Massachusetts Institute of Technology

Architecture of Complex Systems

Week 1: Systems Thinking

Project Portfolio

Name

Tomas Mawyin

Massachusetts Institute of Technology

Instructions

Before you begin, you should save your Project Portfolio on your local drive. We recommend the following format:

Lastname_Firstname_Course1_Week1

Please note: You will not be able to re-download your file after submission; therefore, please keep this file in a central location for future reference.

While you may be working with a group, the project deliverable is an **individual submission**. A scoring rubric can be downloaded from the course in the Resources/Downloads tab on the top navigation.

You will be self-assessing your work. If you have any questions, feel free to start a thread in the Discussion Forum. Although work is strictly individual, sharing ideas and concepts with other students or your team is encouraged.

Note: edX has a 10MB file size limit for document submission. If you have selected large image(s), you may need to resize before submitting, OR you may simply include a web URL for the image in the image location. Be sure to submit your assignment at least one hour before the deadline to provide time for troubleshooting.

Once the deadline passes, you will not be able to upload the document and therefore will not be able to submit and complete the assignment.

Massachusetts Institute of Technology

Week 1 Project

Overview

In this first project activity, you will apply the four tasks of systems thinking to a real-life "system." You will have the option to choose from five different system types, which are shown on the next slide.

We recommend that you do some brief research on each one before choosing; however, if one is particularly relevant to your occupation, feel free to simply select that. You will then follow the steps which appear in the text box to the right to complete this activity.

REQUIRED STEPS:

Step 1: Research and select your project "system."

Step 2: Identify the primary ENTITIES in your system.

Step 3: Identify your system's FORM and FUNCTION.

Step 4: Identify RELATIONSHIPS in your system.

Step 5: Predict your system's EMERGENCE.

Step 6: Develop system DECOMPOSITION.

Step 7: Review and submit your project.

Step 1: SYSTEM DETAILS

The following details pertain to the systems described in the previous slide and provide general guidance. You may find a specific diagram and/or schematic of the system that has a different representation. However, you are expected to work with one of the five systems described while maintaining the same level of complexity as below. For instance, you may *not* choose a half bit adder instead of a full bit adder.

- Balsa wood glider: A beginner-level model hand-launched glider airplane composed of the following: fuselage, rear stabilizer, fin, wings, and front counterbalance weight.

- Crystal radio: Simple early form of a radio receiver circuit composed of the following: variable capacitor, aerial terminal, inductor coil, diode, resistor, earplug output, and a ground terminal.

- Prime number search code: C void function that prints all the prime numbers less than or equal to the number given in variable *n*. The function is composed of: variable *n*, array of length *n* named “primes,” for loop that flags all items in the array as prime numbers, for loop that computes for the prime numbers in the array, for loop that prints out to the prime numbers of the array.

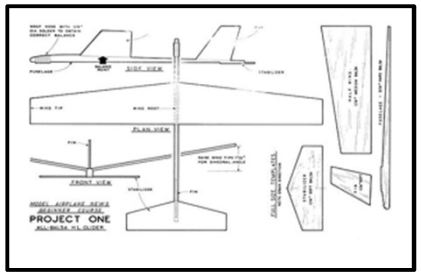
- Simple refracting telescope: Optical telescope based on two lenses. The system is composed of a focal tube, objective lens, eyepiece, eye lens, focus knob, three tripod legs, focal tube mount, and tripod flange.

- One-bit full adder: Digital circuit that adds three one-bit numbers. This system is composed of two XOR gates, two AND gates, and one OR gate.

