Fruit Fighter
Teacher Guide

Summary

Coding skill level:	**Advanced**
Recommended grade level:	**Grades 6-12 (U.S.), Years 7-13 (U.K.)**
Time required:	**50 minutes**
Number of modules:	**1 module**
Coding Language:	**JavaScript**

Teacher Guide Outline

Welcome!

Who's ready to use p5.js and AI (artificial intelligence) to create a fun fruit slicing game? In this lesson, students will create an interactive project where they'll slice fruit on their screen by moving their right pointer finger! Students will begin by completing the "Getting Started" activity where they'll answer open-ended questions. Next, they'll move on to the step-by-step tutorial where they will use JavaScript to create a fun fruit slicing game called "Fruit Fighter." Students are provided code to get started, but are encouraged to modify their code to make it their own. The tutorial also includes a completed sample project and ideas on how to enhance their game. Time permitting, students will reflect on what they learned by answering the discussion questions in the "Extended Activities" section of this teacher guide. By the end of the lesson, students will have combined p5.js, AI, and their creativity to make a webcam-powered game that they can play solo or show to friends! **Important:** Students will need a working webcam in order to complete the project.

How to Prepare

This activity is designed for self-directed learning. Your role will be to help students individually and facilitate as students complete the coding activities on their own. The best way to prepare is to:

1. **Familiarize yourself with the material.** After selecting your Tynker lesson (Fruit Fighter), read through the teacher guide and complete the activity before assigning it to students. This will allow you to troubleshoot anything in advance and plan for potential questions from your students.
2. **Sign up for a teacher account.** Creating a free teacher account will allow you to access teacher guides, answer keys, and tons of additional resources. You'll also be able to create free accounts for your students, monitor their progress, and see their projects.
3. **Create student accounts.** From your teacher account, you can easily create free student accounts for all your students. This will allow them to save their projects and progress, so they can continue coding when they get home!
4. **(Optional) Print certificates to hand out.** While signed in to your Tynker teacher account, you can print certificates by clicking on a classroom from your teacher dashboard, clicking the "Gradebook" tab, going to "Hour of Code," and clicking the "Print All Certificates" button. This will only print certificates for student accounts assigned to the selected classroom.

Activity

Overview

Objectives

Students will...

- Apply coding concepts to create an interactive game using JavaScript
- Demonstrate an understanding of variables and functions
- Apply p5.js commands to draw on the canvas
- Use an AI hand recognition model in p5.js

Materials

- **For web:** Computers, laptops, or Chromebooks (1 per student)

Vocabulary

- **Artificial Intelligence:** The idea of machines (such as computers) demonstrating intelligence, similar to how our brain works
- **Machine learning:** When computers learn from examples or experience to make decisions or recognize patterns on their own.
- **Model:** A program that has been trained on data to make predictions or decisions
- **JavaScript:** JavaScript is a text-based programming language that is widely used in web development
- **p5.js:** A JavaScript library used for creating interactive graphics and animations in a web browser
- **Code:** The language that tells a computer what to do
- **Sequence:** The order in which steps or events happen
- **Command:** A specific action or instruction that tells the computer to do something
- **Loop:** An action that repeats one or more commands over and over
- **Function:** A set of known actions that the computer can perform
- **Variable:** A memory location that is used to store data in a program. The value of variables can vary, or change

U.S. Standards

- **CCSS-ELA:** RI.6.4, RI.6.7, SL.6.1, RI.7.4, SL.7.1, SL.8.1, RI.8.4, RI.9-10.5
- **CCSS-Math:** MP.1
- **CSTA:** 2-AP-11, 2-AP-13, 2-AP-16, 3A-AP-17, 3B-AP-11
- **CS CA:** 6-8.AP.13, 6-8.AP.16, 6-8.AP.17, 9-12.AP.12, 9-12.AP.16
- **ISTE**: 1.1.c, 1.1.d, 1.4.d, 1.5.c, 1.5.d, 1.6.b

U.K. Standards

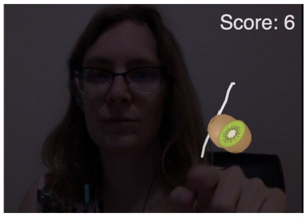
National Curriculum in England (computing):

- **Key Stage 3 (Years 7-9)**
    - Create, reuse, revise and repurpose digital artefacts for a given audience, with attention to trustworthiness, design and usability

   - Understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct, and know how to report concerns
- **Key Stage 4 (Years 10+)**
   - Develop their capability, creativity and knowledge in computer science, digital media and information technology
   - Develop and apply their analytic, problem-solving, design, and computational thinking skills
   - Understand how changes in technology affect safety, including new ways to protect their online privacy and identity, and how to report a range of concerns

