



Welcome to Envision by WorldStrides Family Labs!

Dear Scholar,

Welcome to Envision's Family Lab: Scratch Coding Challenge! Get ready to dive into the fascinating world of computer coding. This is your chance to peek behind the content you see on your devices every day and learn about the language that brings everything from games to online news to life.

You will take on the role of software engineer, using the Scratch platform. Scratch is a project of the MIT Media Lab, designed to help students think creatively, reason systematically, and work collaboratively.

During this lab, you will be introduced to Scratch block coding and challenged to develop your own game or animation. As a software engineer it is up to you to use your imagination and creativity to develop an original Scratch project. Are you wondering how you'll be able to take on the task? You will begin by following the steps listed in your lab manual. Each step in your lab manual will get you closer to understanding the Scratch programming language. Your lab manual is designed to be a resource to successfully complete this challenge.

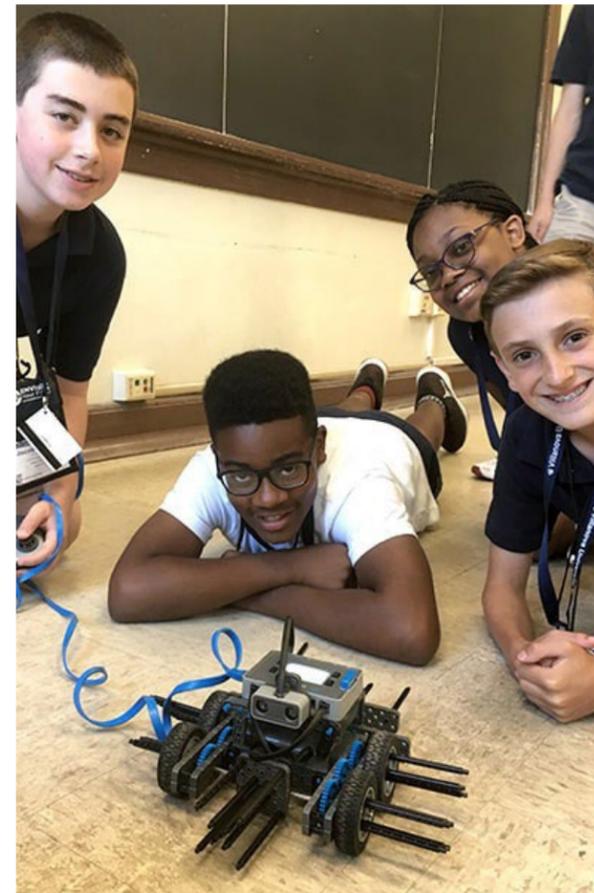
To guide you through the creation process you will use the Design Thinking Process. Professionals in the world of STEM use computer coding and the Design Thinking Process to solve real-world problems. If you enjoy your role as a software engineer, we encourage you to explore different careers in STEM, such as computer programmer, video game designer, network architect, systems analyst, web developer, and many other exciting career paths!

By completing this Family Lab, we hope that you have fun and learn something new about the exciting world of STEM!

Good Luck!

Yolanda Golden

Yolanda Golden
Vice President, Educational Experience
National Youth Leadership Forum: Explore STEM



Know Before You Go!



Why Learn to Code? Learning to code can enhance your problem solving, critical thinking, creativity, analytical, math, and literacy skills, among many others.

Uses for Code... Today most electronic devices and processes require some level of coding to perform their tasks, including computers, smartphones, tablets, even household appliances and cars as well as software, mobile apps, and websites.

Step 1: Create your Scratch account

Go to scratch.mit.edu/join and create an account so you can save and share your animation or game.

To be able to complete the Scratch Coding Challenge and create your own animation or computer game, you must first learn about Scratch coding. Use the resources provided in this lab manual to learn all of the skills and knowledge that you need to complete the challenge.



Click on the magnifying glass to watch a short overview video about Scratch.



Click on the question mark to review the Scratch FAQs

Step 2: Click on the Scratch logo below to watch a video and learn more about MIT's Scratch Coding Platform and the importance of learning to code.

Scratch is a coding language and online community where you can create your own interactive stories, games, and animations -- and share your creations with others around the world. As young people create and share Scratch projects, they learn to think creatively, reason systematically, and work collaboratively. Scratch is a project of the [Scratch Foundation](https://scratch.mit.edu), in collaboration with the Lifelong Kindergarten Group at the MIT Media Lab. It is available for free at <https://scratch.mit.edu>



Coding Languages

Computers use coding languages to understand what people want them to do. Just as humans have developed different languages to communicate with one another, we have also created different languages to communicate with computers. Below are some examples of different coding languages.

Java

One of the most widely used programming languages
Minecraft is an example of a program written in Java

HTML

Stands for Hypertext Markup Language
Most common coding language used for web pages

Swift

Open-source programming language from Apple
Used to create iPhone compatible apps

Python

Versatile coding language considered one of the easiest coding languages to learn

Learn the Lingo!

Step 3: Rollover the terms below and read the definitions to learn some key terms and concepts of coding.

Get to Know Scratch!

Step 4: Click on the video link below and read over the list of Scratch terms to learn more about getting started with Scratch coding.



