THE PALIMPSEST PROGRAM

shifting the culture of computing

noun pa·limp·sest \ ˈpa-ləm(p)-ˌsest , pə-ˈlim(p)- \

:an object or place that reflects its own history

"To close gender gaps in participation in computer science, engineering, and physics, cultures of these fields should signal equally to women and men that they belong and can achieve success in them."

-"Why Are Some STEM Fields More Gender Balanced Than Others?" Sapna Cheryan and Sianna A. Ziegler, University of Washington

PROGRAMMING CLASS STRUCTURE

1. Mindfulness reading material and exercise(s)
2. Students play working exercise; ask questions; record questions; take turns
3. Students examine ‘broken’ exercise; identify what’s different/broken; ask questions
4. Enter the exploration loop:
    a. Students form questions
    b. Students vote on a mini lecture to address questions
    c. Instructors deliver mini lecture
    d. Students hypothesize, implement, and test solutions
    e. Repeat
5. Instructors facilitate a class discussion about observations and outcomes
6. Students save their work and receive printouts:
    a. Walk-through of lesson with extension challenges
    b. Link to lectures of the day
    c. Links to online programming tools

PREPARATION

There are six programming tutorials. Each has:

1. A working version of the tutorial, with classes labelled: class.cs (the working script)
2. A broken version of the tutorial, with classes labelled: class_m.cs (m for "modified")
3. A written walk-through with extension challenges

MINI LECTURE OUTLINES

UNIT ONE

Tutorial: Unity scene "BigBang"

- BREAK BY In "Create Prefab" on "Trigger," give Z = 5 to item position (make sure it's visible)
- BREAK BY In "Trigger," uncheck Trigger

3D SPACE (10min)

- 0d, 1d, 2d, 3d (5 minutes)
- Labels vs relationships
- Tactile Ex: pipe cleaners point, line, plane, space
- Physical Ex: yoga in 0, 1, 2, and 3 dimensions

VECTORS & TRANSFORMATION

- Point vs a vector
- Moving in 3d and on a coordinate system
- Dot Product (direction)
- Transformation matrix
- Unity Helix Ex: Transform scale, rotate, translate. Parent a child. Explore object space vs world space.

HARDWARE

- Inputs
- Processor
- Outputs
- Binary
- Base 10; Base 2
- Tactile Ex: Binary addition

PROGRAMMING LANGUAGES

- Labels vs relationships
- High Level
- Low-level
- Tactile Ex: Pseudocode

UNIT TWO

Tutorial: Unity Scene "PushPull"

- BREAK BY PushPull.cs line 66: swap order

DISTANCE FORMULA

- Distance formula
- Tactile Ex: Physical proof of pythagorean theorem

BASICS

- Assignment
- Comments
- Braces
- Code Block
- Unity Ex: Answer prompts from exercise slide

PROGRAM DESIGN

- Unsolvable problems and constraints
- Define the problem
- Algorithm is a method with a series of steps, like a recipe
- Writing Ex: How do algorithms impact your daily life?

DATA & VARIABLES
- Data types: int, float, bool
- Data in memory
- Variables
- ASCII
- Tactile Ex: Binary program with a lightswitch

UNIT THREE

Tutorials: Unity Scene "Collectable"
- BREAK BY omitting omitting contents of CollectableController.CreateAllCollectables(Vec3 pos)

OPERATIONS
- Arithmetic
- Logical
- Operator precedence
- Boolean type
- If statement, if else
- Conditional Statements
- Tactile Ex: Use media of choice to illustrate patterns of if/else statements, and nested if/else statements. Draw all possible paths. Would they compile? What could you use them for?

LOOPS
- purpose
- types
- application
- common errors
- Update()
- What a frame is
- Unity Ex: frame rate change
- Physical Ex: In teams of 2, choreograph a dance using loops. Write out your dance in as few lines of pseudocode as possible. Run 4 frames of dance moves.

