

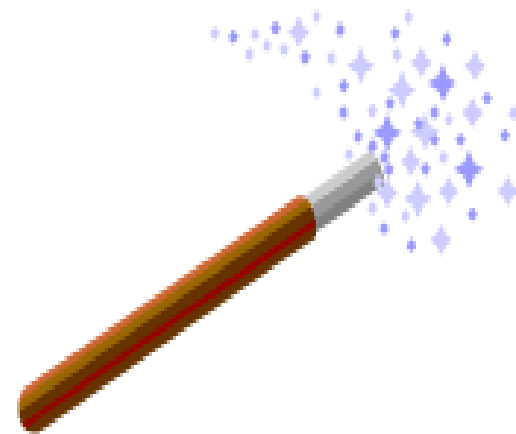
Telling a Story Visually

What Will I Learn?

Objectives

In this lesson, you will learn how to:

- Compare and define an animation and a scenario
- Demonstrate how to use the four problem solving steps to storyboard your animation
- Use functional decomposition to write a storyboard
- Flowchart a storyboard
- Describe an algorithm





Why Learn It?

Purpose

The creation of an animation or game can be enhanced by following a systematic process.

Writing down the components of an animation or game will help you to program and test more effectively, resulting in a more impressive animation or game.





Problem Solving Approach to Animation

A big problem, or sometimes even a little problem, is often solved by breaking down the process into parts.

Functional decomposition is the process of taking a complex problem or process and breaking it down into smaller parts that are easier to manage.

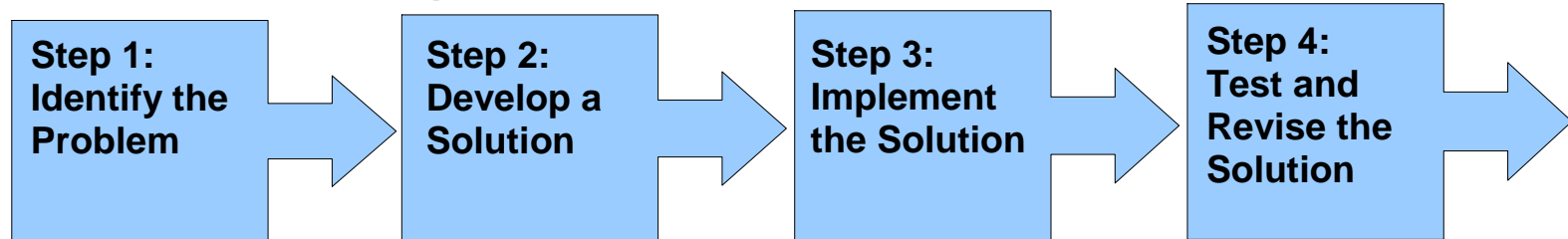
Examine this high level process:

- Consider a math concept that requires many steps.
- Identify the high level steps for the math concept.
- Further refine and define the tasks needed for each high level step.
- Present the problem as an animation.

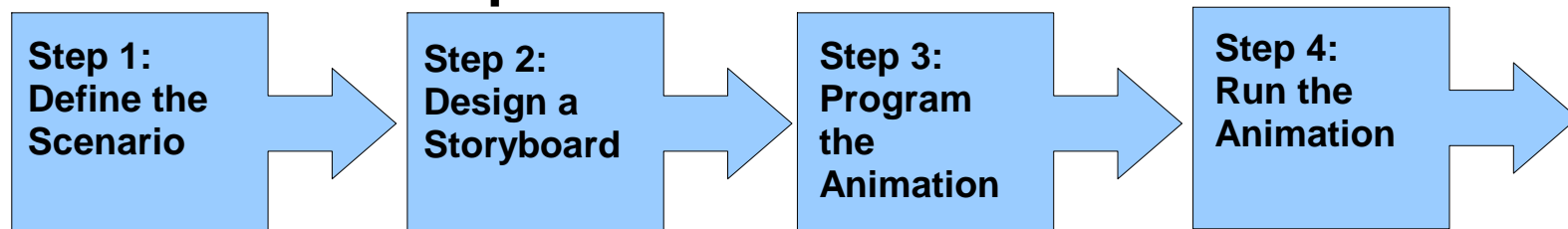
Problem Solving Approach to Animation (cont.)

The process to develop an animation is very similar to the problem solving process. Compare the problem solving and animation development process listed below.