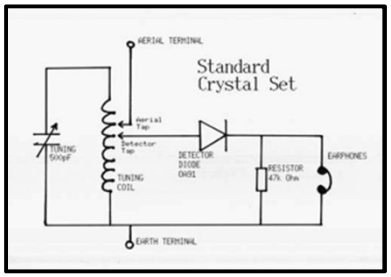
Massachusetts Institute of Technology

Step 1: SELECT YOUR SYSTEM

My system choice: Balsa Wood Glider



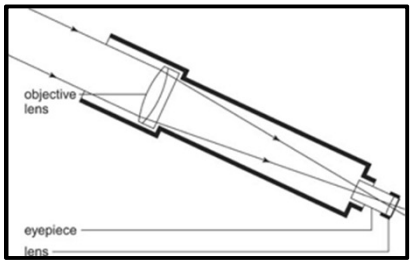
Balsa Wood Glider



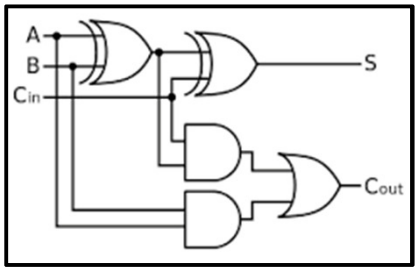
Crystal Radio

```
public void findPrimes(int n){
    int[] primes = new int[n];

    // Flag all numbers as prime initially
    for(int i = 0; i< n; i++)primes[i] = 1;

    // Compute all prime numbers
    for(int i = 2; i <= (int)(Math.sqrt(n)); i++){
        if (primes[i] == 1){
            for(int j = 1; i*j < n; j++)primes[i*j] = 0;
        }
    }

    // Print out all prime numbers less than n
    for(int i = 1; i < n; i++){
        if(primes[i] == 1)System.out.printf("%d ", i);
    }
}
```

Prime Number Search Code



Simple Refracting Telescope



1-bit Adder

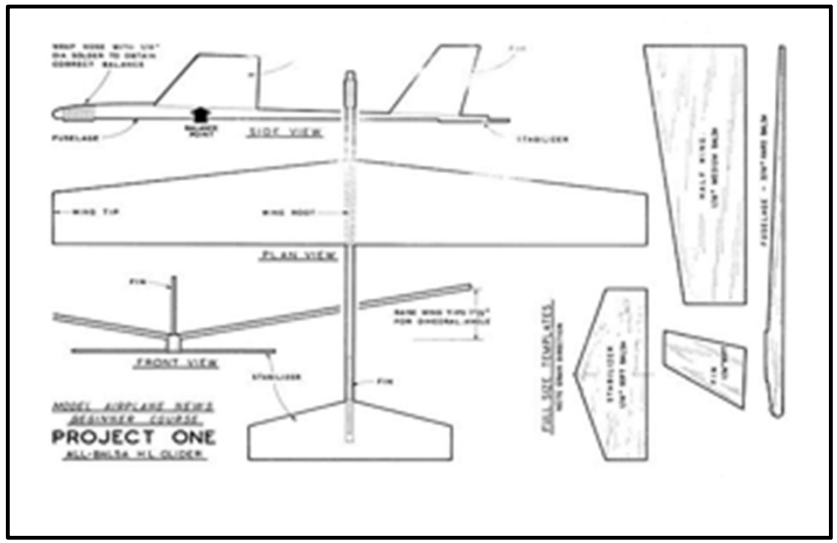
Massachusetts Institute of Technology

Step 2: IDENTIFY SYSTEM FORM & FUNCTION

Insert image you sourced representing your selected system in the box on the left side of the slide below. Then indicate the examples of FORM and FUNCTION that you've identified in the field on the right below.

Note: edX has a 10MB file size limit for document submission. If you have selected large image(s), you may need to resize before submitting, OR you may simply include a web URL for the image in the image location.

System Diagram/Schematic



Primary System FORM:

Balsa Wood Glider

Primary System FUNCTION:

Gliding

Please describe why these elements of your system represent form and function and contextual interrelationship.

As form I choose Balsa Wood Glider because it represents what the system is – what gets built at the end and it's what allows the function of gliding to exist. Gliding or flying is what the system does when hand-launched.

