Facilitation Guide

Design Process Introduction: Animation

# Summary

This facilitator guide is intended to guide the instructor through introducing the design processes, going from the idea stage through a script or storyboard, to an algorithm design or flow chart, to a final programmed animation. It will guide the facilitator through the introduction of the problem solving process in developing Alice programs specifically for animations. This includes the concepts of design thinking, script writing, storyboarding, set design, algorithm development and flow charts.

This should be one of the early lessons in The Alice Project’s curriculum as these skills will be referenced and can be used as a basis for future project-based learning in Alice. This lesson assumes that the participants have learned the basics of the Scene Editor and the Code Editor. While that is the earliest this lesson could be introduced, it could also be be given after more Alice skills have been introduced such as custom procedures or functions. Waiting will increase the complexity of animations possible and possibly extend the time students will want to spend on their projects. This project is the broadest approach to designing and implementing an animation in Alice. The time spent on this project can be shortened for a quick design exercise or it can be extended to a long in-depth creative project depending on the time allotted and/or the exercises assigned. Extensions of this lesson that build on these concepts include lessons on designing interactive worlds and game design at alice.org.

# Learning Objectives

* Define the iterative problem solving process
* Define an animation “problem”
* Identify the key components of a story
* Define a script or storyboard
* Define algorithm design and flowchart
* Translate a story or storyboard into an algorithm design or flowchart
* Translate a story or storyboard into a set design

# Lesson Overview

* Introduction
* Lesson on the Alice Code Editor
* Student Work Session
  + OR Step-by-Step Work Session
  + OR Guided Student Work Session
* Assessment
* Debrief / Students Share Work

# Skills Overview

This project was developed for use with Alice 3. The following Alice 3 skills will be learned through the lesson and additional resources related to the skills can be accessed in the *How To* videos and the [tutorial exercise](https://www.alice.org/resources/exercise-and-project/tutorial-designing-an-animation/) for this lesson. *Optional educational activities can be incorporated based on their relevance to the required steps for the project.*

## Design Basics

Creating a storyboard / script from a problem statement

Identifying object-action pairs in a script or storyboard

Creating an algorithm / flowchart from a storyboard / script

Identifying background information for set design in a storyboard / script

# Vocabulary

**Algorithm** - Set of steps to accomplish a task

**Alice Algorithm** - Set of steps that tells the computer how to animate the virtual environment

**Camera Moves** - Using procedures to move the camera during execution that can be used to create dynamic animations or jump between set locations

**Commenting** - Text in a computer program that is ignored during execution that can be used to plan and document code for human reference

**Flowchart** - A diagram that shows different paths a program will take depending on what data is inputted

**Diamond** - Used in a flowchart for decision and conditional branching

**Parallelogram** - Used in a flowchart for getting inputs

**Rectangle** - Used in a flowchart for processes or actions

**Round Rectangle** - Used in a flowchart to show the start and stop of the program

**Incremental Design** - The process of building small components, testing, refining, and then continuing to add more functionality

**Problem Statement** - The first step in understanding a problem. A concise description of the problem.

**Set Design** - The design of the space that supports a desired narrative.

**Testing** (CS) - Running a program to determine if objectives are being met.

# Prep + Materials

## Classroom Resources

### Computer Access

Each participant should have his or her own computer for the duration of the project. It is also possible to allow pairs of students to work together on a shared computer.

### Presentation + Lecturing

Ideally, you should be able to present the lesson slides in front of the class. Depending on your approach, you may also want to be able to show Alice and be able to demonstrate and guide the class through the exercise. You can also print and distribute these materials if needed.

### Supporting Materials

This lesson requires offline activities that may require pen/pencil and paper. If the students elect to draw storyboards or maps and set design diagrams you may also choose to print out blank storyboard templates. You may want students to have access to the [*How To* videos](https://www.alice.org/resources/lessons/design-process-introduction/) that will assist in the implementation stage of this lesson. This can be achieved by insuring they have access to the Alice.org website and can play the videos or by downloading and making the videos accessible in another way. You may also wish to download, print, and distribute the accompanying Quick Reference Guides that can be found on the tutorial webpage. These materials can be downloaded in .doc format to combine several into one hand out.

### Software Requirements

This lesson requires each computer to have Alice 3 installed and available.

## Time

The lesson is designed to take 45-90 minutes but can easily be extended into a longer project module, depending on:

* The inclusion of the lesson presentation
* Time spent on going over skills training
* Time spent on optional learning activities
* Complexity of exercises or projects assigned
* Time spent debriefing

# Suggested Process

## Introduction

Tell the students that they will be shown how to design an Alice animation. Describe the class activity and discuss the skills they will acquire in the process.