ERRORS
- Syntax and Vocabulary
- Runtime and Logical Error
- Crash
- Unity Ex: delete a semi-colon; delete a bracket; delete variable declaration; empty a field in Editor

GAME ENGINES
- Examples: Unreal, Cryengine, custom engine, Quake, Source, and you could make your own engine! All it does is take care of... physics simulation, rendering, collisions, libraries, plugins... what else...
- Activate / Deactivate
- Instantiate / Destroy
- Collision & triggers
- Raycasting - collision without proximity
- Unity Ex: Use UnityEngine class Input.mouse and Input.key to provide alternate ways to change the tutorial without going into VR. Why might this be helpful?

UNIT FOUR

Tutorial: Unity scene "Drawing"

- BREAK BY changing public variable in code
- OPTIMIZE BY overloading CreateDraw to create the added param int passed in
- OPTIMIZE BY changing to an array rather than a list

METHODS

- Definition
- Tactile Ex: pseudocode a character using behaviors (methods) and properties (variables) (slide 4)
- Declaration: mod, return, params
- Naming
- Return type
- Overloading
- Scope
- Unity Ex: create a method
- Physical Ex: In groups of 2, play computer and programmer. Pseudocode a class with 3 methods. The person playing the computer chooses where a variable is defined - in one of the methods or the class. The computer "reads" the program in 3 frames. The programmer has 3 chances to "breakpoint" and ask the value of the valuable. At the end, the programmer must guess where the variable is defined. Then switch roles.

DATA STRUCTURES - anything more complex than a primitive type

- Lists
- Arrays
  - Vector3
- String
- Object
- Struct
- Queue
- Stack
- Tactile Ex: In groups of 4, draw a memory bank. Store 4 objects in an array (side by side). Store 4 objects in a list (all over the place). Put a processor somewhere. Draw one set of lines from the processor to the array, and another set of lines from the processor to the list. Which takes longer? Why might you use one method over the other?

DEBUGGING

- Investigate what something means on your own: Unity reference manual, Stack overflow
- Intellisense
- Use VS to more deeply investigate code
- *Look at the assembly (first figure out how)
- Unity Ex: Use the debugger with Intellisense. Make a breakpoint with F9, Attach to Unity and Play, observe and record the difference between F5, F10, and F11. How do you "watch" a variable? How do you see the values of any variable? When does a variable value change? Why? How long is one frame taking right now? Who's controlling the frame rate?

UNIT FIVE

Tutorial: Unity Scene "Mixer"

- BREAK BY omitting the raycast offset
- BREAK BY making base class modifiers private

OBJECT ORIENTED PROGRAMMING

- Classes
- Data Members
- Method Members

- "New"
- Assignment
- Unity Ex: EXERCISE: Access Members of a Class from INSIDE and OUTSIDE of the Class. (slide 9)
- Unity Ex: EXERCISE: Construct something

ENCAPSULATION

- Encapsulation and Unity's Game Objects & Components
- Data and Method modifiers
- Principle of least privilege
- Scope
- Class Modifiers
- Physical Ex: Let's play a spy game called, "Encapsulation." Everyone has some data, some functions, and some functions calls. Try to get the most data! Find the people with your functions calls and ask for their data. But be quick! Don't lose all your data in the process. In the bonus round, players find their parent or manager, which has the permission to access the data.

REFERENCING OBJECTS

- Garbage Collection
- "this"
- Pass by Reference vs Pass By Value
- Tactile Ex: Draw the difference between a reference and a value. Where is the original object in memory? If passing by reference, what is passed, the object, a copy of the object, the object's address, or the object copy's address? What about for passing by value? Why pass by reference versus passing by value?

UNIT SIX

Tutorial: Unity Scene "Scene Switch"

- BREAK BY omitting the Scene Manager
- VARIATION: simulate scene switching in one scene
- VARIATION: Make a new child
- VARIATION: Make the mixer behaviors not inherit from the MixerFeature.cs

APPLICATION

- Game concept creation
- Game design document
- Asset list
- Prototyping and priorities
- Written Ex: Create concept document, asset list, set priorities