©Scratch Foundation

Learn some key Scratch terms below:

Stage - Where the sprites perform their actions. You can change the look of the stage by selecting or creating a background.

Sprite Pane - Allows you to add sprites, quickly switch sprites, and view the different scripts each sprite contains.

Code Area - The area where scripts (a stack of blocks) are assembled.

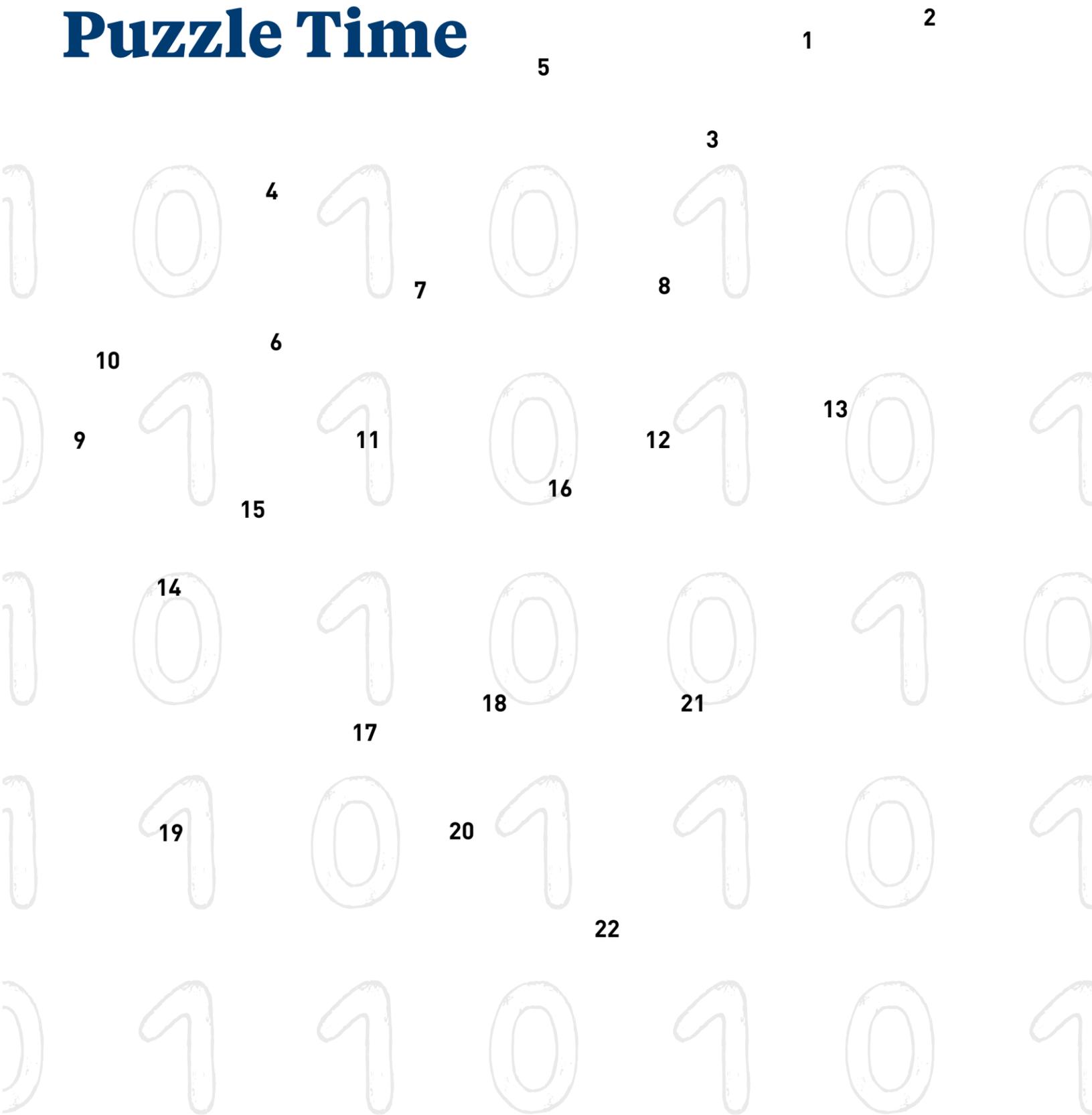
Code Tab - The menu that contains all the different pieces of block code that can be used to program a project.

Sounds Tab - Allows you to choose, record, and upload sounds as well as engineer different aspects of the sound (e.g., fade, speed, and volume).

Costumes Tab - A design tool that allows you to choose, create, upload, and change the appearance of your sprites.

Backdrops Tab - A design tool that allows you to edit your stage backgrounds.