Problem Solving Process



Animation Development Process





Animation Defined

An animation is a sequence of actions that simulate movement.

As the animation creator, you will program in Alice what the animation should do, and Alice will render the animation for you.

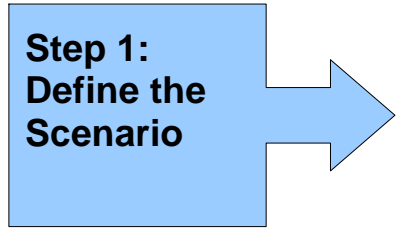
Rendering is the process whereby the system converts the image frames into a seamless animation. Alice3 does all of the work necessary to render the animation based on what instructions you provided for how the objects should act.





Step 1: Define the Scenario

Professional animators begin the process by developing a scenario—or story—that gives the animation a purpose.



Step 1:
Define the
Scenario

A scenario is a story in the form of a problem to solve or task to perform.

Examples

- Story representing a conflict and resolution
- Lesson to teach a math concept
- Process to simulate or demonstrate
- Game to play



Step 1: Define the Scenario (cont.)

Examples of scenarios and animations are described below.

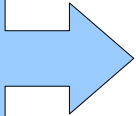
Scenario Type	Scenario	Animation
Story	A cat is stuck in a tree and needs help getting down.	A man climbs up a tree to save the cat.
Lesson	Students have trouble memorizing chemistry symbols.	A timed matching game to match chemistry symbols with their definitions.
Process	A demonstration of how to change a car tire.	Virtual tools to show you how to change a tire on a virtual car.
Game	An airplane flies through the sky but should not hit objects in its path.	An interactive game to fly an airplane around objects in the sky.



Step 2: Design a Storyboard

A storyboard details each scene of your animation. It includes the following components:

Step 2:
Design a
Storyboard



Component	Definition	Example
Object	A moving or non-moving character that you position and/or program to move and act	Animals, cars, people, trees
Scene	The place (or “world” in Alice) where your story occurs	Park, library, school, home
Actions	The instructions for how each object should act in the scene	Walk 2 meters, turn left, say “Hello!”
User interactions	The ways in which the user can manipulate the objects in the animation	Keyboard commands or mouse clicks to make objects move
Design specifications	How the objects and scenery should look in the animation	Size, position, location, color

Step 2: Design a Storyboard (cont.)

There are two types of storyboards:

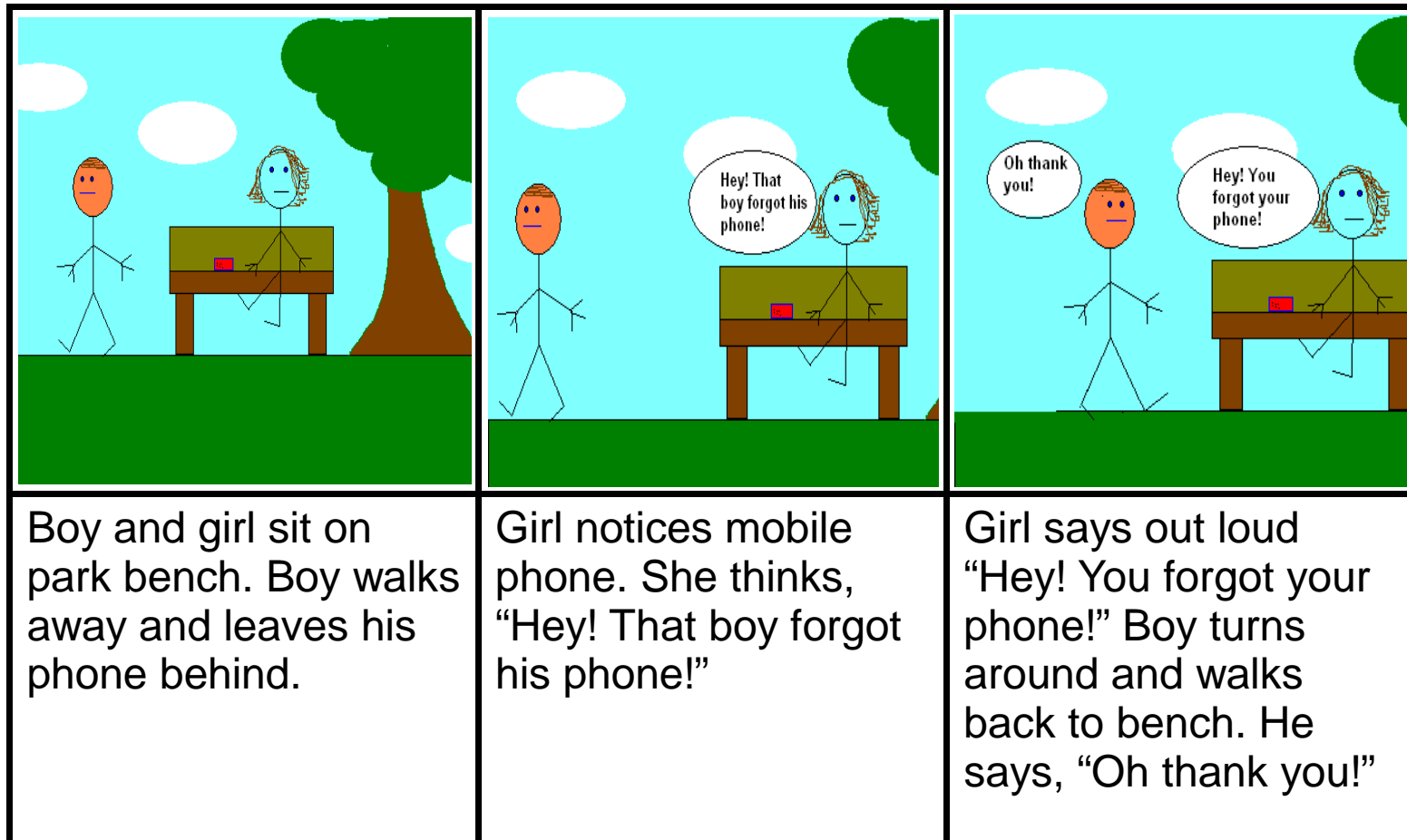
- Visual – a series of illustrated images that represent the main scenes in the animation
- Textual – detailed, ordered list of the actions each object performs in each scene in the animation