## Designing an Alice Animation Lesson (optional)

For this lesson, you can present the lesson several different ways. One option is to run completely through the supplied slides to give an overview of the problem solving process and how it correlates to the Alice design and development process and concepts. You could also integrate sample exercises into the presentation at the relevant topic moments. The level of detail and time spent on the lesson may be determined by how you plan to facilitate the exercise and the skill level of your students.

## Exercise Facilitation

There are several options for how to allow the participants to design and implement an Alice animation using the outlined design process:

Option 1

With groups that do well independently you may allow them to create the storyboard / algorithm for a project of their own design (in this case the tutorial exercise is the open ended option), or you can give them one (or several) of the provided challenge exercises for more directed projects or for students needing ideas to get started. Additionally, you might want to provide them with the tutorial exercise or slides for this lesson as a guide for the steps they should follow in the design process.

Option 2

Alternatively, you may also break the session up into smaller segments or Modules. For each module you will be demonstrating or discussing each skill in detail before students apply them. This allows for break points to check in with participants. You can approach this in two ways:

* Follow the step-by-step directions for the module on the exercise online or as a handout
* OR demonstrate the steps to the class using projection and then have them do the step demonstrated

A facilitation guide is provided below with more details. The session would follow this basic flow:

### Module 1: Brainstorming (Defining the Problem)

Step 1. Brainstorm ideas for the story to be animated. Hand out or display story starter ideas. Ensure everyone has an idea to move forward with. This is a good point to incorporate a pitch session, where everyone presents their idea to the class for feedback.

### Module 2: Write a Script and/or Draw a Storyboard (Understand the Problem)

Step 2 - 3. Have the students write a story script or draw a storyboard. Ensure everyone has successfully created the sketch / script. This is another place where you could interject a pitch or presentation component as a large group or smaller groups to allow students to present their work for review and feedback.

### Module 3: Write an Algorithm Design and/or Flowchart (Design a Plan pt 1)

Step 4 - 5. Have students identify the nouns (objects) and verbs (actions) in the script or storyboard description. Translate the events into steps and directions in an algorithm or components of a flowchart. Ensure everyone has successfully written an algorithm or drawn a flowchart. Students must share their design with the instructor or a peer before implementing.

### Module 4: Plan Your Scene (Design a Plan pt 2)

Step 6 - 8. Have the students identify the background information and characters in the script or storyboard and create a sketch or description of the virtual environment. Ensure everyone has successfully created the sketch / description.

### Module 5: Build Your Scene (Implement Design – Scene Building)

Step 9 - 13. From the set design or description of the environment use the scene builder in Alice to create a virtual environment. Ensure that everyone has created a scene.

### Module 6: Use Comments to Plan Your Program (Implement Design – Optional)

Step 14 – 17. Use the comments tool from the control panel to outline the algorithm design or flowchart in comments in myFirstMethod or custom procedures depending on the skills of the students. Ensure everyone has added comments outlining their story in the Code Editor. Comments are required and essential to the success of the implementation.

### Module 7: Program Your Animation (Implement Design)

Step 18 – 19. Use the Code Editor to construct all or part of the animation. Ensure everyone has successfully added code statements to myFirstMethod.

### Module 8: Run Your Animation (Test)

Step 20 – 21. Run and test your animation as often as desired. Ensure everyone is able to run their animation.

### Module 9: Iterate Until You Are Happy (Iterate)

Step 22. After testing revisit the different steps of the problem solving process as many times as needed until you are happy with the animation. Ensure everyone is continuing to iterate through the design process to incrementally build or revise their animation.

## Assessment (Optional)

You can use the supplied bank of assessment questions, challenges, and exercises to quiz your students on the retention of their new skills. These materials are provided in a separate document that can be downloaded from the webpage associated with this guide. A word document has been provided to allow you to customize as needed.

## Class Regroup + Summary

We recommend regrouping as a class to discuss challenges and successes, and to offer feedback, both among the participants and about the curriculum itself. There are provided reflection questions found below. For this lesson, it is also a great time to have a screening of the projects. If possible, set up a projector and invite each student to introduce their project by giving the title and any desired backstory for their project before playing.

