Student Presentation
What is UML?

- Unified Modeling Language
  - General-purpose notation for software development
  - Standard visualization of a system’s design
  - Allows for easier design and clearer implementation
  - Provides a common model between programmers to follow project standards
UML Design

- Not code based
  - Language agnostic
  - Used beyond code design
- Visual depictions of structures and behaviors
- Diagrams provide standards depending on purpose
Structural UML Diagrams

- Describe static or structural aspects of a system
- Types of structural diagrams:
  - Class
  - Component
  - Composite Structure
  - Deployment
  - Object
  - Package
Class Diagrams

- Describes object-oriented approach to design without any coding
- Components represent classes, objects, and interactions between classes and objects
- Contains symbols to represent object-oriented concepts and code specifications
❖ Class name
❖ Class attributes
❖ Shows class fields and types
❖ Class methods (operations)
❖ Shows how class interacts with data
Inheritance

- Open arrow points to superclass
- Child class assumed to inherit all attributes and methods of parent class
Class Interactions

- Models:
- Dependencies
- Compositions
- Relations
- Many, many more
Consider...

❖ The Spot class has an (x, y) position and a radius as well as methods `display()` and `move()`

❖ How would this look as a UML class diagram?

❖ What would the UML class diagram look like if we create the subclass `ColoredSpot` that also has a field for color?
Behavioral UML Diagrams

- Represent dynamic aspects of a system
- Types of behavioral diagrams:
  - Activity
  - Communication
  - Interaction Overview
  - Sequence
  - State Machine
  - Timing
  - Use Case
Activity Diagrams

- Provides a flow chart for application functionality
- Similar to “wire frames” in design
- Useful for describing dynamically-driven systems
- Sequential, branching, or concurrent
- High level view of the system
Sequence Diagrams

- Tracks interactions between objects and sending/receiving of messages
- Depicts objects in terms of methods and function calls
  - Lifeline shows creation and deletion of objects
- Useful for providing detailed description for implementation
- Low level view of the system
State Machine Diagrams

- Describe the different states of an object throughout its lifetime
  - Focus on single object rather than entire system
- Capture object’s dynamic reactions to events
- High or low level view of object
Using UML Diagrams

- Formal diagrams extremely useful for large group projects
  - Prevents misunderstanding
- Diagrams still useful for small group development
  - Clarifies thoughts
  - Unifies systems goals
- Useful at high and low levels of development
- Online programs for drawing (but hand drawn works!)
Question

- What are some behavioral diagram types and their properties?
Online UML Diagram Tools

- Lucid Chart <https://www.lucidchart.com/>
- Gliffy <https://www.gliffy.com/uses/uml-software/>
Hands-on: Using UML Diagrams

❖ Today’s activities:

1. Get together with your group and think about the animation you’d like to create

2. Create a class diagram of an object you might implement for that animation

3. Create either an activity, sequence, or state diagram to capture the flow of the animation and/or user interactions