

Objectives

- Team Final Project
- Teamwork
- Design and Analysis

Project Deliverables Timeline

Worth 20% of your course grade

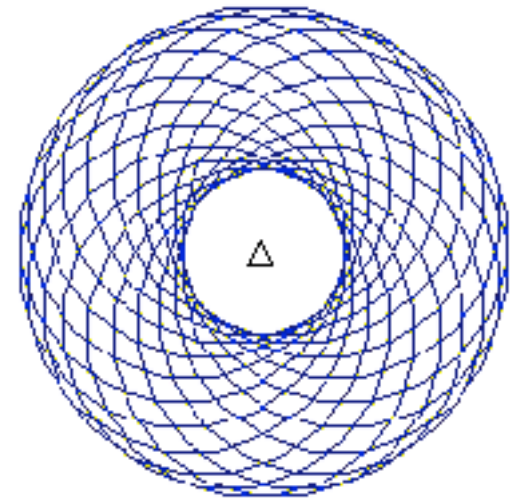
Deliverable	Who	Weight	Due Date
Preparation	Individual	8%	Fri, 11/18
Preliminary Implementation	Team	20%	Wed, 11/30
Intermediate Implementation, Demo	Team	22%	Wed, 12/7
Final Implementation, Documentation	Team	35%	You decide → latest 12/15, midnight
Analysis	Individual	15%	12/16, 5 p.m.

Teams Work Best When They are Interdependent

- Are you allowing your team to truly be interdependent?
- Who might be you be ignoring?
- Who might be allowing themselves to feel inadequate?
- How do you show appreciation for each other and yourself?

SLogo Project Overview

- Goal: Create an IDE for simplified version of Logo
- Logo: programming language designed to teach children to program
 - Low floor, high ceiling



ANALYSIS & DESIGN: FORMALIZED

Analysis Phase

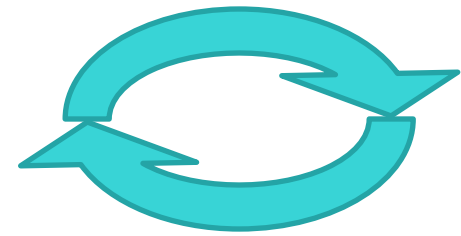


“Dohickey”

- Create an abstract model in client’s vocabulary
- Strategy:
 1. Identify classes that model (shape) system as set of abstractions
 2. Determine each class’s purpose or main responsibility
 - member functions
 - data members
 3. Determine helper classes for each
 - Help complete responsibilities

Analysis Phase Discussion

- Expect to **iterate**
 - Won't find all classes at first
 - Especially helpers
 - Won't know all responsibilities
- Uncertainty in problem statement
 - May be concerns that need to be settled
 - Try to understand requested software system at level of those requesting software
- Rarely one true correct best design



Identification of Classes

- Potentially model the system
- Usually **nouns** from problem description or from domain knowledge
- Model real world whenever possible
 - More understandable software
 - Helps during maintenance when someone unfamiliar with system must update/fix code

Identifying Responsibilities

- Responsibilities convey purpose of class, its role in system
- Questions to Ask:
 - What are the other responsibilities needed to model the solution?
 - Which class should take on this particular responsibility?
 - What classes help another class fulfill its responsibility?

Have You Modeled Everything?

- Strategy: Role playing
- Act as different classes: can you do everything you want in various scenarios?
 - Fill in missing classes, responsibilities
 - Methods: parameters, what returned
 - Restructure as necessary
 - No code yet so not actually refactoring
- Example **use cases**/scenarios:
 - User borrows a video and returns it two days late
 - User tries to borrow book that is already checked out

Definition of Use Case?

- Description of steps or actions between a user and a software system towards some goal

SLogo Functionality Overview

- User enters SLogo commands
 - Commands defined by specification
- Interpreted and batch modes for entering commands
 - User can save files of commands
- Have turtle execute the commands
 - Or descriptive error messages
- Many possible extensions