# Standards and Integration

## Interim 2016 CSTA K-12 CS Standards

Algorithms and Programs - 1B-A-2-1:

Apply collaboration strategies to support problem solving within the design cycle of a program

Algorithms and Programs - 2-A-2-1:

Solicit and integrate peer feedback as appropriate to develop or refine a program

## 

# Lesson Material Talking Points

### 

### One: Design Process Introduction

### 

### Slide 3-4 Design Process

* At its core, computer programming designs and implements creative solutions to address human needs.
* The problem solving approach typically used in computer science is based on the traditional problem solving steps used in mathematics courses, such as algebra word problems.
* The problem solving approach can be applied in Alice represented in a cyclic model.
* The cycle includes four major steps:
  + Understand the problem
  + Design a plan
  + Implement the design
  + Test

### Slide 5 An Alice Problem Statement

* A problem can be a creative problem. In Alice your problems are something like:
  + I want to create a story where ….
  + I want to build a world where….
  + I want to build a game where…..
* Defining your problem for Alice can be done through a creative brainstorming process
* For a story brainstorm think about starters such as favorite songs, jokes, eCards, comic strips, books and more.

### Slide 6 - 7 Understand the Problem

* The goal of understanding your problem is to be able to better plan a design solution.
* Different Alice goals will have different options for deconstructing the problem and have different key points of information.
  + An animation can be represented by a script or a storyboard
  + An interactive open world might require a map with discreet storyboards for points of interest
  + A game will require a design document that helps you think about how you win or lose and conditional game interactions and scoring
* This lesson will focus on understanding an animation problem

### Two: Script Writing and Storyboarding

### 

### Slide 9-13 Understanding the Problem

* The primary objective of writing a story or drawing a storyboard is to identify the following:
  + Where does the story take place?
  + What characters are involved in the story?
  + What are the key events in the story?
* Writing a script or storyboard will help you think through the key elements of the story to be animated. Both tools are a good way to explore and understand your story before starting to build the scene and program your animation.
  + A script is a set of directions for a director to program a play on a stage. The important parts of a script are:
    - Set design descriptions introducing the scene
    - Directions for movements around the scene
    - Specific dialogue
  + A storyboard is a frame by frame visual representation of a story used for planning animations or films. Each frame is a sketch, and sometimes includes a caption describing the intention of the frame, to communicate:
    - Visual cues about where the event is happening
    - Representations of who or what is the focus of the scene
    - Representations or notes describing what is happening
    - The details of the sketches are minimal, with only enough to convey key story bits. The goal is to tell the entire story in the simplest visuals possible.
  + Both scripts and storyboards are great tools to better understand and flesh out the details of your story. Iterating a storyboard for animation can be very helpful because it:
    - Can be quicker than writing out all of the details
    - Gives extra information about the framing of the camera that is important to think about in animations
    - Gives more visual information about the scene that can be helpful for planning your animation
  + Be sure to turn your storyboard into comments before starting to develop

### 

### Slide 14 - 15 Design a Plan

* The planning process is not just for the entire program
* There will be times when a portion of the problem requires decomposition
  + Solve a complex problem by breaking it down into a series of simpler problems to be solved
  + When all the simpler problems are solved, the complex problem will be solved
  + This involves the same problem solving process
* It would be good practice to refer to the problem design diagram throughout the problem-solving process, and throughout the course
* To design a plan for animating our story, we combine set design with program design
  + The set design will be translated into the 3D scene of an Alice world
  + The program design will be translated into program code
* Remember, a design plan can be broad and generalized to span an entire task, or it can be a segment of a larger whole

### Three: Set Design - Using Maps and Diagrams

### 

### Slide 16 - 19 Set Design

* Design will be translated into the 3D scene of an Alice world
  + Scripts often include a description of the setting
  + Each frame of a storyboard will give you information about the background
* A quality set will give the story a sense of place and history
* Each prop needs a reason for being included, whether to describe a location or show aspects of a character’s personality
* Some animations will only need a single location and set design similar to a theater set. More complex animations, that move through space and include camera moves, may require a larger environment and will require a level or world design approach.