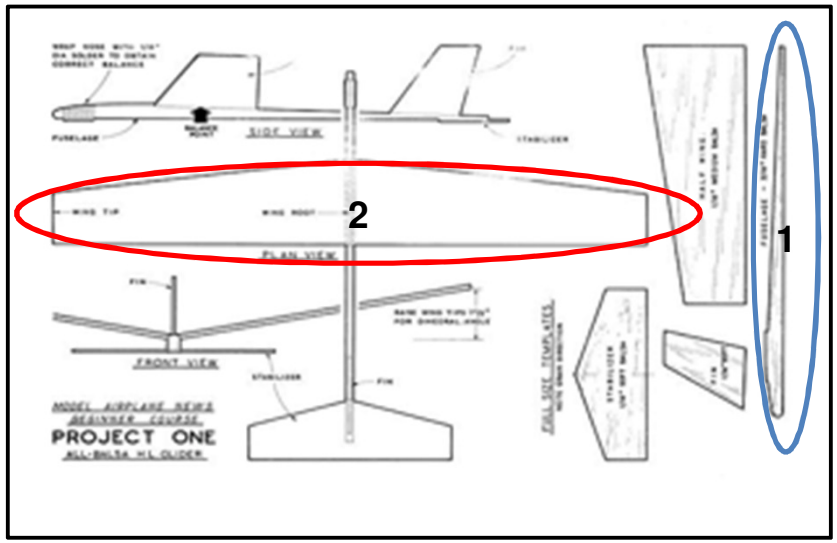
Massachusetts Institute of Technology

Step 3: IDENTIFY SYSTEM ENTITIES

Copy the image you inserted into Slide 6 here. Then highlight or circle the different entities in your system and indicate them along with their respective form and function in the fields at right.

Note: edX has a 10MB file size limit for document submission. If you have selected large image(s), you may need to resize before submitting, OR you may simply include a web URL for the image in the image location.

System Diagram/Schematic



System Entity 1:

Fuselage

Form: Balsa Wood Stick

Function: Support all components

System Entity 2:

Wings

Form: Balsa Wood Wing

Function: Provide lift

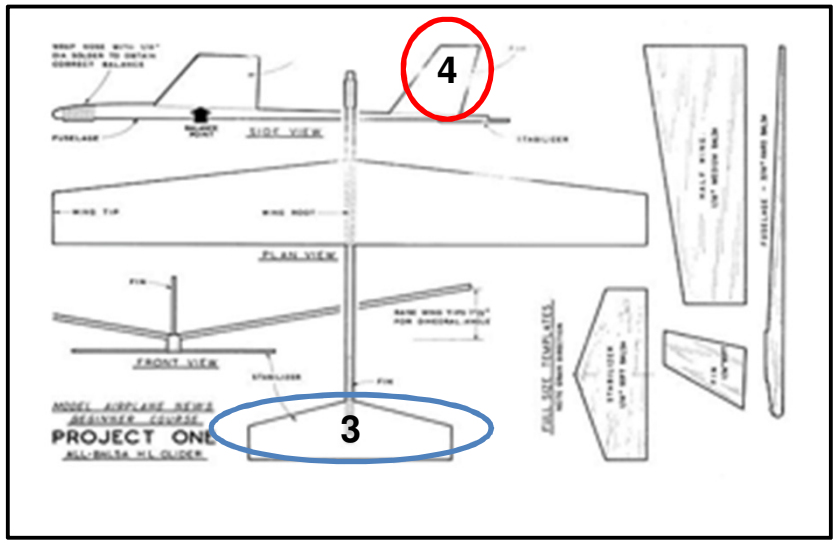
Massachusetts Institute of Technology

Step 3: IDENTIFY SYSTEM ENTITIES

Copy the image you inserted into Slide 6 here. Then highlight or circle the different entities in your system and indicate them along with their respective form and function in the fields at right.

Note: edX has a 10MB file size limit for document submission. If you have selected large image(s), you may need to resize before submitting, OR you may simply include a web URL for the image in the image location.

