

Beat The Clock

60 minutes (2 periods)

Introduction to Game Design

WHAT THIS LESSON DOES

In this lesson, students will learn to create an engaging game called 'Beat The Clock' using MakeCode Arcade. They will gain hands-on experience in creating game elements, setting up a timer, designing a scoring system, and coding for sprite movement and interaction. This lesson offers a fun and interactive way to learn coding concepts.

WHAT STUDENTS SHOULD BE ABLE TO DO

- Develop skills in creating game elements using MakeCode Arcade.
- Understand how to set up a timer for a game.
- Learn to create a scoring system in a game.
- Gain proficiency in using code blocks for sprite interaction.
- Enhance problem-solving skills by experimenting with different settings and codes.

BEFORE THE LESSON

- Open and test <https://arcade.makecode.com> on class devices so the site is not blocked.
- One device per pupil or pair, charged, with keyboard arrow keys or on-screen joystick ready.
- Have the site on a shared screen if you plan a brief whole-class demo before independent work.

Beat The Clock

HOW THE LESSON RUNS

Introduction

Start a New Project

Set Up Game Elements

Move the Player

Move the Target

Set Up the Timer

Detect the Overlap

Create a Scoring System

Wrap Up

TEACHING MOVES

1. Introduction. Frame the goal in one line: a game where the player chases a target, scores on contact, and races a 60-second clock.
2. Start a New Project. Get every device to arcade.makecode.com and a new project named before anyone adds blocks.
3. Set Up Game Elements. Guide pupils to create both sprites and design them so the player and target are easy to tell apart on screen.
4. Move the Player. Add controller movement, then have each pupil drive the player once to confirm the joystick or arrow keys work.
5. Move the Target. Place the target with a random position at the start so it does not always appear in the same spot.
6. Set Up the Timer. Drop in the countdown block set to 60 and run it once so pupils see the clock before scoring is added.
7. Detect the Overlap. Stress SpriteKind Food on the overlap block and mySprite2 in set position, with X 0 to 160 and Y 0 to 120.
8. Create a Scoring System. Add change score by 1 inside the same overlap handler, then let them play and count hits in 60 seconds.
9. Wrap Up. Invite a few pupils to name one setting they would change next, such as time, speed or sprite design.

WHAT TO WATCH FOR

- Pupils leave the overlap kinds wrong, so player and target never trigger the handler. Pause and check the overlap block reads Player and Food, and that the target sprite is kind Food.
- Pupils set position on the wrong sprite or type fixed numbers, so the target does not jump away. Point them to mySprite2 in set position and the randint ranges 0 to 160 for X and 0 to 120 for Y.
- Pupils add the score block outside the overlap handler, so the score never rises on contact. Show that change score by 1 must sit inside the same onOverlap function as the reposition code.

DIFFERENTIATION

- Emerging: Pair them with a partner and complete one step at a time, checking the player moves before adding the target. Stay with them to match each new block to the step name on screen.
- Developing: Have them explain the overlap handler aloud: what moves, what scores, what the timer does. If the game runs but feels unfair, ask them to adjust only the countdown value and test again.
- Proficient: Challenge them to experiment with countdown length or sprite design while keeping overlap and score working. Ask them to teach another pair how the overlap block relocates the target and adds a point.