Discovery EDUCATION | CODING

Unit 4a Lesson 1: Pop game
50 minutes

Cross-curricular links

Maths: addition, multiplication, time

Overview

In this lesson children will be introduced to variables and how they can be used in computer programming. They will begin to understand that a score in an app is written into the code as a variable.

Learning objectives

Learn how to use variables to keep track of the score in a game

Success Criteria

ALL I can write code which includes a variable that will increase in value each time a balloon is popped
MOST I can program the variable to increase in value by different amounts when different balloons are popped
SOME I can add a time limit to my app and explain how I have used a variable to keep the score

Key words

variable, condition, score, start, click, place, time

1. Engage

Prior knowledge:

- Show an example of an app that has been created during Step 5 of this lesson
- Using the interactive whiteboard, ask children to compete in groups of three to see who can achieve the highest **score**
- Discuss how this app may have been created and gain understanding of the children's prior knowledge
- Start a learning wall and add key vocabulary children use when talking about computer programming

New learning:

- Share the Learning objective and Success criteria, using the 'I can' statements
- Discuss children's experiences of keeping score - include quizzes, computer games and sports
- Host a short quiz. Show children a container marked with an 's'
- Discuss with children that there is nothing in the container yet, so the value of 's' is 0
- Write 's=0' on the board and explain that 's' is going to represent the score in the quiz
- Ask a few questions and throw a ball into the container when the children answer a question correctly
- Change 's=0' to 's=1' and so on when children get answers right - emphasise the value of 's' changing
- Explain that, in this quiz, 's' is a **variable** - a space for a value that can change - and this lesson looks at how **variables** can be used in code to represent a score
- Add key words '**variable**', '**score**'and '**value**' to the learning wall

Key questions

- *How do you achieve the highest **score**?*
- *What other games have you played that have a **score**?*
- *How do you increase the **score** in those games?*
- *How do you think this app could be programmed? Can you explain?*

2. Explore and learn

Challenges

- Before starting to use **variables** in code, it is important to recap how to use code to program the balloons' movements and visibility on the screen
- Set Steps 1-2 as independent or pair challenges

Feedback

- Ask children to explain what they have done so far. Address any misconceptions

My Design screen

- Watch the video for Step 3 - Count how many pop
- Talk about '**var_s**' as the space where the **score** (**variable**) will be displayed

My Code screen

- Write the code to make the balloons move at the start and hide when they are clicked on
- Add the '**var_s**' code icon into a click function box and then +1
- Ask children to predict how this is going to work (how this code will execute) when the app is run
- Recap how to **save** and **share** an **app** so it can be used on different computers
- **Challenge:** To make a program in which balloons move around the screen and a score increases each time a balloon is popped
- **Algorithm:** Break the challenge down into parts and discuss what could be included in the algorithms
  - The balloons need to move up or down
  - When they are clicked on, the balloons need to hide and the score needs to increase

Final challenges

- Set Steps 3-6 as final challenges. Ask children to design an app and write the code that implements the **algorithms** and solves the challenge. In Step 5 children can set a **time limit** (time events were covered in Unit 3b). In Step 6 children can change the balloons to different objects and design their own apps

Key questions

- *What do you want the balloons to do at the start of the game?*
- *How do you code the balloons to move automatically when the app is started?*
- *What is '**var_s**'?*
- *How do I hide or pop my balloon?*
- *How could scoring in our balloon pop game work?*
- *How many points would you like each balloon to be worth when you pop it?*
- *What code will you use to make the balloon change position on the screen once it has been popped?*
- *What do games usually say at the end of a level?*

3. Evaluate

- Children explain to each other how their app is working using the key words '**score**' and '**variable**'

Key questions

- *Can you describe your app using the words '**score**' and '**variable**'?*

4. Share

- Show a few examples of children's apps to the class
- Revisit the Learning objectives and Success criteria, and ask children to reflect on their achievements

Key questions

Have you achieved the Learning objectives?

Learning beyond school

Children to play different computer games at home and note down: if they keep score, how they keep score and if they give messages during gameplay.

Next steps

Next lesson children will consolidate their understanding of variables and how they can be used in an app to keep score. They will learn that the value of a variable can change as a result of an input or event, or in response to a condition being met.

