

## Learn Real Programming Skills Through Play.

In the 21st century, understanding that machines are driven by rules and logic is a superpower. At Lando, we believe the best way to demystify this technology is to open the hood and let students drive. We teach the code that powers the world by building games that are fun to play and exciting to create.

### This Semester's Theme: Critter Capital

This semester, our young developers will focus on "guiding critters across the big city." Students will blend logic with creativity to build games where animals navigate their landscapes.



## What Your Child Will Learn.

We cover fundamental computer science concepts used in real-world software development and engineering, but we introduce them in a way that feels like play.

- **Variables** Think of these as "digital boxes" that hold information that changes, like a player's Score or Health Points.
- **Events** Cause and effect coding—"When I press the Spacebar, the character jumps."
- **Conditionals** The logic of decision making—"IF the character touches a wall, THEN they bounce back."
- **Loops** How to make actions repeat automatically without rewriting code.
- **Debugging** The critical skill of finding mistakes and solving problems efficiently.

## How the Class Works.

1. **The Interactive Spark (10 Mins)** Every session begins with a short, interactive lesson. Students work together with friends to answer questions and unlock the core concept of the day.

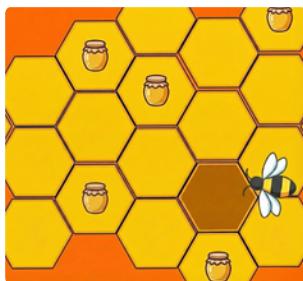
2. **The Challenge Phase (Self-Paced)** Students move into "Build Mode," tackling 6–10 challenges that increase in difficulty.

- **Personalized Pace** We meet students where they are. A beginner might spend 40 minutes mastering the fundamentals of a challenge, while a proficient coder might sprint through it in 10 minutes and move to advanced extensions.
- **Prerequisites** All skill levels are welcome! We simply require a basic reading level to navigate the coding blocks.

3. **Bringing Learning Home** We believe learning shouldn't stop at the classroom door. After every class, parents receive a "Play & Share" email. This includes a link to the actual game your child built that day, giving you a chance to play it together and let them explain how it works!



## Weekly Schedule.



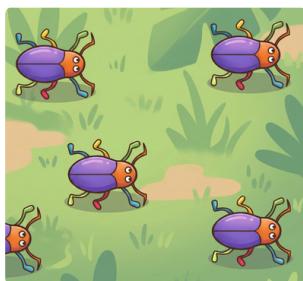
### Class 1 Bee Harvesting

Students begin the semester with a playful project that reinforces core skills from Semester 1. They program a bee to fly around the screen and earn points by clicking at just the right moment, strengthening their understanding of motion, timing, and basic conditionals. This opening game builds early confidence and sets the foundation for more complex game logic.

Motion Basics

Costumes

If Conditionals



### Class 2 Clear the Spiders

Students design a maze game where a fox must avoid spiders while navigating home. This class deepens their comfort with collision detection and reinforces how conditionals shape game outcomes. The project introduces simple game difficulty and teaches students to debug movement challenges.

Motion Basics

Costumes

If Conditionals



### Class 3 Froggy

Here, students begin working with x and y coordinates in a more intentional way as they help a frog hop across lily pads. This game strengthens spatial reasoning and prepares students for projects that require controlled jumps, lanes, and direction-based logic.

X Y Coordinates



### Class 4 Treasure Diver

Students design an underwater obstacle course where a diver weaves between rocky walls. This is their first introduction to broadcasting, which lets different sprites communicate with each other. Broadcasting unlocks multi-sprite coordination, a big step forward in game design sophistication.

Motion

Broadcasting

If Conditionals

## Weekly Schedule.



### Class 5 Bug in a Jug

This class introduces sprite duplication and variables as students catch jars falling from the sky to collect bugs. They learn how games create repeating objects and how scores or timers make gameplay engaging, which are essential skills for building more dynamic, replayable games.

Sprite Duplication

Variables

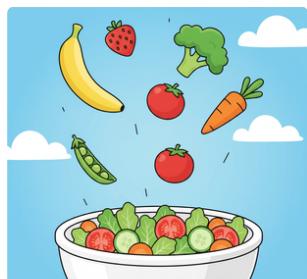


### Class 6 Foxes and Boxes

Students build a classic shell game where a fox hides under a box and the boxes shuffle. This project strengthens logical reasoning as students work with ask-and-answer blocks and broadcasting to manage game interactions. It prepares learners for more advanced input-based games later in the semester.

Ask and Answer

Broadcasting

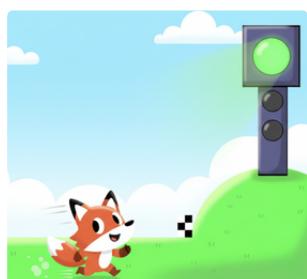


### Class 7 Fruits and Vegetables

Students expand their use of sprite duplication to create two categories of falling objects that must be sorted correctly. Randomization makes every round new, encouraging students to think about pacing, challenge, and balance in game design.

Sprite Duplication

Random Numbers



### Class 8 Traffic Light Racer

In this red-light green-light style game, students use conditionals and loops to create stop-and-go mechanics that respond to a changing traffic signal. This class shows learners how to control game flow and manage ongoing actions, an important stepping stone to advanced game systems.

Variables

Forever Loop

If Conditionals

## Weekly Schedule.



### Class 9 Sprint

Students build a racing game where alternating keystrokes move a character around a track. This introduces more nuanced use of variables and conditionals to create sprint mechanics, helping students see how coding can simulate real-world movement.

[Costumes](#)[Variables](#)[If Conditionals](#)

### Class 10 Shadow Match

In the final class, students design a drag-and-drop matching game that blends spatial thinking and sprite communication. Broadcasting and conditionals allow them to check accuracy and build satisfying feedback loops. This project ties together many of the semester's core skills in a polished, interactive challenge.

[Motion](#)[If Conditionals](#)[Broadcasting](#)