### Four: Design a Plan - Algorithm Design

### 

### Slide 20 - 25 Algorithm Design

* An algorithm consists of a step-by-step list of actions that provides a description of how to perform the task
* In Alice, the objects and actions are needed for outlining the steps to be performed in carrying out the task
* To recognize a task description, ask if this describes the plot directly in the form of:
  + Objects and actions (e.g., penguin skates)?
  + Information about how an action is performed (e.g., penguin skates slowly)
* In the story example, the objects (nouns) are highlighted in blue and the actions (verbs) in red.
  + Note that not all nouns and verbs are highlighted. We have highlighted only those that are essential to creating the animation
* The algorithm design will need to add formatting that describes how multiple actions should be performed:
  + The *Do In Order* format indicates the instructions are to be performed in sequence
  + The *Do Together* at the beginning of a block set of instructions tells you that they are to be performed simultaneously

### Five: Program Planning - Using Flowcharts

### 

### Slide 26 – 29 Using Flowcharts

* A common way to present programs and user experiences are different types of activity diagrams, or flowcharts. These charts visually represent the flow of the program or process (smaller part of a program) and data, as the program executes.
* Flowcharts are good for representing non-linear program flows and can also help to show more complex interactions and dependencies like simultaneous events, user inputs decision points, and repetitions.
* There are a standard set of symbols that can be used
  + Filled circle or round rectangle – show where the program (or process) begins
  + Arrows – show the directional path or flow through the program or process
  + Boxes – typically the main component of an activity diagram.
    - Show the activity that will take place at that stage in the process
    - Can represent a simple activity, or a more complex sub-process
  + Solid lines – allow you to represent concurrent activities; in other words activities that take place at the same time in the process
  + Diamond – represent decision points in the flow of the program

### Slide 30 - 32 Implement Design

* Computer programming problem solving begins with the goal of writing code to perform a task and often involves repeated steps
* Newly written code requires frequent testing, and it is possible that revisions will be needed along the way.
* The problem-solving approach used in Alice is represented in a cyclic model
* Remember, a design plan can be broad and generalized to span an entire task, or it can be a segment of a larger whole
* You can approach implementing your design incrementally for small parts of the whole and continue to revisit the design cycle into smaller components until it is easy to translate the algorithm into Alice code
* Use comments to outline your program, before implementing, to make it easier to track your progress and for you and others to understand what parts of the code are intended to be implemented
* Be sure to have a classmate or your instructor review your design before you begin implementing

### Slide 33 - 35 Testing

* Testing in Alice was designed to be easy so that you can test often. This allows you to easily:
  + Test your code’s functionality to find bugs
  + Test your design to see if you like the way it feels
* Alice has tools that allow you to disable code to test parts of your program or fast forward to later parts of your program to help you incrementally test parts of your implementation quickly

### Six: Tips and Tricks

### 

### Slide 36 – 39 Tips and Tricks

* Break down the problem into smaller components, if the problem doesn’t make sense
* You can time camera moves with ground, sky color, and texture changes to make new locations if you need more than one scene
* Remember when setting up your scene to position objects that you want to enter the scene or appear later in your animation off screen or set to 0 opacity

# Exercise Facilitation Step-by-Step

These step-by-step directions are for the guided facilitation option 2 that uses the Design Tutorial as a basis for the hands-on experience for the session. They can be followed in addition to having first gone through the slides or we have provided information for integrating the exercise into the lesson allowing you to jump back and forth from the lesson to practice.

## Module 1: Brainstorming (Defining the Problem)

### Goal – Complete Step 1 of Tutorial Exercise

Students will be able to define a story problem by brainstorming a story idea.

### Media

* Allow students to surf the internet looking for ideas based on the provided starting points
* OR provide story starter ideas materials of your choosing
* OR facilitate a brainstorming activity as a group

### Talking Points

* Emphasize that problem solving is a dynamic process, that there is testing and revision of the design throughout the problem solving process so their initial idea can change and be improved upon.
* Some ideas for sparking story ideas are:
  + Characters singing your favorite song or animating a music video
  + Characters telling jokes (ex. A knock knock joke)
  + Your favorite internet meme
  + Funny or sentimental eCards
  + Existing comics strips and cartoons
  + Even a serious conversation or event
* You might want the students to present / share their story ideas

## Module 2: Write a Script and/or Draw a Storyboard (Understand the Problem)

### Goal – Complete Steps 2-3 of Tutorial Exercise

Students will be able to understand their story by translating their story idea into either a script or storyboard.

### Media

* Show slides 8-12 of the Program Design In Alice presentation
* OR share example scripts or storyboards
* OR demonstrate writing part of a script or drawing a frame of a storyboard

### Talking Points

* Emphasize that students should be sure to include background information about the scene including information about the environments and the props in the world
* Be sure students focus on the critical events or actions from the story
* You might want the students to present / share their script / storyboards

## Module 3: Write an Algorithm Design and/or Flowchart (Design a Plan part 1)

### Goal – Complete Steps 4-5 of Tutorial Exercise

Students will be able to translate a script or storyboard into an algorithm design or flowchart by identifying the objects (nouns) and actions or events (verbs) of their story