System Diagram/Schematic



System Entity 3:

Rear Stabilizer

Form: Balsa Wood Stabilizer

Function: Provides stability and control

System Entity 4:

Fin

Form: Balsa Wood Fin

Function: Provide stability and control

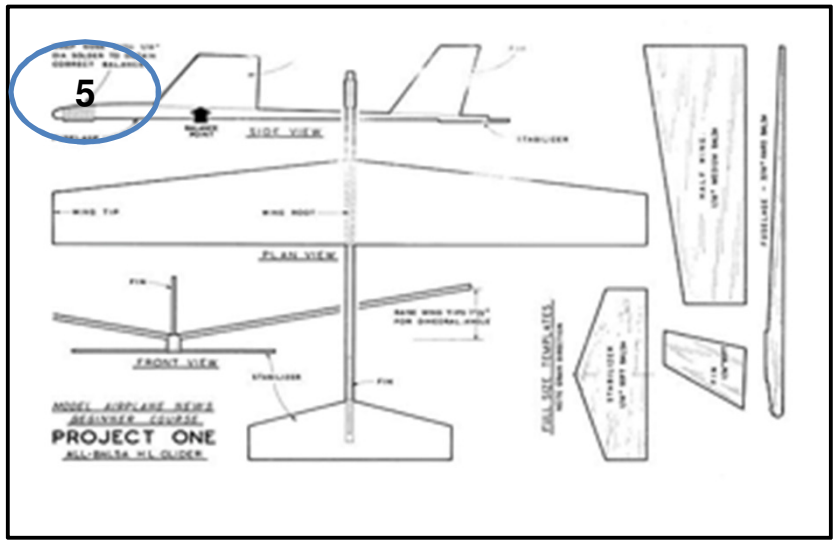
Massachusetts Institute of Technology

Step 3: IDENTIFY SYSTEM ENTITIES

Copy the image you inserted into Slide 6 here. Then highlight or circle the different entities in your system and indicate them along with their respective form and function in the fields at right.

Note: edX has a 10MB file size limit for document submission. If you have selected large image(s), you may need to resize before submitting, OR you may simply include a web URL for the image in the image location.

System Diagram/Schematic



System Entity 5:

Counterbalance weight

Form: Weight

Function: Provides balance

System Entity 6:

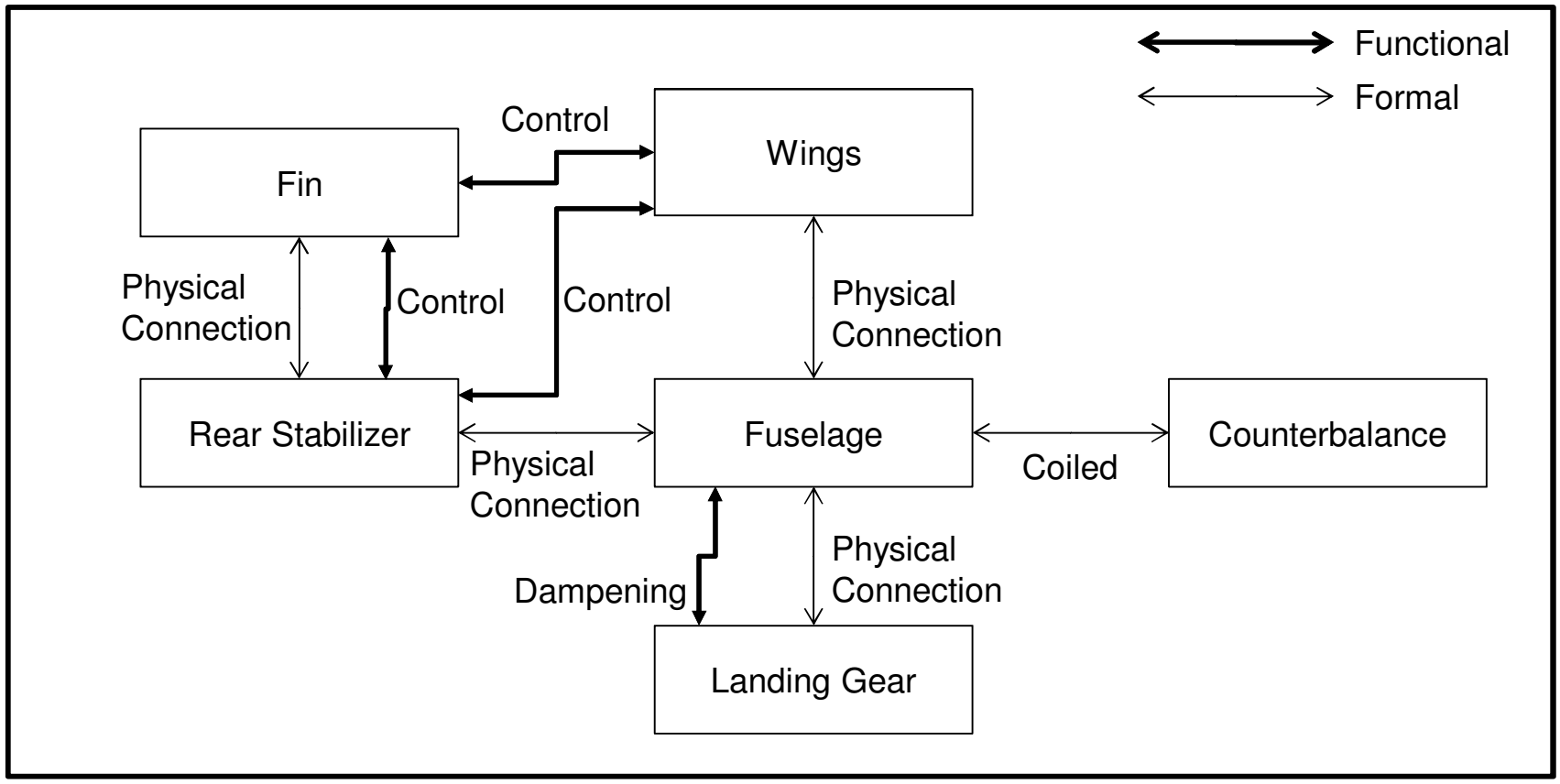
N/A

Form: N/A

Function: N/A

Step 4: IDENTIFY SYSTEM RELATIONSHIPS

Replace the names of Entities 1 and 2 in the diagram below with the first two entities you identified on the previous step. Identify at least 4 more entities in your system. Then, use the connectors to define the relationships between your entities. You may copy the objects you use more than once and delete any you don't use.



Massachusetts Institute of Technology

Step 5: PREDICT SYSTEM EMERGENCE

For this step, you will predict two types of emergence in your system: intended and unintended, including emergence failures. Write a brief description of each emergent behavior of your system, and then explain how it occurred by describing it functional emergence.
You may also upload an image for each emergence type if you prefer. Please remember the file size limit and resize or paste the image URL instead, as needed.

Intended Emergence

Flying (Gliding)

Functional Interaction

Components are properly attached to balance the glider and provide stability when launched in the air.

(optional) **Image**

Insert image here.

Unintended Emergence

Crashing

Functional Interaction

Components are not attached properly which reduces balance and stability causing the glider to crash.