Discovery EDUCATION | CODING

Unit 5a Lesson 2: Simple Driving game
50 minutes

Cross-curricular links

Maths: position and direction, angles

Overview

In this lesson children will use variables to control the direction and speed of a car within a game. They are introduced to the concept of working iteratively and explore what it means to use computational thinking to solve challenges.

Learning objectives

Learn how to change an object's direction and heading to create a driving game

Success Criteria

ALL I can use values in my code to control the speed and direction of a car
MOST I can use conditional events and values that represent angles in my code
SOME I can use computational thinking to design and create an app that solves a challenge and explain how my app works

Key words

angle, speed, heading, if, assign, decompose, iteratively

1. Engage

Prior knowledge:

- Show an example app made using Unit 5 Lesson 1 Step 6
- Discuss what the challenge was and how the code has been written to make this app work

New learning:

- Share the Learning objective and Success criteria using the 'I can' statements
- Explain to children that they will be challenged to create an app in which an object can change the direction, **speed** and **heading**

Key questions

- *What are the technical words for **speed** up and slow down?*
- *Why might programmers need to keep trying and testing out ideas?*

2. Explore and learn

Watch the video

- Watch the video for Step 1 - Simple Driving game
- Work through Step 1 with the children recapping how to **assign** values in the code to make the car accelerate/ decelerate when different keys are pressed

My Design screen

- Show the My Design Screen for Step 2. Explain that this time values are going to be used to make the car turn and move in different directions, add **angles** to the learning wall
- Ask children how they would normally measure the size of a turn - elicit their understanding of **angles** - ask them to stand up and turn 90, 180, 270 and 360 degrees
- Explain to the children that the values used in the code to make the car turn will be **angles**

My Code screen

- Choose a child to drag the correct code icons into the function boxes and **assign** values to set the change in the **heading** (the direction in which the car moves) and the **angle** (way the car faces) when different keys are pressed. Run the app to demonstrate this working
- Explain to the children that they can work **iteratively** to try out different values

Challenges

- Set Steps 1-3 as independent or pair challenges

Feedback

- Ask some children to explain what they have done so far - address any misconceptions
- Show the My Design screen for Step 4 and explain that the car needs to stay on the track
- Ask children to come up with a conditional statement that could be used in the code to make this work (e.g. if background is green then car stop)
- Discuss the challenge and ask children to use computational thinking to help them solve it
- **Computational thinking**: Break the challenge down into parts and think logically about what could be included in the algorithms to help plan the code that will solve it. Test and debug code to make sure it is accurate.
- Add 'computational thinking' and the description to the learning wall

Final challenge

- Set Steps 4-6 as final challenges, question children as they work about what happens if the **heading** is set but not the **angle**, and vice-versa

Key questions

- *What happens if you change the values of the variables that control the speed?*
- *Which part of the code controls the way the way the car steers?*
- *How do you need to change the direction in which the car moves?*
- *Can you tell me what you have done so far?*
- *Can you explain how the code controls the **speed** and direction of the car?*
- *What happens if you set the **heading** but not the **angle**?*
- *What happens if you set the **angle** but not the **heading**?*
- *How can we ensure the car stays on the track?*
- *Can you explain what the **if** statements in your app do?*

3. Evaluate

- Children share their app with a partner and explain to each other how their app works using key words **angle** and **heading**, they suggest next steps to develop their apps

Key questions

- *What do you like about your partner's app?*
- *How could you develop your app?*

4. Share

- Open ashared app and share it with the class, open another and discuss the differences
- Revisit the Learning objective and Success criteria, ask children to reflect on their achievements

Key questions

- *What did you learn today?*
- *What was your favourite part of today's lesson?*

Learning beyond school

Children visit Espresso Coding at home and create an app that enables a player to the move an object around a course controlling its speed and direction, while preventing it from going off the track.

Next steps

Next lesson children practise setting values in their code to control the movements of a boat. They are introduced to using co-ordinates in code and using negative numbers to alter the location of the boat along the X axis when it hits moving waves.