Puzzle Time



Across

1. Computer program usually run on a smartphone or tablet
3. Computer graphics that move via code or LeBron's favorite soda
4. The individual 1's and 0's that make up binary code
6. Language used to tell a computer what to do
7. An error in a program or a small insect
8. An order or instruction for a computer
9. Checks for a condition and then runs a code block
11. Coding language that is also the name of world's most populous island
12. The name of Google's phone that is also short for "picture element"
14. Method of getting input on a project by asking a large number of people
17. Popular singer who needs you to "calm down" and shares a name with a coding language
19. Most common fundamental unit of digital data
20. Both the act of writing computer code and a collection of code
22. Information stored in or used by a computer

Down

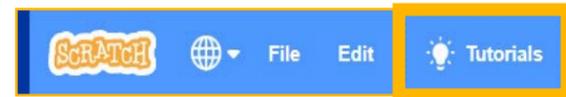
2. Coding language that is also a large snake found in Africa, Asia, and Australia
4. Legos or the type of code used in Scratch
5. The _____ Thinking Process
7. Testing done by real users
8. Mac or PC
10. What you do with a computer mouse
13. An action that causes something to happen
15. An early sample or model of something
16. Where actors find their lines or a set of steps for a computer to follow
18. A way to give information to a computer or the opposite of output
21. A placeholder for information that can change

[Answers on Page 10 of Lesson Plan](#)

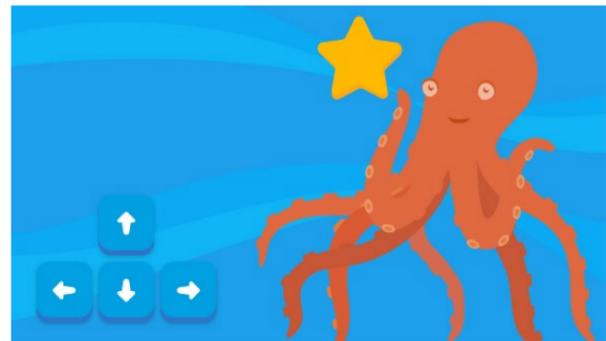
Get to Know Scratch!

Step 5: Use the link below to take a self-guided tour of the Scratch interface.

Click on **Tutorials** in the **Navigation Bar** to explore and test all of the Scratch interface's tools and functions.



Below are a few great tutorials to start out.



Make a Chase Game



Animate a Character



Make a Clicker Game



Use Arrow Keys

Fun Fact

There are more than 1,000 programming languages.

Code an Action!

Step 6: Take what you learned from the tutorials and try coding your own action using different blocks.

Types of Blocks

Motion Block

Controls a sprite's movement

Looks Block

Controls how a sprite looks

Sound Block

Controls sound in project

Events Block

Sets rules for events to occur

Control Block

Controls script reactions based on user input

Sensing Block

Creates reactions based on variables such as distance or color

Operators Block

Creates controllable elements based on conditions being met

Variables Block

Holds variables and lists such as scores

My Blocks

Custom block code creator



The Challenge!

Your challenge is to create and code an original animation or game using the Scratch platform. The guidelines on Page 13 outline the required components and factors to keep in mind when building your Scratch animation or game.

Are you up to the challenge?

Step 7: Read the challenge guidelines!

Make sure you understand all elements of the challenge before you get started.

Helpful Tips and Important Reminders

1. **Practice:** It may take some time to understand the exact workings of Scratch. Make sure you complete the previous steps and try out all the features before creating your animation or game.
2. **Plan:** Decide on what you want the plot and outcome of your animation or game to be. Write out all your characters and what you would like your storyline to be. This will make it easier to code each action of your storyline.
3. **Crowd Source:** Use your family for ideas. They can be a great audience to ensure you have met all the guidelines and can give great advice to help you improve your product.

Challenge Guidelines

Original Characters - Your project should include original characters that fit into your storyline. The character should have a name and introduce themselves in the beginning of your project. Use the costume feature to aid in originality.



Entertainment Factor - Your project should be a new and exciting idea. Use your family to help gauge the entertainment factor.

Complex Coding - Your project must include Move Blocks, Sounds Blocks and Events Blocks. For additional complexity create a code block that contains all 3 kinds of blocks.



Product Creativity - Your project should offer dynamic components of game design or animation that capture the user's interest.

Product Completion - Your project is fully complete when all portions of the game or animation have a beginning, middle, and end with a continuous flow.



Product Effectiveness - Your project performs the intended actions and does not need a lot of support or instruction in order for it to work.

Want an extra challenge?

Try adding an **If Block** and **When Block** to your project.

Word Search

Want to complete the puzzle on your computer or tablet?
Use the highlighter tool in your PDF reader to highlight the words.

F K R C O L U X J Y I V R U J G X J G A C O Y I K
 I C C C J Z D Y N V C M Z H C S N W K P A F W L
 O V D E R J U E J W I W V Z V G H Q E P B X M J Y
 J M D X J E R Z F E R S T E M V V A D I E P W X A
 G R E H Y Q A M X I G X P J K N G W L I M I D A J
 Y Y B F M T Z T H T N L V R M N Z Y A L L J J X Q
 F N U O S M O I I H U E X E O F T X D B E J C C Z
 M C G D H R U Q Y V O A Q A S T S E T L M N Z Z C
 A U Y G Q J M L J S I I I F X Y O E I A Z E G X S
 V B F H W I V J T Z U T M Y E C G T C L B M D E B
 Q M T V U M S O E N K P Y P B J U T Y P P P V C H
 G J W O O T Z A S U F J J N L Q C I B P K A H L H
 A F I S N C I N T Y X F L U B E O G B E E T P E E
 M Z Y A I L Q R J S O Z I H H R M J W D V H B Y U
 E T A N J E I K N U S E R N A M E E D Q C I M X S
 O T Y A V B G N L I E D S K T A T C N Q J Z N Y Q
 A C P L X C Z W E O K R W Q R E D H S T Q E U B S
 A J S Y Z B T N X V L T L Q V C R V Y F N O E W I
 H E H Z F I N P X I D E A T E S F F Y U V T X M B
 M S D E F N W R A O Z R A C S G R U A X I C K C U
 B E V F L A V O J H S X M P Z D Q X I C D N M N B
 P R T T B R Y C P R S P C O D M E E L R E A O P A
 I V F L V Y P E A N I M A T I O N A R V H W A Y H
 O E Q U U X S S M B L M C J J U H B H Z O V T Q F
 Z R I K A B H S N K S C R A T C H R L X E R S F A