### Media

* Show slides 21-24 and 27-29 of the Program Design In Alice presentation
* OR share examples of Alice algorithm designs and Alice flowcharts
* OR demonstrate translating a story into an algorithm design or flowchart
* You may distribute the StoryboardTemplate.pdf

### Talking Points

* The objects and actions are needed for outlining the steps to be performed in carrying out the task
* To recognize a task description, ask if this describes the plot directly in the form of:
  + Objects and actions (e.g., penguin skates)?
  + Information about how an action is performed (e.g., penguin skates slowly)
* Identify the nouns (objects) of the story and verbs (actions)
  + Note that not all nouns and verbs are highlighted. Identify only those that are essential to creating the animation.
* Identify the supporting details (information about how an action is performed)
* Identify what order the actions occur in and if they happen in order or simultaneously
* You might want the students to present / share their algorithm / flowchart.

## Module 4: Plan Your Scene (Design a Plan part 2)

***Goal – Complete Steps 6-8 of Tutorial Exercise***

Students will be able to translate background information into a simple set or level/world design

### Media

* Show slides 17-19 of the Program Design In Alice presentation
* OR share examples of a set design or level design
* OR demonstrate translating a script or storyboard into a set or level design sketch

### Talking Points

* Analyze your script or storyboard for determining where your story takes place and what background props you need to support your story
* Think about what camera locations you need to support the different views represented in storyboards or locations in a script

## Module 5: Build Your Scene (Implement Design – Scene Building)

***Goal – Complete Steps 9-13 of Tutorial Exercise***

Students will use their set or level design as a blueprint to build out their scene using the Scene Editor

### Media

* Use materials from the Building a Scene Lesson and related *How To* videos
* OR demonstrate and refresh the basics of the Alice Scene Editor

### Talking Points

* You can animate the camera the same way you can animate any other object in Alice.
* Remember what you learned about using the Scene Editor and refer to *How To* videos and Quick Reference Guide to help you build out your scene

## Module 6: Use Comments to Plan Your Program (Implement Design – Optional)

***Goal – Complete Steps 14-17 of Tutorial Exercise***

Students will be able to translate their algorithm design or flowchart into comments inside Alice for reference when building their program.

### Media

* Show an example of a translated algorithm in comments in myFirstMethod in Alice
* OR demonstrate adding comments into myFirstMethod in Alice

### Talking Points

* Comments can be a great way to keep track of what segments of your code are supposed to do and are critical for helping others decipher your code

## Module 7: Program Your Animation (Implement Design)

***Goal – Complete Steps 18-19 of Tutorial Exercise***

Students will use their transferred comments, algorithm design, or flowchart as a blueprint to build out their animation using the Code Editor.

### Media

* Use materials from the Programming in Alice Lesson and related *How To* video materials
* OR demonstrate and refresh the basics of the Alice Code Editor

### Talking Points

* Remember what you learned about using the Code Editor and refer to *How To* videos and Quick Reference Guides to help you build out your scene
* Don’t forget that you can follow the design process and problem solving steps for parts of the whole so don’t feel that you have to implement the whole animation all at once

## Module 8: Run Your Animation (Test)

***Goal – Complete Steps 20-21 of Tutorial Exercise***

Students will be able to run all or parts of their animation to evaluate the outcome and determine if they need to revisit and rethink the design problem or the technical design plan or implementation.

### Media

* Show unsuccessful animations to show how they can be critiqued as designs or technically
* OR demonstrate a quick failure such as the do-together cancelation issue and show the technical iteration

### Talking Points

* The design and problem solving process is a cyclical process designed to allow you to test and redesign as often as you need
* Alice is designed specifically to allow you to easily compile and run your world so that you can test and fix often

## Module 9: Iterate Until You Are Happy (Iterate)

***Goal – Complete Step 22 of Tutorial Exercise***

Students will understand that the problem solving and design process is a cyclical system and that after testing you can return to understanding the problem, designing a plan, or implementation to fix problems that arise depending on the issue.

### Media

* Show the problem solving process diagram (slide 3 of Design a Program in Alice slides)

### Talking Points

* The problem solving process is a cycle where each step can be revisited as often as you need. Be sure to think about whether the problem you are facing is a story problem or a technical problem and jump back to that step in the cycle and make changes.

# Reflection Questions

* What is the value of planning?
* Did you comment? Were they helpful when laying out your program?
* Did you change your design when implementing? Did you revisit your design and comments before starting to program again?
* When reviewing the program or finding things you wanted to change did you add comments into the program first?
* What are comments for? Why do you need them?