

Unit I: Introduction: Installation and Alice Navigation

Chapter 3: The Software-Lifecycle

In this chapter, you will learn how to use planning tools and techniques to design and manage larger software projects. You will explore how programmers plan a solution before writing code and how top-down design breaks a complex program into smaller, more manageable parts. By organizing a project into individual tasks and components, you can make the program easier to develop, test, debug, and maintain.

Objectives

- Learn to use several tools traditionally used by programmers;
- Learn to manage larger programs using top-down design,
- Create flowcharts,
- Write pseudocode,
- Create storyboards.

Pseudocode

Pseudocode is step by step instructions (statements) that the program will perform, but written in ordinary human language rather than a programming language. It allows the programmer to focus on the steps that need to be taken rather than the exact syntax of the programming language.

Example: Penguin Family

Story: The story we want to tell is about a penguin family. A mother penguin is walking on an iceberg with an egg. She walks over to the father and gives him the egg. The mother walks to the edge of the iceberg, says "I'll be back." and dives in the water. The father walks back and forth with the egg for a long time. The egg hatches. The chick says "Daddy!" The father and chick walk back and forth together for a long time. The mother jumps onto the iceberg with a fish in her beak. She walks over to the father and chick. The chick says "Momma!" The father takes the fish and says, "Thanks Honey!"

Pseudocode:

1. Mother penguin is walking on an iceberg with an egg resting on her feet.
2. She walks next to the father
3. Mother transfers egg to father. (Need to change the vehicle for the egg?)
4. Mother walks to the edge of the iceberg.

5. Mother says, "I'll be back."
6. Mother dives in the water.
7. Father penguin walks back and forth with the egg 5 times.
 - He walks to the left edge
 - turns
 - walks to the right edge.
 - show passage of time by changing sky
8. The egg hatches.
 - the egg becomes invisible
 - the baby penguin becomes visible
9. The chick says "Daddy!"
10. The father and chick walk back and forth together for a long time.
11. Father penguin and chick walk back and forth together 5 times.
 - They walk to the left edge
 - turn
 - walk to the right edge.
 - show passage of time by changing sky
12. The mother jumps onto the iceberg with a fish in her beak.
13. Mother walks over to the father and baby penguin.
14. Baby penguin says "Momma!"
15. Father takes the fish from the mother
(Change vehicle for fish?)
16. Father says "Thanks Honey!"
17. Scene of happy penguin family.

You may find that there are some steps that you haven't figured out the details for yet. Such as how to transfer the fish and the egg from the mother to the father.

Example: Rock, Paper Scissors

The game

Rock, paper scissors is a well-known game in the United States. Two players count 1, 2, 3, then each throw down one hand as a rock (closed fist), paper (all fingers out), or scissors (two fingers out).

The winner is as follows:

- Paper beats rock. (*Paper covers rock*)
- Scissors beats paper. (*Scissors cuts paper*)
- Rock beats scissors. (*Rock crushes scissors*)
- If the two are the same it is a tie.

Story: Odin and the Troll will play a game of rock paper scissors.

Pseudocode:

1. Odin will count down 3, 2, 1.
2. A random number from 0 to 2 (0,1,2) will be selected.
3. If the number is 0, Odin will show rock.
4. If the number is 1, Odin will show paper
5. If the number is 2, Odin will show scissors.
6. Another random number from 0 to 2 (0,1,2) will be selected.
7. If the number is 0, Troll will show rock.
8. If the number is 1, Troll will show paper
9. If the number is 2, Troll will show scissors.
10. Decide the winner

You may find that there are some steps that you haven't figured out the details for yet. Such as how to show the winner and how to manipulate the hands to show rock, paper, and scissors. We will use step wise refinement to work out all the details.

Flowcharts

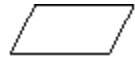
Flowchart: A flowchart uses specific symbols to show action, loops and decisions. This might be a good choice if you are creating a game or a story that depends on user

input.

A flowchart is a drawing that shows the steps in an algorithm. The order of the steps can be seen by following the arrows. Programs can have branches and loops, and the flowchart makes it easier to see the sequence of the algorithm. Different shapes are used to represent each type of action.



Terminal: The oval is used to indicate the beginning and end points of a flowchart.