Word Bank

Binary	Empathize	Implement	Game
STEM	Define	Challenge	Interface
Creativity	Ideate	Scratch	Debug
Animation	Prototype	Process	Online
Server	Test	Analyze	Username

[Answers on Page 11 of Lesson Plan](#)

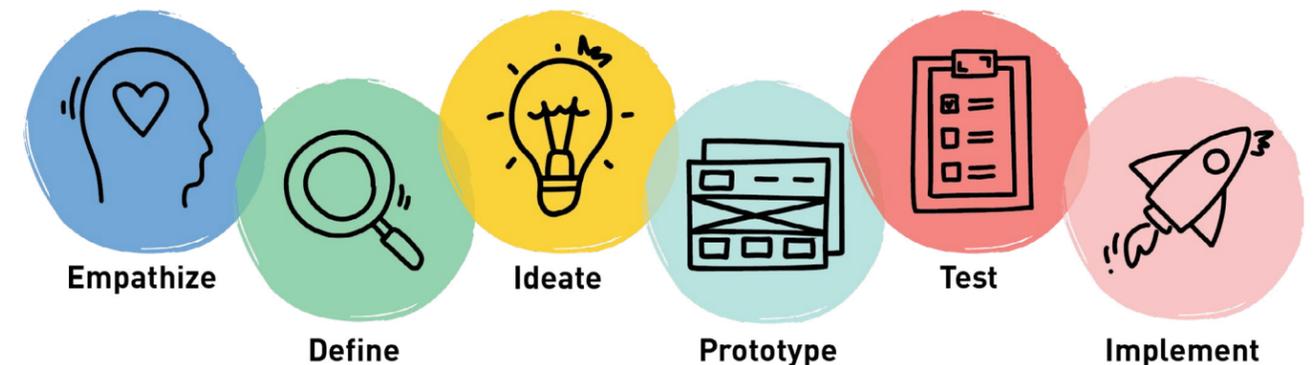
Design Thinking Process

You may already have several animation and game ideas bursting out of your head, but all major projects first require some careful, structured planning before the work begins!

To create the best game possible, you need to first create a plan for how you will build it. To help you create your plan, let's learn about the Design Thinking Process. This is the same process that software engineers and computer scientists use when they come up with a design and are coding their own machines, systems, or structures. The steps of this process will help you complete the challenge.

Step 8: Watch the video and read the explanations on Pages 16 and 17 to learn about the Design Thinking Process.

Click on the picture below to watch a video to learn about the Design Thinking Process.



Design Thinking Process



Empathize: I see a problem. How do I fix it?

Research the people who will benefit from the solution you are creating.

- What problems are they currently encountering?
- What do they need?

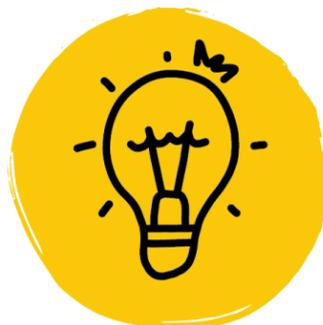
Try to put yourself in the shoes of the person the problem is affecting to explore how these users currently think, act, work, and feel.

Define: I learned more about the problem. How do I solve it?

Combine all of the information you collected as you empathized with your users.

- What trends do you notice across multiple experiences or multiple users?

Review this information and define the exact problems your users are experiencing based on the trends you identified. Put together the information you gathered, to form a one-sentence description of the problem you're solving



Ideate: I have a lot of options. Which path do I choose?

Brainstorm! Create a list of ways that the users' problems might be solved. Think outside the box; no idea is too far-fetched!

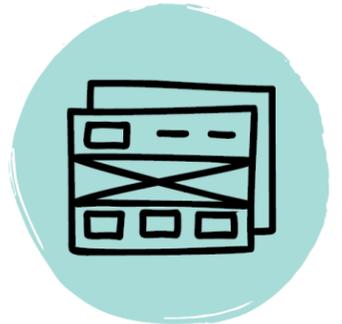
This is a major brainstorming session with the goal of capturing the largest quantity of potential ideas as possible.

Document all ideas at this phase without evaluating whether or not the idea would work.

Prototype: I found a solution. How do I create or tweak it?

Select the few most promising ideas from the ideation session and begin to build a rough sketch of what they might look like and how they could function.