Getting Started **(20 minutes)**

- Show students the completed sample project by clicking this image in the tutorial:



- Next, help students think about Fruit Fighter's gameplay by asking them…
   - What did you observe when the user's finger touched the fruit? (Answer: the fruit sliced open.) What other gameplay features did you observe? (Suggested answer: the game keeps score, fruit appears one at a time, there's a visible line in the game to help the user see where their hand is located on the screen, etc.)
   - What ideas do you have for why the program is able to detect that specific part of the hand? (Suggested answer: The program is using AI hand recognition.)
   - What are some gameplay features you can add to further enhance the game? (Suggested answer: Customize the fruit's behavior, have multiple fruits appear at once, etc.)

DIY Module **(30 minutes)**

This lesson has one DIY (do-it-yourself) module. Facilitate as students complete the Fruit Fighter module on their own:

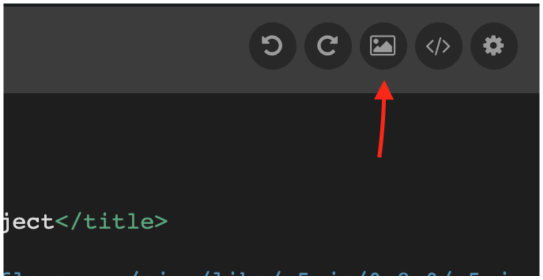
Fruit Fighter

- In this DIY, students will use the text code editor to create an interactive game that uses an AI hand recognition model to slice fruit by moving your finger.
- Note that this project utilizes an AI model. This model has already been "trained" to recognize the human body by being shown many photos of people and data

about what's in the photos. Now the model is ready to be used in the project, where it can "find" your students' hands using their webcam.

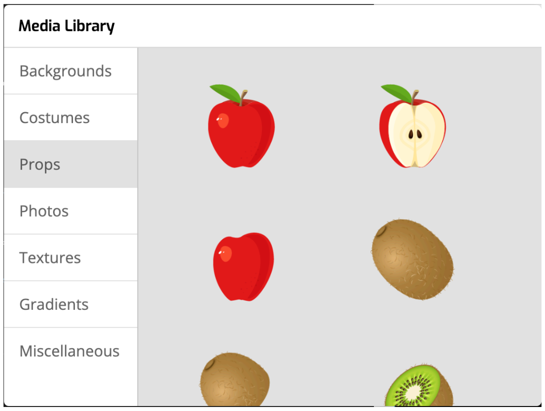
- Coding activities include defining variables, setting up the canvas, using functions to load fruit images, and programming the game to look for and follow the user's right pointer finger.
- To add image assets from the text code editor's Media Library, ask students to follow these steps:

1. Navigate to the Media Library:

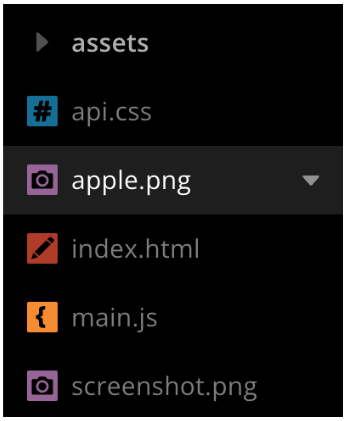


```
ject</title>
```

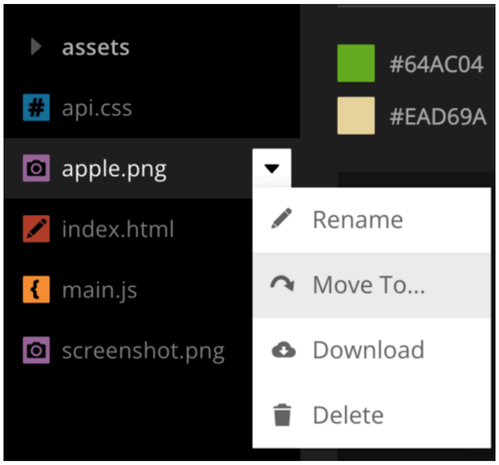
2. In the window that appears, select the "Props" category:



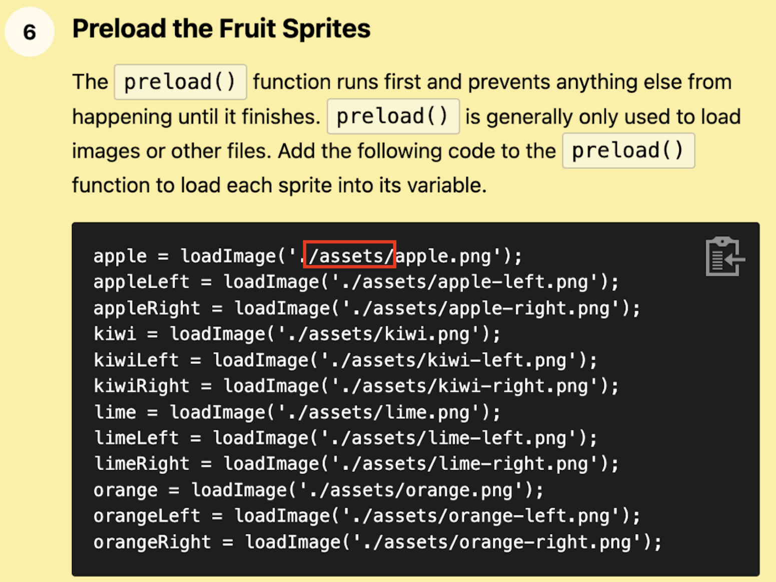
3. Click on each asset to add it. You will see the assets appear in the sidebar. Click the icon to close the Media Library window when you are done.



4. To keep your project more organized, click on the arrow next to the image, select Move To, and select the "assets" folder:



- Step 6 of the tutorial assumes students are using the provided assets folder to store their images. If students are using images outside of the assets folder, they'll need to change the code.



6 **Preload the Fruit Sprites**

The `preload()` function runs first and prevents anything else from happening until it finishes. `preload()` is generally only used to load images or other files. Add the following code to the `preload()` function to load each sprite into its variable.

```
apple = loadImage('./assets/apple.png');
appleLeft = loadImage('./assets/apple-left.png');
appleRight = loadImage('./assets/apple-right.png');
kiwi = loadImage('./assets/kiwi.png');
kiwiLeft = loadImage('./assets/kiwi-left.png');
kiwiRight = loadImage('./assets/kiwi-right.png');
lime = loadImage('./assets/lime.png');
limeLeft = loadImage('./assets/lime-left.png');
limeRight = loadImage('./assets/lime-right.png');
orange = loadImage('./assets/orange.png');
orangeLeft = loadImage('./assets/orange-left.png');
orangeRight = loadImage('./assets/orange-right.png');
```

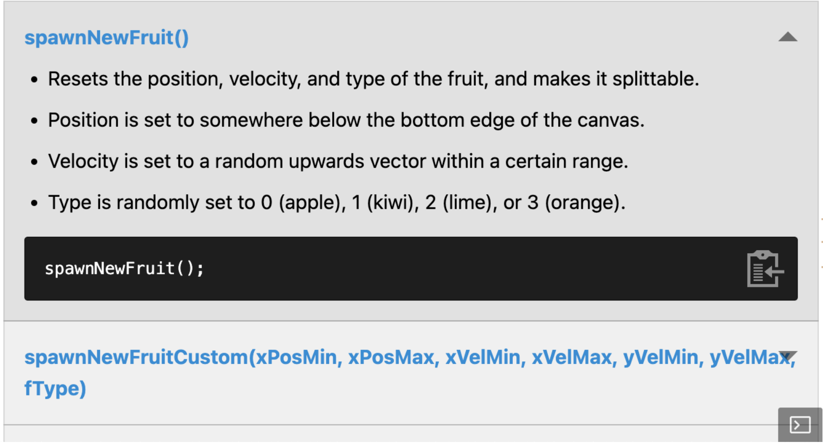
- Do students want to increase the level of difficulty? Give a hint: Encourage them to increase the frame rate which will make their game run faster: `frameRate(60);`
- Are students struggling to change the color of the background? Give a hint: Direct their attention to this part of their code:

```
background(0);
```

- Time permitting, encourage students to gain a better understanding of the impact of how changing variables can affect their game by directing their attention to this part of their code:

```
/**
 * Customize these 3 numerical variables to change how the game behaves
 * gravity           How fast the fruit falls towards the bottom of the canvas
 * shrinkFactor      How much to shrink the fruit images (between 0 and 1)
 * bladeLength       How many segments the blade can have
 */
const gravity = 0.3;
const fruitRadius = 20;
const splitShrinkFactor = 0.75;
const bladeLength = 12;
```

  Then, encourage students to make one small change at a time and note what they observe.
- Do students need further guidance? Direct their attention to the "Helper Functions Reference Guide" section, which includes code snippets and descriptions. Here's a sample:



spawnNewFruit()