In computing, a textual storyboard is called an algorithm: a list of actions to perform a task or solve a problem.

A storyboard can be created on paper, or using digital tools such as a word processing program, paint or drawing program, or presentation program.

Step 2: Design a Storyboard (cont.)

Visual Storyboard Example



 Step 2: Design a Storyboard (cont.)**Textual Storyboard Example**

Do in order

Boy and girl sit on park bench.

Boy stands up and walks away, leaving his mobile phone on the park bench.

Girl turns to look at phone.

Girl thinks, “Hey! That boy forgot his phone!”

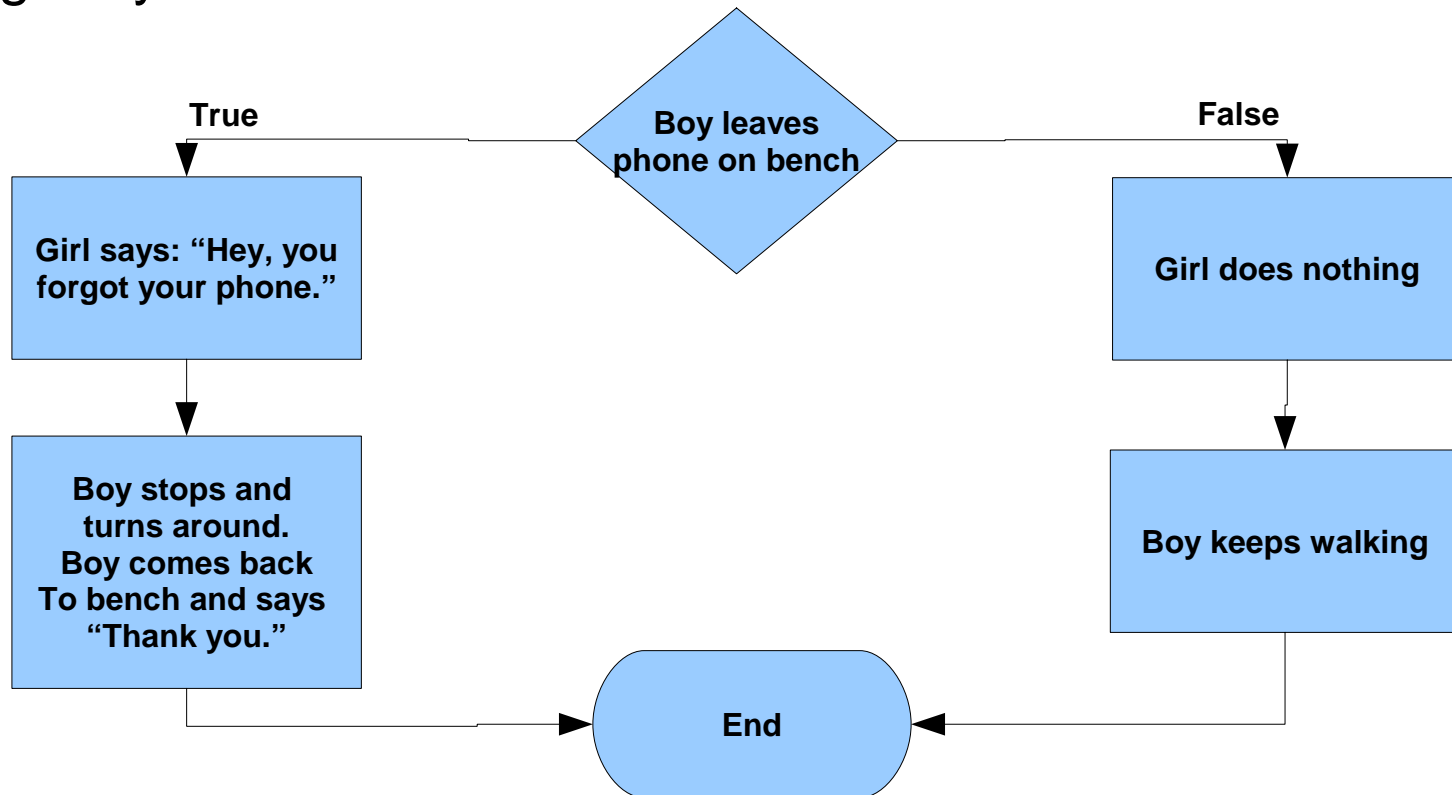
Girl says out loud, “Hey! You forgot your phone!”