Think through how the product solves the problem and ensure that this is captured in the prototype. You want to experiment with a solution. Then go to the next step and test. You may come back and repeat this step many times, if your test doesn't go the way you want it to.



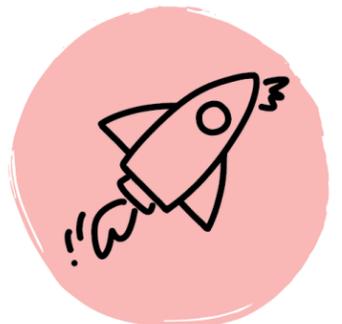
Test: I have a new tool to solve the problem. How well does it work?

Try out your prototype! If your prototype doesn't work the way you want it to, you may need to go back a step and build another prototype to test. Return to the prototyping phase to refine the product. Continue prototyping and testing until the product effectively and efficiently solves the problem.



Implement: I finished my tool. How do I share it?

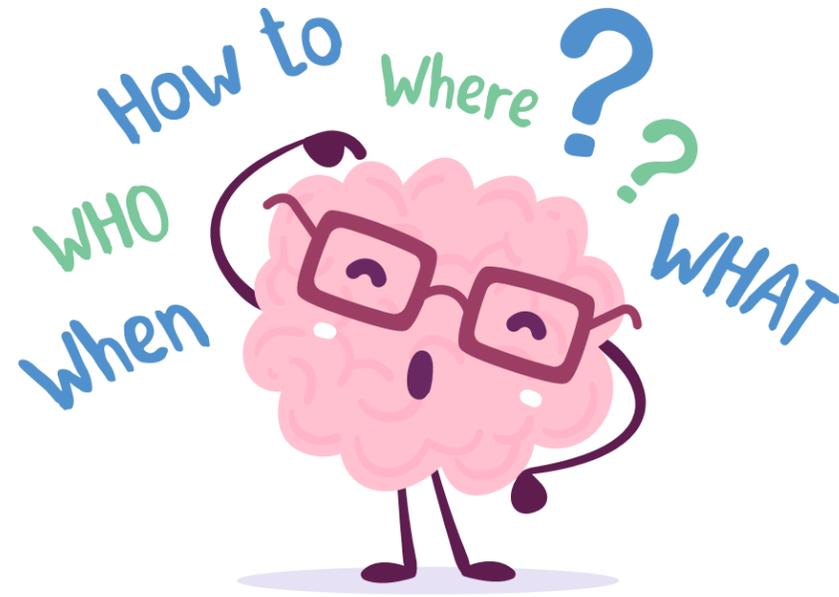
Once the prototyping and testing has resulted in an end product that solves the problem, you have the final product to share with other users.



Fun Fact

Ada Lovelace is often called the "first computer programmer" for writing an algorithm for a computing machine in the mid-1800s. She was an English mathematician and the daughter of the poet Lord Byron.

Let's Get Started!



Now you are ready to begin the challenge!

Bringing an idea from something you brainstormed to a living reality is harder than it sounds.

It requires a deep understanding of the problem you want to address (empathy), anticipating and overcoming obstacles, working effectively with different people who hold different viewpoints and priorities, and ultimately making compromises that reflect the reality of the world you are working in.

The Design Thinking Process can help you along this journey by helping you organize your thoughts and map out the stages of the work.

Step 9: Determine the problem that you are trying to solve with your animation or game. Use the first two design thinking steps to help you identify and define your problem.

Empathize

Step one of the Design Thinking Process requires you to empathize with the needs of whoever your game is designed to benefit.

- What experience do you hope a person playing your game will have?
- What is the key goal your game will achieve? Who do you think will play your game and why? What are their preferences?

