

Curriculum Map

Unit 0: Beginnings

Lesson	Objectives	Lab
0.1: The First Day	Identify the class they are taking. List the high-level goals of the course. Describe classroom procedures, rules, and norms.	N/A
0.2: Algorithms	Define "algorithm." Construct algorithms for performing simple tasks.	N/A
0.3: Programming Languages	Complete levels in the game LightBot 2.0. Complete small programs in SNAP with guidance. Explain why computer programs are written in specialized languages.	N/A
0.4: Snap Self Portrait	Create a simple "program" in SNAP to describe themselves	Getting to Know You

Unit 1: SNAP Basics

Lesson	Objectives	Lab
1.1.1: Welcome to SNAP	Define and identify "blocks," "scripts," "sprites," and "the stage" in SNAP. Write simple SNAP programs. Describe what simple SNAP programs do without executing the code.	Welcome to SNAP!

1.1.2: Welcome to SNAP (Day Two)	Continued lab	Welcome to SNAP!
1.2: Building Blocks	Name the categories of blocks in SNAP and describe what the blocks in each category do. Describe the function of several common SNAP blocks. Be able to use common blocks to build simple SNAP programs.	SNAP Scavenger Hunt
1.3: Drawing Shapes	Construct simple algorithms to draw shapes. Convert algorithms into SNAP programs.	Squares and Triangles and Stars, Oh My!
1.4: Animation	Animate SNAP sprites using costume changes and movement. Trigger action in other sprites using broadcasts.	Sprites in Action
1.5: Nursery Rhyme Project	Apply basic programming and SNAP skills to create an animated movie, play, nursery rhyme, or other scene. Practice good debugging skills to correct issues as they arise while programming	Project 1: Animated Nursery Rhyme
1.6: Project Day 1	Project Work	Project 1: Animated Nursery Rhyme
1.7: Project Day 2	Project Work	Project 1: Animated Nursery Rhyme
1.8: Project Day 3	Project Work	Project 1: Animated Nursery Rhyme

1.9: Project Day 4	Project Work	Project 1: Animated Nursery Rhyme
1.10: Culture Day	Connect CS Unit topics with current events	(see Culture Day LPs)

Unit 2: Loop-de-Loop

Lesson	Objectives	Lab
2.1: Loops	Define "loop" in a programming context. Explain why loops are useful. Implement simple repeat and forever loops in SNAP. Utilize loops to reduce redundancy in code.	Squares and Triangles Redux
2.2: Nested Loops	Use nested loops to solve programming problems.	Another Brick in the Wall
2.3: Inputs and Conditionals	Ask for and receive user input in a SNAP program. Use simple conditional (if and if-else) blocks to alter control flow in a SNAP program.	What Shape Is That?
2.4: Variables	Use variables to track values throughout a program.	Guessing Game
2.5: Boole in the House	Define and identify Boolean expressions and operators. Evaluate Boolean expressions. Utilize Boolean operators (and/or/not) to create compound conditions.	Triangle of All Kinds

2.6: Combining Loops and Conditionals	Combine loops with conditionals to create models with repeated but conditional behavior.	What Goes Up...
2.7: Pong Project	Implement a well-written version of Pong. Practice good style and conventions to create readable and maintainable code.	Project 2: Pong
2.7: Project Day 1	Project Work	Project 2: Pong
2.8: Project Day 2	Project Work	Project 2: Pong
2.9: Project Day 3	Project Work	Project 2: Pong
2.10: Project Day 4	Project Work	Project 2: Pong
2.11: Project Day 5	Project Work	Project 2: Pong
2.12: Project Day 6	Project Work	Project 2: Pong
2.13: Project Day 7	Project Work	Project 2: Pong
2.14: Project Day 8	Project Work	Project 2: Pong
2.15: Culture Day	Connect CS Unit topics with current events	(see Culture Day LPs)
2.16: Culture Day	Connect CS Unit topics with current events	(see Culture Day LPs)

Unit 3: Variables and Customization

Lesson	Objectives	Lab
3.1: Abstraction and Friends	Define abstraction, detail removal, generalization, and procedural decomposition in a computer science context. Describe how utilizing procedural decomposition can improve the readability and maintainability of algorithms and/or code. Recognize opportunities to improve algorithms by abstracting or generalizing parts into subprocedures.	N/A
3.2: Procedures	Build custom command blocks in Snap. Utilize detail removal and generalization to construct blocks that practice abstraction.	Drawing Shapes (Again)
3.3: Customizing I: Arguments	Build custom SNAP blocks that take arguments.	Let Me Check My Calendar
3.4: Customizing II: Reporters and Predicates	Build custom reporter and predicate blocks in SNAP.	If My Calculations Are Correct...
3.5: Platform Game Project	Use loops, variables, and Boolean expressions to implement a Super Mario Bros. style platform game. Practice good debugging skills to correct issues as they arise while programming.	Project 3: Platform Game
3.6: Project Day 1	Project Work	Project 3: Platform Game
3.7: Project Day 2	Project Work	Project 3: Platform Game

3.8: Project Day 3	Project Work	Project 3: Platform Game
3.9: Project Day 4	Project Work	Project 3: Platform Game
3.10: Project Day 5	Project Work	Project 3: Platform Game
3.11: Project Day 6	Project Work	Project 3: Platform Game
3.12: Project Day 7	Project Work	Project 3: Platform Game
3.13: Project Day 8	Project Work	Project 3: Platform Game
3.14: Culture Day	Connect CS Unit topics with current events	(see Culture Day LPs)
3.15: Culture Day	Connect CS Unit topics with current events	(see Culture Day LPs)