- Resets the position, velocity, and type of the fruit, and makes it splittable.
- Position is set to somewhere below the bottom edge of the canvas.
- Velocity is set to a random upwards vector within a certain range.
- Type is randomly set to 0 (apple), 1 (kiwi), 2 (lime), or 3 (orange).

```
spawnNewFruit();
```

spawnNewFruitCustom(xPosMin, xPosMax, xVelMin, xVelMax, yVelMin, yVelMax, fType)

- If students want to learn more about p5.js, ask them to read this fun blog titled "Explore the Power of p5.js: A Beginner's Guide." Here's the link: https://www.tynker.com/blog/exploring-the-power-of-p5-js-a-beginners-guide/
- Did students finish early? Direct their attention to the last step of the tutorial where they're encouraged to experiment with their code and customize their game. Tip: Encourage students to develop their project a little at a time, testing frequently. This will help them figure out where their project is/isn't working as expected.
- Time permitting, encourage students to explain and show how they modified their Fruit Fighter game. What unique features did they add? What are some obstacles and successes they encountered?

Extended Activities

Reflect on today's coding adventure by pairing up students and asking them to discuss the following questions:

- How does this project use AI? (Answer: This project uses a hand-tracking model which is a form of AI designed to recognize and track hand movements in real-time.)
- Think of another classic game (e.g. tic-tac-toe, pong, chess, pictionary, etc.). Now think of how you could add an AI hand-tracking component to the game. What ideas do you have? (Answer example: In pong, you could control the paddles on the screen by moving your hands up and down.)
- What questions do you have about AI? Can you think of other examples of how AI could be utilized in games? (Answers will vary.)

Going Beyond an Hour

If your students enjoyed an Hour of Code, they're sure to enjoy the rest of what Tynker has to offer! Tynker offers a complete premium solution for schools to teach computer science. Over 400 hours of lessons are available to take K-8 students from block coding to advanced text coding. We offer tons of resources for teachers, including comprehensive guides, free webinars, and a forum to connect with other educators.

More Hour of Code Activities

Tynker offers many other tutorials for the Hour of Code, including STEM Hour of Code lessons that you can integrate into the subjects you already teach. Check out the main Tynker Hour of Code page to see all the tutorials!

More p5.js Activities

In Tynker's Intro to Programming and Art course, students will learn foundational concepts of art and computer science as they're introduced to p5.js! Here's a link to the course:
https://www.tynker.com/high-school/coding-curriculum/intro-cs-art/1-unit-1-introduction-to-p5-js

Do More with Tynker

With Tynker, kids don't just acquire programming skills--they explore the world of possibilities that coding opens up. Tynker has several interest-driven learning paths that make coding fun, both inside and outside the classroom:

- **Even More Free AI Tutorials** — https://www.tynker.com/blog/tynker-toolbox-the-ai-blocks/

- **Artificial Intelligence 101** — This AI course uses Tynker's fun and friendly code blocks. Webcam projects powered by AI put your students at the center of the action! Create interactive Snapchat-style costumes, hand-tracking apps, activity games, and so much more. Along the way, students understand how AI models are driving their amazing inventions behind the scenes. Students even explore natural language processing (NLP), the technology used by modern chatbots like ChatGPT. Find out more here: https://www.tynker.com/k8-school/coding-curriculum/artificial-intelligence-101
- **Artificial Intelligence 401** — This JavaScript course is perfect for experienced students looking to implement AI in their programs. Learn how to use MediaPipe, an AI solution from Google, and explore topics like neural networks, supervised learning, and much more. More than that, students dive into AI ethics and consider how AI might impact our future. Find out more here: https://www.tynker.com/k8-school/coding-curriculum/artificial-intelligence-javascript-401
- **Coding and Game Design:** Your students can use Tynker Workshop, a powerful tool for crafting original programs to make games, stories, animations, and other projects. They can even share their work with other kids in the Tynker Community.
- **Robotics:** Tynker integrates with connected toys, including Lego WeDo robotics kits, so kids can see their code come to life.
- **Minecraft:** Tynker integrates with Minecraft so your students can learn coding through a game they love. Tynker offers skin and texture editing, as well as a custom Mod Workshop that lets kids try their original code in Minecraft.

Tynker for Schools

Used in over 90,000 schools, our award-winning platform has flexible plans to meet your classroom, school, or district needs. All solutions include:

- Grade-specific courses that teach visual coding, JavaScript, Python, robotics and drones
- A library of NGSS and Common Core compliant STEM courses that are great for project-based learning
- Automatic assessment and mastery charts for whole schools and individual classes and students
- Easy classroom management with Google Classroom and Clever integration
- Professional training, free webinars and other teacher training resources

Need help getting Tynker started at your school? Contact us to learn more about teaching programming at your school with Tynker!