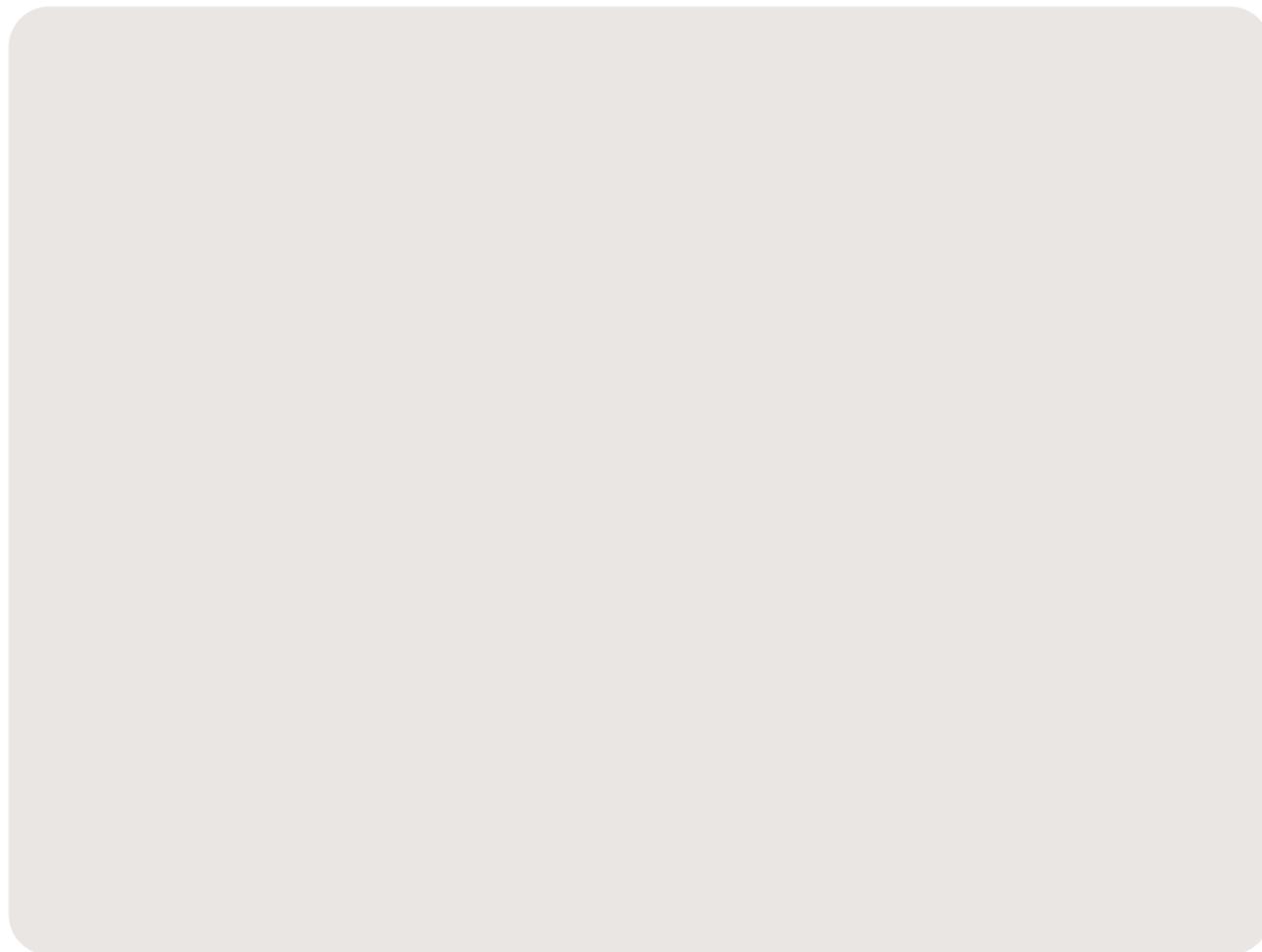
In the space below, write your initial brainstorming thoughts about the experience that your game is attempting to create.

Define

Defining is all about breaking down the end goal itself and beginning to map out a pathway to achieving it. This is not where you start designing—it is where you make sure that you fully understand the work and anticipate the constraints that you might face. Imagine what kinds of challenges and constraints you might face as you create your game.

Write thoughts to the following questions to further define your design process.

- What information are you likely to need?
- What are the realities or constraints of the platform that you might need to consider?
- What actions are possible and what aren't?
- What time constraints do you have?



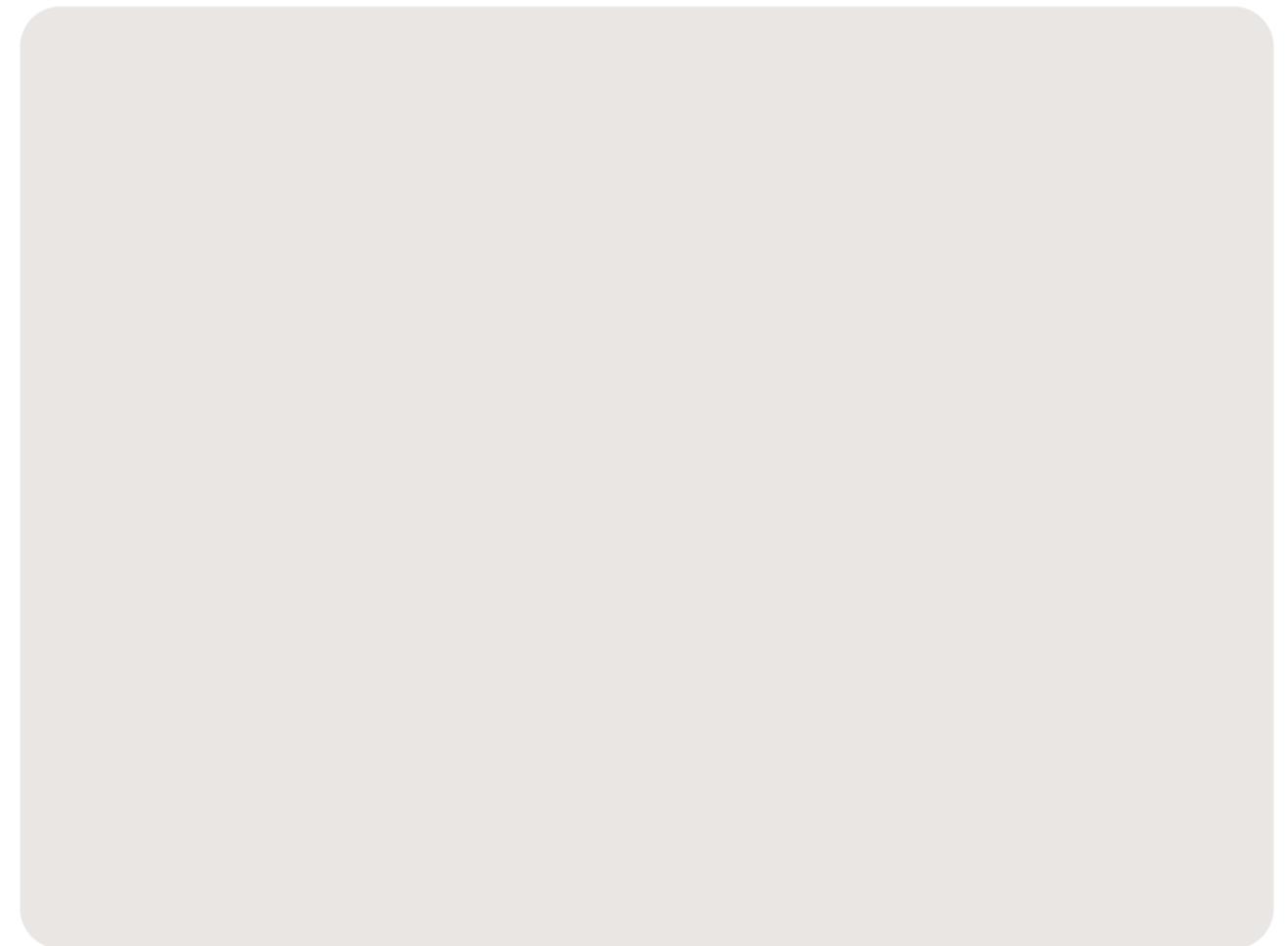
Time to Design!