Boy stops and turns around.

Boy walks back to park bench and says, “Oh thank you!”

Step 2: Design a Storyboard (cont.)

Flowchart the storyboard so that you can organize the process of the animation's actions, and ensure each flows logically.



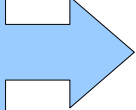
Step 3: Program the Animation

After completing your visual and textual storyboards you program your animation in Alice.

Your storyboard identifies the design specifications for the animation (how objects appear, move, speak, interact, etc.)

Your program code provides the run-time instructions to Alice.

Step 3:
Program
the
Animation

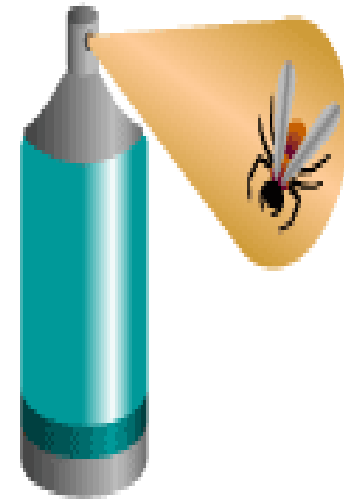


Step 4: Run the Animation

When you finish programming your animation, you run and test the animation to ensure proper execution. This process is often referred to as testing and debugging the software.

Programs are tested by entering unanticipated commands to try and “break” the code. When something is broken or doesn't work as you intended in a software program, it is often referred to as a “bug”. Debugging is the process of finding software bugs in a software program.

Step 4:
Run the
Animation





Step 4: Run the Animation (cont.)

Test and debug your animation frequently. Below is a list of debugging tasks you may consider:

- Adjust the direction, distance, and duration that objects move.
- Adjust math calculations that help refine the distance or duration that objects move.
- Change procedures (instructions in the code) that do not work as intended to ones that work better.
- Resolve errors created by the programmer.



Terminology

Key terms used in this lesson:

Algorithm

Animation

Bug

Debug

Functional decomposition

Problem solving steps

Program

Render

Scenario

Textual storyboard

Visual storyboard



Summary

In this lesson, you learned how to:

- Compare and define an animation and a scenario
- Demonstrate how to use the four problem solving steps to storyboard your animation
- Use functional decomposition to write a storyboard
- Flowchart a storyboard
- Describe an algorithm



Practice

The exercises for this lesson cover the following topics:

- Understanding animations, scenarios, and problem solving steps
- Flowcharting and writing a storyboard
- Demonstrating functional decomposition