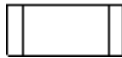
Input and Output: The parallelogram is used to represent both input and output.



Assignment: A rectangle is used for assignment statements, or calculations.



Decision: The diamond is used for decisions. A Boolean condition (an expression that will evaluate to either true or false) is shown inside the diamond. There will always be 2 lines out of the diamond: one marked T (true) and one marked F (false). Evaluate the Boolean expression to determine which line to follow.

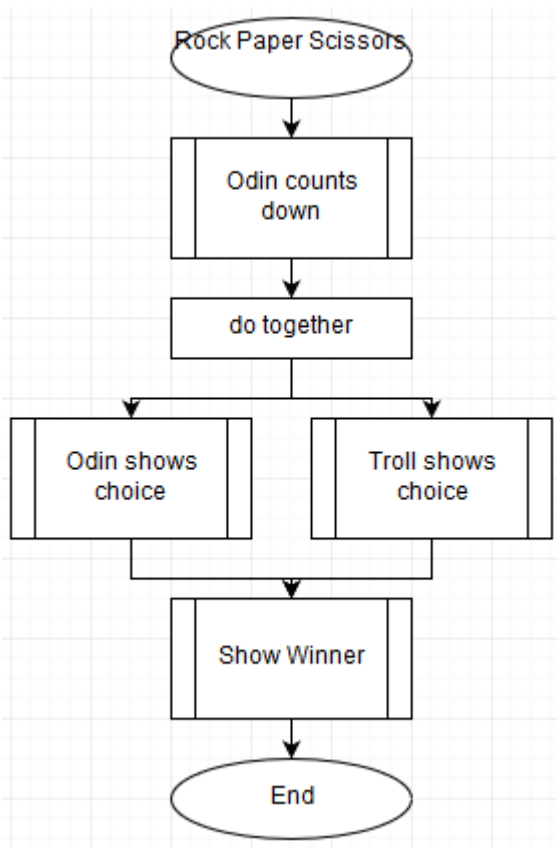


Procedures and Functions: A procedure is a group of statements that can be grouped together logically to perform a task such as walking, waving or moving next to another object.

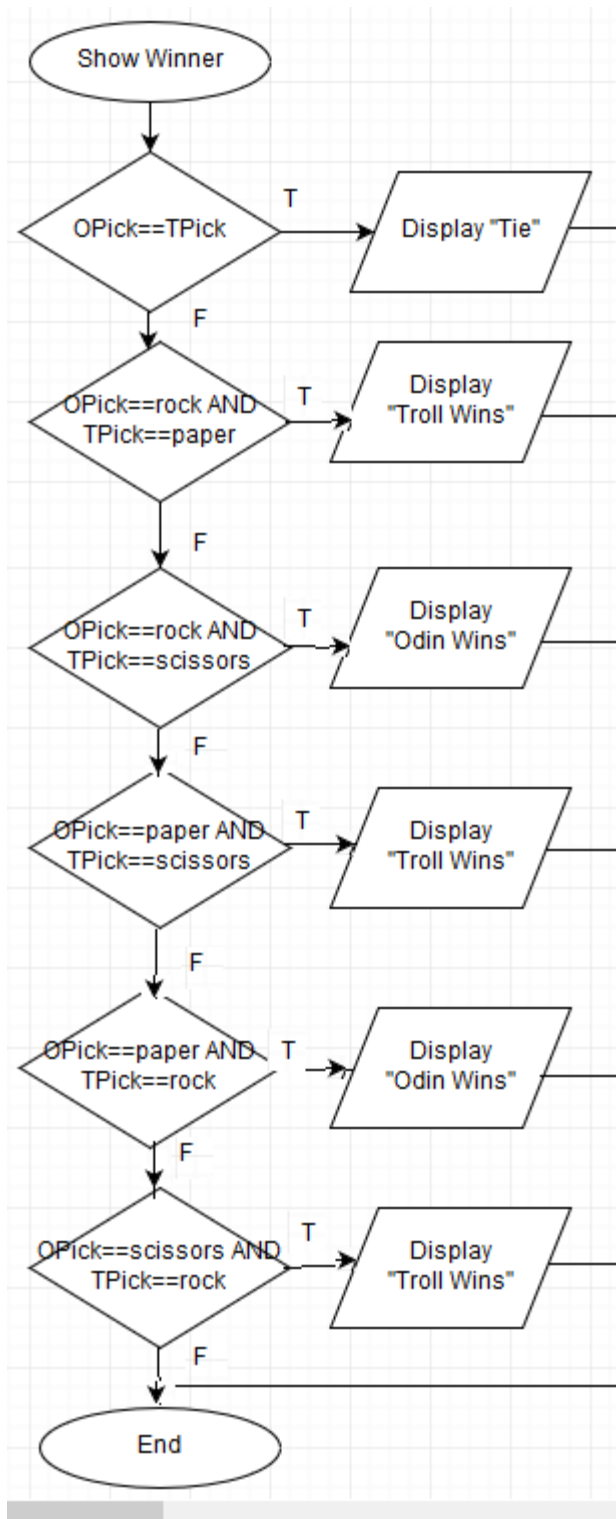
A procedure is represented by a rectangle with a bar on each side. Each procedure within a program will have its own flowchart. The flowchart for the example is not complete.