Step 10: Plan out and design your animation or game.

Use the next two design thinking steps to help you plan out your design.

Ideate

This is the fun, creative part, where you brainstorm ideas for a solution. As you think about possible solutions and approaches, remember that no idea is too far out there. The ideate stage of design thinking should have an *anything goes* approach. **Take some time to write all your wild ideas down, then go through all ideas and decide which one you will move forward with.**



Prototype

In this stage, the goal is to create a pathway that will lead to a more clearly defined end-product, what you will from now on call your prototype.

How do you narrow down such a big list of ideas and insights into a single prototype?

Review your ideation list and choose two potential prototype ideas that could effectively create the desired experience.

The first idea will be the one you develop and the second will serve as a backup idea if you encounter impossible challenges with your initial prototype.



Idea #1 _____

Idea #2 _____

Remember, as you work towards this prototype, you may encounter difficulties that require you to revisit your shortlist again and revise your plans. That is expected and is all part of the Design Thinking Process. First, you want to plan out what your game will be like.

Planning can be done in different ways. You may be highly visual and need a sketch, flowchart, or a concept map illustrating your plan, or need lists and descriptions. Decide what your plan for making the prototype will look like and get to work! Once you have your plans all set, you can start the coding process.

Fun Fact

The coding language Python is named after the 1970s British sketch comedy TV show Monty Python's Flying Circus.

Test and Improve!

Step 11: Test your animation or game to see if you have successfully completed the challenge.

Test

Test your game. Does it solve your problem and accomplish the final goal you set in the *Define* stage? If it doesn't, how can you adjust your game and make it better? Test and try out different codes to see what works and continue making changes until you successfully solve your problem.

Remember, it might not work the first time, and that is a good thing! An important part of the invention process is failure, because without it we won't learn how things really work. As you make mistakes and try different things, you will get a better understanding of how each of the parts of your game work, which will allow you to create the best game possible.



Notes for future improvements:



Congratulations!

If your animation or game successfully solves your problem, you have completed the challenge! Amazing job creating and coding your own animation or game!

Time to Share!

Implement

Time to take the final step of the Design Thinking Process and share your newly created game with your family and friends!

We encourage you to share your creation and showcase your hard work with Envision by WorldStrides on social media. Connect with us on Instagram, Twitter, and Facebook by tagging Envision and using hashtag #EnvisionFamilyLabs.



Connect with Us



@envisionexperience



@envisionexp



Envision Experience

Tell Us What You Think

We hope you enjoyed this Family Lab! We encourage you to share it with friends and family. We would love to get your feedback!

[Click here to complete the survey!](#)

Please complete this optional two-minute survey and provide us your thoughts. Thank you!

More Coding!

Did you have fun completing this challenge and are you interested in another challenge? If so, we have a challenge for you!

Brainstorm what other projects you could create with Scratch. Use Scratch to create one of these projects. Ideas for additional Scratch challenges can be found [here](#).

Check out the other coding websites listed below and try your hand at more coding!