Help

Need help? Below you'll find answers to frequently asked questions about using Fruit Fighter.

What is Hour of Code?

The Hour of Code is a global learning event in which schools and other organizations set aside an hour to teach coding. No prior coding experience from you or your students is needed! The event is held every December during Computer Science Education Week. You can also organize an Hour of Code year-round. The goal of the Hour of Code is to expand access to computer science education for people of all backgrounds. Learning computer science helps students develop logic and creativity, and prepares them for the changing demands of the 21st century. Tynker has been a leading provider of lessons for the Hour of Code since the event began in 2013. Since then, over 100 million students from 180 countries have finished an Hour of Code.

How can Tynker help me manage my Hour of Code?

Tynker has several free features for registered teachers that will help you manage your Hour of Code. If you set your students up with a Tynker classroom, you will be able to track their progress and print Hour of Code completion certificates for them to keep.

How do I prepare for Fruit Fighter?

Familiarize yourself with the material. After selecting your Tynker lesson (Fruit Fighter), read through the teacher guide and complete the activity before assigning it to students. This will allow you to troubleshoot anything in advance and plan for potential questions from your students.

How does this project use AI?

This project utilizes a pre-trained AI body tracking model. The AI model has been trained on a large dataset of images featuring the human body. When your students interact with the project, the model analyzes the incoming video feed to identify and follow the movements of the user's body landmarks, specifically of the hands, providing accurate and instantaneous tracking. Using the provided AI JavaScript library, students can apply AI technology to build interactive, sophisticated projects. Let's see what your students will create!

Who is this activity for?

Fruit Fighter is intended for students in grades 6-12 (U.S.) and Years 7-13 (U.K.) with some coding experience.

What devices do I need?

Computers, laptops, or Chromebooks (1 per student) with an internet connection **and** a working webcam.

What will my students learn?

Hour of AI
Fruit Fighter

Students will reinforce JavaScript syntax as they create a fun Fruit Fighter project! Students are also encouraged to expand on their project by customizing their game. In this process, students will develop debugging and logical reasoning skills.

What does the tutorial include?

The tutorial includes several features and resources to help your students get started! Here’s what you will find:

- A completed sample project to show your students what their project could look like
- Step-by-step directions to help your students create a Fruit Fighter game using JavaScript
- Ideas on how to customize the game

How can I help my students understand their code?

- Make sure students are reading the tutorial carefully. It describes each step of the Fruit Fighter project.
- By the end of the project, their code could look similar to this:

```
// Initialize the hand recognizer and game settings
const hands = new HandRecognizer();

let fruitPos;
let fruitVel;
let fruitColor;
let fruitCanSplit = true;

let splitPosLeft;
let splitPosRight;
let splitVelLeft;
let splitVelRight;

let target = {x: 0, y: 0};
let bladePositions = [];
let score = 0;

/**
 * Customize these 4 numerical variables to change how the game
behaves
 * gravity                How fast the fruit falls towards the bottom of
the canvas
 * fruitRadius            How big the fruits are
 * splitShrinkFactor      How much smaller the split pieces are (should
be less than 1)
 * bladeLength            How many segments the blade can have
 */
const gravity = 0.3;
const fruitRadius = 20;
const splitShrinkFactor = 0.75;
const bladeLength = 12;

function setup() {
    createCanvas(480, 360);
    frameRate(30);

    spawnNewFruit();

    // Show video but make it somewhat transparent
    hands.showVideo(35);
}

function draw() {
    background(0);

    // Get predictions from the hand recognizer
    hands.predict();

    if (hands.rightHandVisible()) {
        target = hands.getRightLandmark(Landmarks.Pointer_Tip);
    }

    // Update and draw the blade
    bladePositions.push(target);
    if (bladePositions.length > bladeLength) {
        bladePositions.shift();
    }
    renderBlade(bladePositions, [255, 255, 255], 4);

    updateFruit();

    // Check if this fruit got sliced
    if (detectSlice()) {
        splitFruit(1, -3);
    }

    renderFruit();

    // Launch a new fruit when the old one is fully offscreen
    if (fruitIsOffScreen()) {
        spawnNewFruit();
    }

    renderScore(score, 255, 32, RIGHT, width-20, 40);
}
```

Note: This is an open-ended project. Students are encouraged to experiment with their code, try the bonus activities, and expand on the provided code samples.

How can I contact the Tynker support team?

If you have any issues or questions, send us an email at support@tynker.com.