Flowchart: Rock, Paper Scissors

The top-level flowchart for the rock paper scissors game is shown and the procedure for finding the winner. The procedures for counting down and for the bipeds to select and show their choice is not shown.



This flowchart starts with an oval labeled with the title of the program: "Rock paper Scissors". An arrow at the bottom of the oval points down to a rectangle with a bar on the left and right (indicating a procedure). It is labeled "Odin counts down". A down arrow points to a rectangle labels "Do together". The do together rectangle branches to two procedures: "Odin shows choice" and "Troll shows choice". These two procedures both branch to a procedure labeled "Show Winner". There is a down arrow from the show winner procedure to an oval labeled "End" **This is top-down design,**



This flowchart shows the details for the "Show Winner" procedure. Notice that this is the same name shown in the top-down flowchart. It starts with an oval labeled "Show

Winner". A down arrow point to a diamond (decision) labeled "**OPick==TPick**" (Note: the double equal, **==**, indicates **comparison** rather than a single equal that is used for **assignment**.) An arrow to the right if label "T" (True) goes to an IO (input/output) symbol that is a parallelogram. This is labeled "Display Tie". If **Opick** is not equal to **TPick**, a down arrow labeled "F" (false) points to a diamond labeled "Opick==rock and TPick==paper" an arrow to the right is labeled "T" and points to an IO block labeled "Odin wins". If the decision if false, A decision block says Opick==scissors and TPick==rock. If this is true it displays "Troll Wins". All of the display parallelograms go to the right and down to the parallelogram that is labeled "End".

Storyboards

Storyboard: In movies and animation, a storyboard is often used.

Note: the storyboard is a design tool. It is NOT clips from the finished movie.

A storyboard shows each object's position, pose and orientation. It should also show you where the camera will be. The storyboard will show if the camera is moving. It also shows the text, dialog and sound that will be included.

The storyboard should show:

Frame: Each frame shows the action, location, and characters in the Alice virtual world where the story takes place.

Frame title: Each frame must have a unique title.

Objects: actors, props, billboards, text.

Actions: all actions that occur with objects, camera or scenery changes.

User Interactions: Any input or response to questions or other interactions of the user.

Sounds: Including background sounds such as ocean, thunder, or horns honking, or spoken dialog.

Storyboards can also help to identify actions that are repeated.

Truth Table

A **truth table** is a tool programmers use to show the possible results of Boolean expressions. It lists all the different combinations of values and shows the resulting value. Truth tables are especially useful when working with logical operators such as **AND**, **OR**, and **NOT**. For example, an AND expression is true only when both conditions are true, while an OR expression is true when at least one condition is true. A NOT operator reverses a Boolean value, changing true to false or false to true. Programmers can use truth tables to plan and check the logic of decisions before incorporating them into **if/else statements** and other parts of a program.

Try to make a truth table for the Rock, Paper, Scissor game before you turn to the next page.

Computer	User	Result
rock	rock	It's a tie
rock	paper	You win
rock	scissors	Computer wins
paper	rock	Computer wins
paper	paper	It's a tie
paper	scissors	You win
scissors	rock	You win
scissors	paper	Computer wins
scissors	scissors	It's a tie