More Student Coding Sites:

Blockly Games - <https://blockly.games/>

Code Combat - <https://codecombat.com/>

Code.Org Projects - <https://studio.code.org/projects/public>

Code Monster - <http://www.crunchzilla.com/code-monster>

Tynker - <https://www.tynker.com/>

Technology and Coding Resources/News:

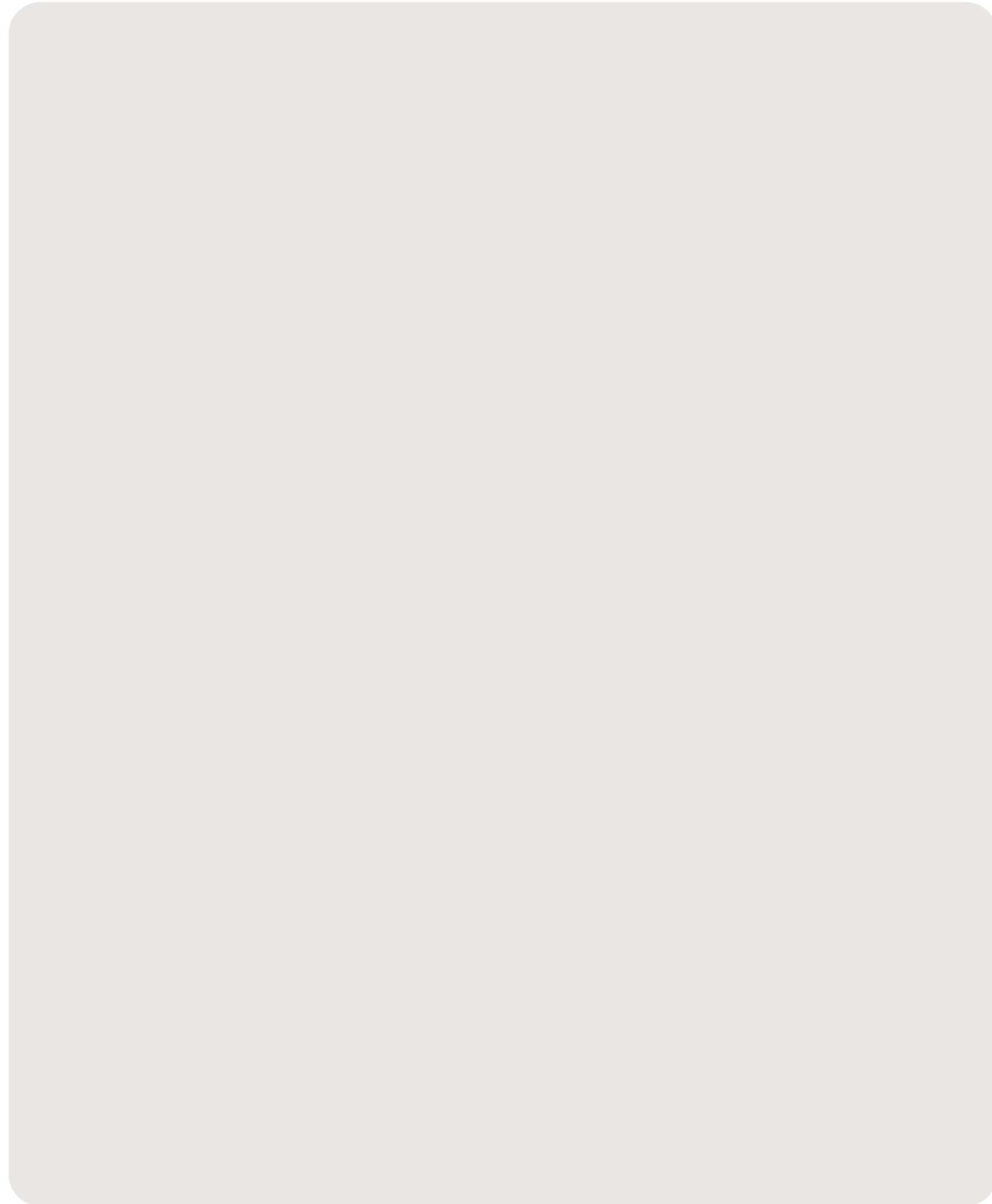
Code.Org Video Library - <https://code.org/educate/resources/videos>

Ars Technica - <https://arstechnica.com/>

The Next Web - <https://thenextweb.com/>

Tech Crunch - <https://techcrunch.com/>

Notes



Resource Page

Scratch Resources

Scratch FAQs

<https://scratch.mit.edu/info/faq>

Scratch Parents Page

<https://scratch.mit.edu/parents>

Scratch Getting Started Tutorial

<https://scratch.mit.edu/projects/editor/?tutorial=getStarted>

Scratch Ideas Page

<https://scratch.mit.edu/ideas>

Scratch Wiki

<https://en.scratch-wiki.info/>

Videos

Scratch Overview Video

ScratchEd. (2013 May 6). *Scratch Overview*. [Video]. Vimeo. <https://vimeo.com/65583694>

TED Talk Video

Resnick, Mitch. (November 2012). *Let's teach kids to code* [Video]. https://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code?utm_campaign=tedsread&utm_medium=referral&utm_source=tedcomshare

Design Thinking Process Video

Crash Course Kids. (2015 May 29). *The Engineering Process: Crash Course Kids #12.2* [Video]. YouTube. <https://www.youtube.com/watch?v=fxJWin195kU&list=PLhz12vamH0nZ4ZDC0dS6C9HRN5Qrm0jHO&index=2>