Discovery EDUCATION | CODING

Cross-curricular links

Maths: algebra, multiplication, measurement

Unit 6a Lesson 2: Unit conversion (cm to inches) 50 minutes

Overview

In this lesson children will create a mobile app that uses variables and calculations to convert between imperial and metric units of measure (inches and centimetres). They will also practise writing code that uses prompt boxes which ask the user to input values for variables.

Learning objectives

To learn how to code functions which use formulae to convert one measurement into another

Success Criteria

ALL I can write code that prompts the user for the value of a variable when they click on a button
MOST I can write code for an app that performs a calculation which uses variables to convert inches to centimetres
SOME I can design and create an app that enables the user to convert between inches and centimetes and explain how my app works

Key words

variable, prompt, multiply, pixels, inches, centimetres, imperial, output

1. Engage

Prior knowledge:

- Show an app made during Unit 6b Lesson 1 Step 4
- Ask children to explain how the code for this app is making it work
- Recap key concepts covered in the previous lesson: writing code to **prompt** a user to input a value for a **variable**; writing calculations in code, using the **variable** to create an **output**

New learning:

- Share the Learning objective and Success criteria, using the 'I can' statements
- Explain to children that they will be challenged to create a mobile app that enables the user to convert between **inches** and **centimetres** (and vice versa)
- Discuss the calculation needed to convert **inches** to **centimetres** (**inches***2.5) and vice versa
- Identify the keys on the keyboard used to **multiply** (*) and divide (/)
- Add 'inverse', '**imperial**', 'metric', '**inches**' and '**centimetres**' to the learning wall

Key questions

- *How does the user input the data into the app?*
- *How has the code been written to trigger a **prompt**?*
- *How does this app use **variables**?*
- *How does the app work out the area?*
- *What do **centimetres** and **inches** measure?*
- *How do we convert from **inches** to **centimetres**?*
- *How do we convert from **centimetres** to **inches**?*

2. Explore and learn

Watch the video

- Watch the video for Step 1 - How many **inches**?

- Work through Step 1 together. Recap the writing of code to trigger a **prompt** box which asks for user input when a button is pressed
- Discuss what the user input represents (**inches**)

My Design screen

- Ask children to look at the My Design screen for Step 2 and discuss how they think this app will work

My Code screen

- Children help write code that executes as follows: On click of the **inches** button, a **prompt** asks the user to input a value which will set the value of var_a, var_b equals the value of var_a*2.5
- Recap the term **algebraic formula** and refer to the calculation a*2.5 as the formula that will generate the **output** (in this case the **centimetres**)
- Run the program to demonstrate how it works
- Discuss that *2.5 is approximate - explore how to arrive at a more accurate value
- Discuss how conversion apps like this are useful, and where **inches** and **centimetres** are used in real life to measure length

Challenges

- Set Steps 1-4 as independent or pair challenges

Feedback

- Ask some children to explain what they have done so far - address any misconceptions
- Watch the video for Step 5. Discuss the challenge and ask children to use **computational thinking** to help them solve it

Final challenge

- Set Step 5 as a final challenge. Ask children to explain what they are doing as they work

Key questions

- *How do we write code to trigger a **prompt** box?*
- *What do var_a and var_b represent?*
- *Why is * used as a symbol in code for multiplication instead of x?*
- *Where are programs like this used in real-life situations?*
- *How much screen space can the app use if it is to be displayed on a mobile phone?*
- *What is 'computational thinking'?*

3. Evaluate

- Children share their app with a partner. They talk through how it was created and how it works, using key vocabulary '**conver**
- Add key word '**conversion**' to a learning wall

Key questions

- *What do you like about your partner's app?*
- *How could you develop your app?*

4. Share

- Open a completed app and share it with the class. Dicuss how the code is making it work
- Revisit the Learning objective and Success criteria. Ask children to reflect on their achievements

Key questions

- *What did you learn today?*
- *Have you achieved the Learning objective?*

Learning beyond school

Children share their app with members of their family. Family members download it onto their smart phones and use it.

Next steps

Next lesson children will create another mobile app that uses variables and calculations to convert between imperial and metric measurements (miles and kilometres), consolidating their understanding from this lesson.