

Cherry Collection

60 minutes (2 periods)

Introduction to Game Design

WHAT THIS LESSON DOES

This lesson guides students through creating a game using MakeCode Arcade. They will learn to create a character, move it around the screen, generate food sprites in random locations, and collect them for points. The lesson also includes adding sound effects for a more interactive gaming experience.

WHAT STUDENTS SHOULD BE ABLE TO DO

- Develop skills in using MakeCode Arcade to create a game character and manipulate its movements.
- Understand and apply the concept of sprites in game development.
- Implement random generation of game elements on the screen.
- Apply collision detection to enable interaction between game characters and elements.
- Incorporate sound and scoring system to enhance game interactivity.

BEFORE THE LESSON

- Open <https://arcade.makecode.com> on the school network and confirm it loads past any filter.
- One device per pupil or pair, charged and logged in, with sound on or headphones ready for the collect sound.
- Check pupils can use arrow keys or the on-screen joystick in the Arcade simulator.

Cherry Collection

HOW THE LESSON RUNS

Introduction
Create a new project
Add your player
Move your player
Create the cherries
Random places
Collect the cherries
Get a point
Play a sound
Wrap up

TEACHING MOVES

- 1. Introduction. Frame the goal in one line: a movable character that collects cherries for points and a sound. Keep it short, then move straight to the editor.
- 2. Create a new project. Project the Arcade site. Have every pupil open the link, start a new project and give it a clear name before coding.
- 3. Add your player. Guide them to `sprites.create`, open the grey image box and pick a gallery character. Confirm `SpriteKind` is set to `Player`.
- 4. Move your player. Add controller movement so the sprite responds to the joystick or arrow keys. Pause for a quick move test before the next block.
- 5. Create the cherries. Add `onUpdateInterval` at 2000 ms. Create the second sprite as `SpriteKind.Food` and choose a cherry from the gallery.
- 6. Random places. Inside the interval, set each food sprite to a random screen position so cherries appear in different places, not stacked.
- 7. Collect the cherries. Add `onOverlap` for `Player` and `Food`. Destroy `otherSprite` only. Stress selecting `Food` in the overlap block and swapping out `mySprite`.
- 8. Get a point. In the same overlap handler, add `info.changeScoreBy(1)`. Have them play and watch the score rise on each collect.
- 9. Play a sound. Add the `powerUp` melody in the overlap handler. Confirm device sound is on, then play to hear it on each collect.
- 10. Wrap up. Celebrate the working game. Ask a few pupils to say one skill they used or one change they would try next. Spoken only.

WHAT TO WATCH FOR

- Overlap never fires because the food sprite is still Player kind, or Food was not selected in the overlap block. Pause and check both create blocks and the overlap block show Player with Food before testing again.
- Pupils destroy mySprite (the player) instead of otherSprite, so the character vanishes on first collect. Show the destroy block empty, then place otherSprite in it. Replay once so they see only the cherry disappear.
- Cherries stack in one spot because random position code was skipped or placed outside the interval. Have them open the onUpdateInterval body and confirm the random position sits inside it, then run again.

DIFFERENTIATION

- Emerging: Pair with a partner for the first create and gallery pick; teacher confirms Player kind before movement is added. Stay on movement until the sprite reliably responds, then add cherries together.
- Developing: Prompt them to read SpriteKind on create and overlap aloud before pressing play. If score or sound fails, ask which blocks sit inside onOverlap and have them fix the order.
- Proficient: Invite them to change the interval time or the score increment using the same blocks already in the project. Ask them to explain to a partner how player, food, overlap, score and sound connect.