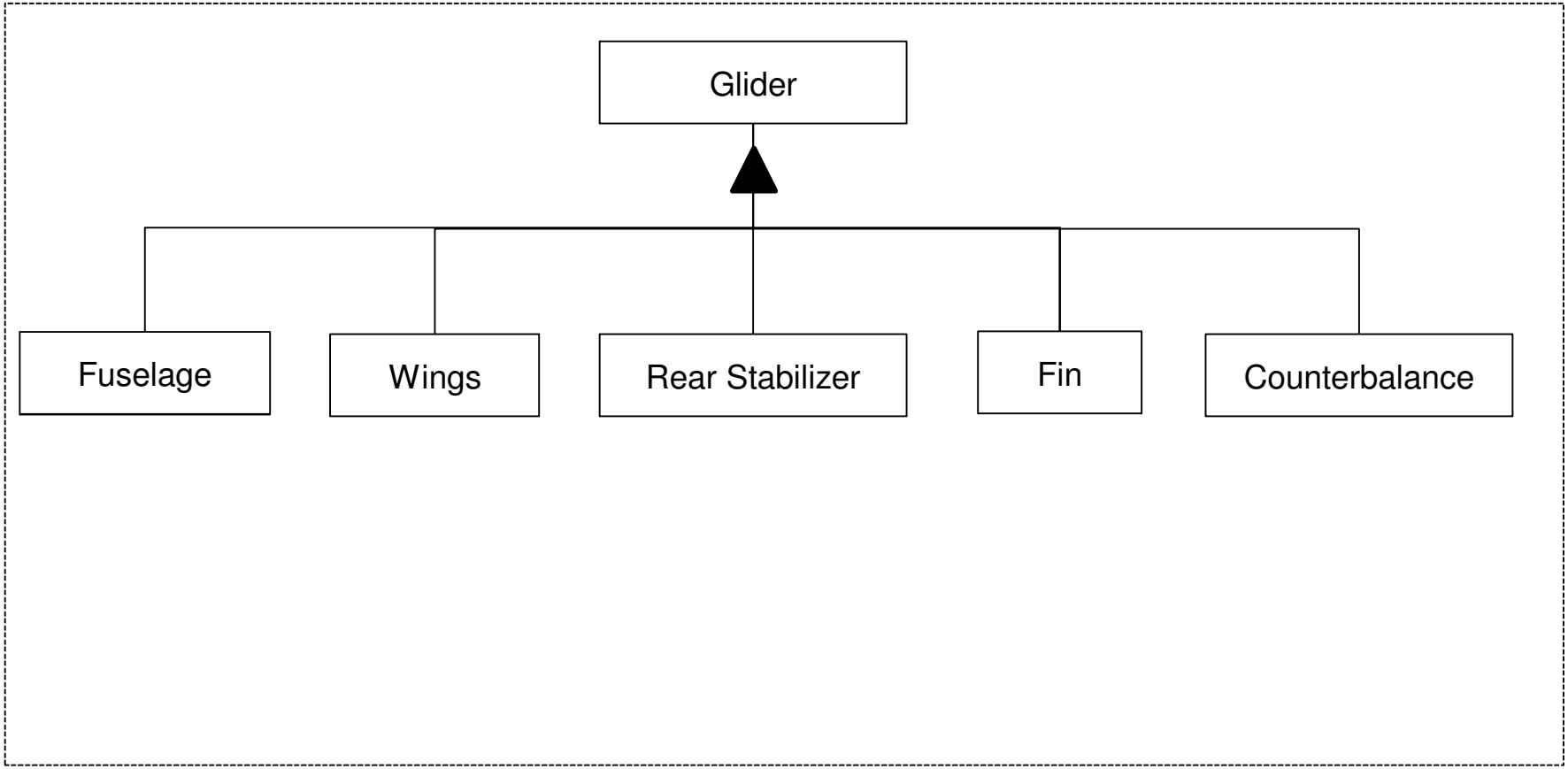
(optional) **Image**

Insert image here.

Massachusetts Institute of Technology

Step 6: DEVELOP SYSTEM DECOMPOSITION

For your last step, develop a Level One system decomposition. Draw a decompositional view of your system that includes Level Zero and Level One.



Massachusetts Institute of Technology

STEP 7: REVIEW & SUBMIT PROJECT

- Submit and self assess your completed Week 1 Project Portfolio file Note: The maximum file size that can be submitted is 10MB.
  - A sample project submission and scoring rubric can be downloaded from the course in the Resources/Downloads tab on the top navigation.
  - Please remember that there are two steps to this assignment: Submission and self assessment. Please be sure to provide enough time to complete both steps.