are shown in bottom right below the stage
 - How to save in the proper location (the default is to save in the Scratch Projects folder (C:\\Program Files\\Scratch\\Projects))
- Quiz:
 - Most of the terms are listed in the order in which they were introduced.

Resources:

- Game Programming and Scratch : [Lec_1_GameProgScratch.ppt](#)
- Quiz: [GameProgDesign_Session1_Quiz.pdf](#)

Instructional Session(s): 2: A simple Scratch game - (Egg Catcher)

Topic Description:

Objectives:

The student will be able to:

- [objective]

Outline of the Lesson:

- [opening exercise] (5 minutes)
- [lecture or lab name] (40 minutes)
- [closing exercise] (10 minutes)

Student Activities:

- [journal entry]
- [lab exercise]
- [quiz]
- [homework]

Teaching/Learning Strategies:

- [lesson component or activity from above]
 - [description]
 - [emphasis topics]

Resources:

- [lecture name]: [link to lecture]
- [lab name]: [link to lab]
- [handout/quiz/etc]: [link]

Activities

[printouts of all lectures & labs, as well as instructions for any student activities]

Final Project

Description: The purpose of the project is to create a GAME. Whether or not your game includes a NARRATIVE or is presented as an INTERACTIVE STORY is entirely up to you.

Instructions: This project has two parts:

- 1) A Game Proposal Worksheet (GD_Proj_Worksheet.pdf)
 - a) This worksheet should be completed before you start programming your game.
 - b) It is OK if your finished game is different than your original proposal.
- 2) A complete and playable game written in Scratch.
 - a) You may use one of the templates shown in class as a starting point.
 - b) You may work together in groups of up to 4.

Suggestions:

- 1) Use iterative development:
 - a) Get a very very simple version of the game up and working as soon as possible.
 - b) Start by creating the object that the player will interact with or control and get its functionality complete.
 - c) Don't worry about getting the graphics, or sound right at the start. Use "placeholder" graphics if necessary.
 - d) Then add any extra objects that are required and get another working version of the game complete.
 - e) Then work on getting every aspect of the functionality of the game correct in another working version of the game.
 - f) Then go back and add sounds, improved graphics, instructions, credits and all of the other aspects of the game application.
- 2) Be creative!
- 3) Impress me!
- 4) Impress your classmates!
- 5) Have fun with this!
- 6) Consider the fact that the idea you come up with could be the next great game idea!

Final Project Suggested Rubric

Group Members Names: (up to 4)

Do you have?	Points Possible	Yes	No	Points Earned
Worksheet (25 Points)				
Page 1 completed	5			
Page 2 completed	5			
Page 2 completed	5			
Page 2 completed	5			
Page 2 completed	5			
Game (75 Points)				
Is the game suitable for all ages?	10			
Game has "Title" screen	5			
Game has "Instruction" screen.	5			
Game has multiple levels or modes of play.	10			
At least one object (sprite) is controlled by player.	5			
At least one object (sprite) moves independently.	5			
At least one instance of object interaction (this could include objects that respond to mouse clicks, or it could be two objects that interact with each other).	10			
Game has clear victory/loss conditions and/or scoring.	10			
Player can interact with game to a successful conclusion.	10			
Game runs without errors.	5			