

## Program Design

When you write very short programs with only a few steps, trial and error may work. However, as your programs get longer you will need to do more planning. Start with a general idea of what you want to do, then break it down into subparts. Gradually, you will add more detail to the subparts. This is called top-down design. You may break each of the subparts into even smaller parts. As you gradually add details to the subparts it is called step-wise refinement.

**Step 1:** Write down a brief description of what you want the program to do. In Alice, this is usually the story that you want to tell.

Example: A mother penguin leaves her egg with the father to go fishing. The mother returns home with a fish and finds that the egg has hatched.

**Step 2:** Start refining the story to add more step by step details.

Example: 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!"

**Step 2:** Identify the scenery, actors and props.

Scenery: A snowy setting with an iceberg and water.

Props: Egg, fish

Actors: Mother penguin, Father penguin, baby penguin (chick)

**Step 3:** What actions are performed?

Penguins walk

Mother penguin transfers egg to father

Penguin dives

Egg hatches

Penguin jumps out of the water.

Mother penguin transfers fish to father.



Author: Janet E. Joy; Publisher: Zebra0.com

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/)

What is unknown? How to show passage of time.

**Step 4:** Develop an algorithm. There are many tools that a programmer can use to design a program.

**Pseudocode:** 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:

1. Mother penguin is walking on an iceberg with an egg resting on her feet.
2. She walks next to 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.
  - a. He walks to the left edge
  - b. turns
  - c. walks to the right edge.
  - d. show passage of time by changing sky
8. The egg hatches.
  - a. the egg becomes invisible
  - b. 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.
  - a. They walk to the left edge
  - b. turn
  - c. walk to the right edge.
  - d. 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.**



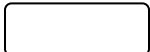
Author: Janet E. Joy; Publisher: Zebra0.com

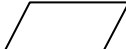
This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/)

## Flowcharts

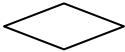
A flowchart uses specific symbols to show action, loops and decisions. This might be a good choice if you are creating a game or a story that depends on user input.

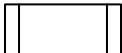
A flowchart is a drawing that shows the steps in an algorithm. The order of the steps can be seen by following the arrows. Programs can have branches and loops and the flowchart makes it easier to see the sequence of the algorithm. Different shapes are used to represent each type of action.

**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. 

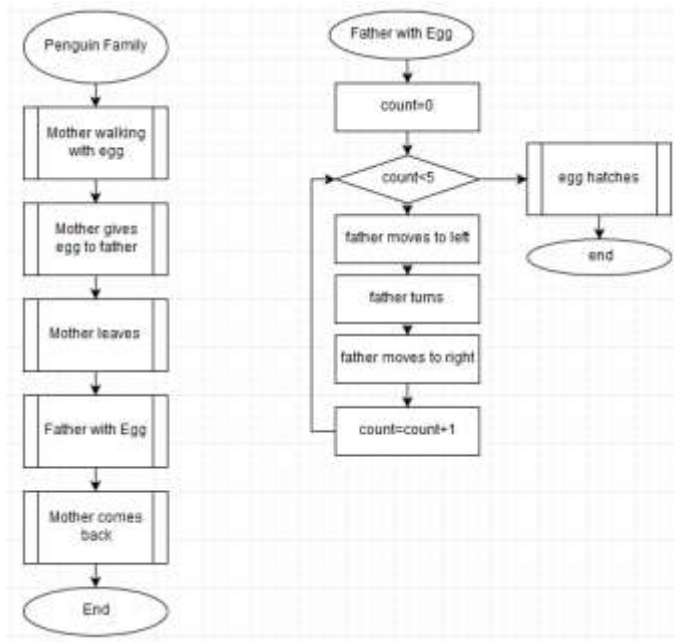
**Procedures and Functions:** A procedure is a group of statements that can be grouped together logically to perform a task such as walking, waving or moving next to another object. 

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.



Author: Janet E. Joy; Publisher: Zebra0.com

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/) Creative Commons Attribution-NonCommercial 4.0 International License



Author: Janet E. Joy; Publisher: Zebra0.com

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/)

## Storyboard

A storyboard is often used in movies and animation to show the sequence of scenes, the position of the actors and the dialog. It looks like a comic strip.

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

A storyboard shows each objects 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:

**Scenes:** the Alice virtual world where the story takes place.

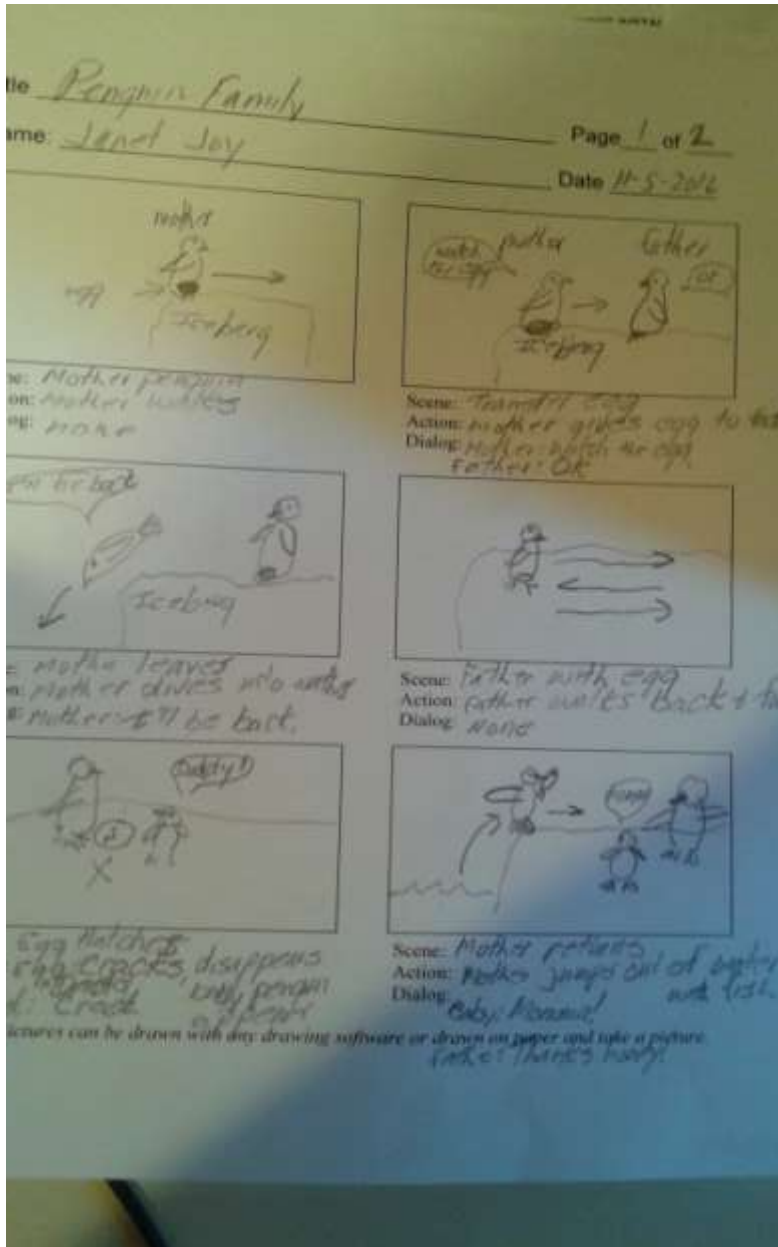
**Objects:** actors, props, billboards, text.

**Actions:** all actions that occur with the 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.



Author: Janet E. Joy; Publisher: Zebra0.com

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/) Creative Commons Attribution-NonCommercial 4.0 International License

## Truth Tables

A truth table is used to show all of the possible combinations for values for each variable and the desired outcome. In the illustration below, a truth table for the rock, paper, scissors game is shown. This was created using a spreadsheet such as Excel.

	A	B	C
1	Computer	User	Result
2	rock	rock	It's a tie
3	rock	paper	You win
4	rock	scissors	Computer wins
5	paper	rock	Computer wins
6	paper	paper	It's a tie
7	paper	scissors	You win
8	scissors	rock	You win
9	scissors	paper	Computer wins
10	scissors	scissors	It's a tie

To create a truth table, first make columns for each variable and the outcome. Put the first value in each column. Then change the last column to the next value and leave the rest of the columns. It is a bit like counting: you start with a 0 in the tens place and use values 0 to 9 in the ones place. After you get to 9 in the ones place, you increment the tens place to 1 and start over at 0 in the ones place. . In this illustration, V1 can have values 1 or 2. V2 can have the values a, b, or c. You can see that V1 is 1 while V2 goes from a to b. Then V2 goes back to the starting value and V1 goes to the next value.

	A	B
1	V1	V2
2		1 a
3		1 b
4		1 c
5		2 a
6		2 b
7		2 c

A spreadsheet is good for creating truth tables because you can sort it by the result column:

	A	B	C
1	Computer	User	Result
2	rock	scissors	Computer wins
3	paper	rock	Computer wins
4	scissors	paper	Computer wins
5	rock	rock	It's a tie
6	paper	paper	It's a tie
7	scissors	scissors	It's a tie
8	rock	paper	You win
9	paper	scissors	You win
10	scissors	rock	You win



Author: Janet E. Joy; Publisher: Zebra0.com

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/) Creative Commons Attribution-NonCommercial 4.0 International License

A truth table can then be used to write the code. A truth table can be used when you test the program to make sure that each desired result is actually produced correctly.

**Step 5: Implementation.** After you have designed the algorithm using any or all of the tools discussed above, you are ready to implement the algorithm, or put it into action. Implementation is the act of translating your algorithm into the necessary computer code.

Implementation will involve creating the program one step at a time. You can start in several ways.

One way is to start at the beginning and add one piece of code at a time. Be sure to save as you go along. You will be testing and revising over and over. It helps to save using incrementing numbers such as penguinFamily1, penguinFamily2, etc. That way if you mess up, you can go back to the previous working program.

Another way is to start with just one piece, such as getting the egg to sit on the mother penguins feet. Then make the mother penguin walk. The father penguin and baby also walk, so make sure that the walking procedure can be used by each of the penguins.

Sometimes you get stuck: How do I get the penguin to dive in the water? If you are stuck, just put in a comment at this point such as //mother penguin dives into water. Or create an empty procedure and call it. Of course, you will have to come back and make sure the penguin actually dives into the water, but you can work on other parts of the program in the meantime.

Reward yourself! If you are stuck, switch to a part of the story that is easy or more fun.

As you go along, you will try several different ways to get the actors to perform the desired actions. Instead of deleting code that doesn't work, disable it instead of modifying it or deleting it. You can even add comments about why it didn't work.

You can also create multiple versions of procedures, such as walk1 and walk2. Then you can test each to see which works better.

Don't get discouraged if it isn't perfect! Remember, in animation and cartoons, the user wants to believe the story. They will accept the story even if the animation is not perfect. Professionally written games and cartoons have large teams of people working on them and spend thousands of man hours in the development.



Author: Janet E. Joy; Publisher: Zebra0.com

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/) Creative Commons Attribution-NonCommercial 4.0 International License