Unit 4: Lists

Lesson	Objectives	Lab
4.1: Intro to Lists	Explain the concept of a "list" in a programming context. Identify scenarios in which lists are useful	[none]

4.2: Static Lists	Create static lists in SNAP. Access elements of a list. Add and remove elements from a list.	You Talkin' to Me?
4.3: List Practice I	Traverse a list, accessing each element one at a time. Perform operations combining all elements in a list. Select defined subsets of elements in a list.	Guess Who
4.4: List Practice II	Traverse a list, accessing each element one at a time. Perform operations combining all elements in a list. Select defined subsets of elements in a list.	Number Cruncher
4.5: Sequential Search	Explain the sequential search algorithm. Implement several variations of sequential search.	It's Around Here Somewhere
4.6: Word Guessing Game Project	Use lists to implement a complete version of a word guessing game. Exercise good programming practices to produce code that is not only functional but also elegant and well-written.	Project 4: Hangman
4.7: Project Day 1	Project Work	Project 4: Word Guessing Game
4.8: Project Day 2	Project Work	Project 4: Word Guessing Game
4.9: Project Day 3	Project Work	Project 4: Word Guessing Game
4.10: Project Day 4	Project Work	Project 4: Word Guessing Game

4.11:	Project Day	Project Work	Project 4: Word Guessing Game
5			
4.12:	Project Day	Project Work	Project 4: Word Guessing Game
6			
4.13:	Project Day	Project Work	Project 4: Word Guessing Game
7			
4.14:	Project Day	Project Work	Project 4: Word Guessing Game
8			
4.15:	Project Day	Project Work	Project 4: Word Guessing Game
9			
4.16:	Project Day	Project Work	Project 4: Word Guessing Game
10			
4.17:	Culture Day	Connect CS Unit topics with current events	(see Culture Day LPs)
4.18:	Culture Day	Connect CS Unit topics with current events	(see Culture Day LPs)

Unit 5: Cloning

Lesson	Objectives	Lab
5.1: Intro to Cloning	Explain why prototyping and clones can be useful. Describe how complex goals can be accomplished using cloning.	Connect the Dots
5.2: Cloning Sprites	Demonstrate the difference between sprite and global variables. Explain how cloning and prototyping simplify working with numerous similar sprites in the same program. Create prototype sprites and clones of the prototype sprite. Explain the difference between a "master" sprite and a "clone" sprite.	Lots of Balls
5.3: Communicating With Clones	Pass information to individual clones. [Optional] Describe a race condition that might occur due using global variables and clones. Delete clones when they are no longer needed.	Fewer Balls
5.4: Space Invaders	Use cloning to implement a complete version of "Space Invaders." Exercise good programming practices to produce code that is not only functional but also elegant and well-written.	Project 5: Space Invaders
5.5: Project Day 1	Project Work	Project 5: Space Invaders
5.6: Project Day 2	Project Work	Project 5: Space Invaders
5.7: Project Day 3	Project Work	Project 5: Space Invaders
5.8: Project Day 4	Project Work	Project 5: Space Invaders

5.9: Project Day 5	Project Work	Project 5: Space Invaders
5.10: Project Day 6	Project Work	Project 5: Space Invaders
5.11: Project Day 7	Project Work	Project 5: Space Invaders
5.12: Project Day 8	Project Work	Project 5: Space Invaders
5.13: Project Day 9	Project Work	Project 5: Space Invaders
5.14: Project Day 10	Project Work	Project 5: Space Invaders
5.15: Project Day 11	Project Work	Project 5: Space Invaders
5.16: Project Day 12	Project Work	Project 5: Space Invaders
5.17: Culture Day	Connect CS Unit topics with current events	(see Culture Day LPs)

5.18: Culture Day	Connect CS Unit topics with current events	(see Culture Day LPs)
-------------------	--	-----------------------

Unit 6: Final Project

Lesson	Objectives	Lab
6.1: Design Basics	Identify the key considerations when designing a piece of software. Describe methods for prioritizing features, use cases, and/or scenarios. Explain why design and planning are necessary steps in the software engineering process	Final Project
6.2: Brainstorming and Evaluating	Identify factors to use when choosing between project ideas. Rank a group of proposed project ideas using the identified factors	Final Project
6.3: Spec Writing	Identify the main components of a functional project specification and explain the purpose of each section. Develop a project idea into a full, detailed specification	Final Project
6.4: Building a Plan	Identify the main components of a functional project specification and explain the purpose of each section. Develop a project idea into a full, detailed specification.	Final Project
6.5: Project Implementation	Use the skills developed throughout the course to implement a medium- to large-scale software project. Realistically evaluate progress during software development and identify when cuts are necessary. Prioritize features and scenarios and choose which should be eliminated or modified if/when resources and/or time become limited.	Final Project
6.6: Project Day 1	Project Work	Final Project

6.7: Project Day 2	Project Work	Final Project
6.8: Project Day 3	Project Work	Final Project
6.9: Project Day 4	Project Work	Final Project
6.10: Project Day 5	Project Work	Final Project
6.11: Project Day 6	Project Work	Final Project
6.12: Project Day 7	Project Work	Final Project
6.13: Project Day 8	Project Work	Final Project
6.14: Project Day 9	Project Work	Final Project
6.15: Project Day 10	Project Work	Final Project
6.16: Culture Day	Connect CS Unit topics with current events	(see Culture Day LPs)