iCompute teaching

helping schools to inspire and shape creative computing

bett awards 2014 FINALIST bettawards.com

iCompute

For iPad

Year 5 - Unit 2

bett AWARDS 2015 FINALIST

iPad

Computing with iPads

Year 5 – Unit 2

Overview

This unit duplicates iProgram (Year 5) from the whole-school pack, providing the option to teach algorithms and programming using iPads instead of pcs. Using the context of art and drawing, the children will be engaged in creatively developing simple animations using the Pyonkee app – a visual programming language based on Scratch 1.4.

Objectives

- See p2 for a detailed breakdown of lesson assessment focuses and associated success criteria.

2014 Computing Programme of Study

- design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
- use sequence, selection, and repetition in programs; work with variables and various forms of input and output
- use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs
- understand computer networks including the internet; how they can provide multiple services, such as the world wide web; and the opportunities they offer for communication and collaboration
- use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content
- select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information
- use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.

Assessment

P24 contains a record of progress pro-forma

Apps

Pyonkee (free) app available here:

- Pyonkee - Visual Programming with iPad

Other Resources

Scratch reference guide download available here:

- http://info.scratch.mit.edu/support/reference_guide_1.4

Royalty free sound files available here:

- http://fxhome.com/sound-effects
- http://www.grsites.com/archive/sounds/

Curriculum Links

- Mathematics
- Art/Design

Objectives

Lesson	Title	NC Links	Objectives	Vocabulary	Curricular Links	Success Criteria
5.2.1	iMove	• design, write and debug programs that accomplish specific goals; solve problems by decomposing them into smaller parts • use sequence, selection and repetition in programs	• To understand that computer programs containing graphics use x y coordinates and turns are measured in degrees • To use conditional (if) statements	Sprite; up; down; left; right; xy coordinates; condition; if		• The children can move a sprite on stage by programming xy coordinates to change and make it turn • The children's sprites bounce off the edge of the stage when they touch the stage boundaries
5.2.2	ISense	• design, write and debug programs that accomplish specific goals; solve problems by decomposing them into smaller parts • Use sequence, selection and repetition in programs; work with variables and various forms of input and output; • Use logical reasoning to explain how a simple algorithm works and detect and correct errors in algorithms and programs	• To understand that some variables can only be true of false (boolean) • To understand that programs can do different things if the value of a boolean variable is true or false (conditional statements)	Condition; if; boolean; true; false; variable		• The children's 'goodie' sprites do something when they are touching the 'baddie' sprite
5.2.3	iNavigate	• As above	• To create a game that senses events on screen • To program statements that make something happen in response to events on screen	Sense; boolean; true; false		• The children create a simple maze game that senses when a sprite touches a wall • The sprite is moved back to its starting position if it touches a wall
5.2.4	iVary	• As above	• To be able to understand what a variable is and why they are useful	Variable; value		• The children can describe to others the possible uses of variables in a computer game
5.2.5	iScore	• As above	• To understand that variables can be used in programming to keep track of values • To program statements that make something happen in response to the value of a variable	Variable; value; selection		• The children use a 'lives' variable to keep track of how many times their sprite has touched the wall of a maze • The children program a 'game

Lesson	Title	NC Links	Objectives	Vocabulary	Curricular Links	Success Criteria
						over' script to execute when the sprite has touched the wall a number of times
5.2.6	iDesign	• design, write and debug programs that accomplish specific goals; solve problems by decomposing them into smaller parts • Use sequence, selection and repetition in programs; work with variables and various forms of input and output	• To identify an appropriately scoped project • To develop an outline of tasks and activities required to develop a project	design; storyboard; sequence; input; output	Art/Design	• The children create a storyboard of a computer game • The storyboard details a main character, the objective of the game and how the game can be won or lost
5.2.7	iCode	• design, write and debug programs that accomplish specific goals; solve problems by decomposing them into smaller parts • Use sequence, selection and repetition in programs; work with variables and various forms of input and output • Detect and correct errors in algorithms and programs	• To use the computational concepts of sequence, selection, repetition and variables to program a computer game	Condition; variable; boolean; true; false; repeat; loop; repetition; statement; algorithm; selection		• The children produce a simple computer game
5.2.8	iTest	• design, write and debug programs that accomplish specific goals; solve problems by decomposing them into smaller parts	• To develop strategies for testing and debugging computer programs	• test; bugs; debug; amend; systematically		• The children test their computer games • The children make changes to their code when things do not execute as expected

Preparation

- Read the lesson plans
- Check the Scratch reference guide glossary of terms to make sure you are using the correct vocabulary
- Spend an hour or so familarising yourself with the software you will be using and the application's interface

Resources

- iPads
- Ensure that the app is installed on each iPad you will be using
- Worksheets for each lesson – entitled: Worksheet<year.unit.lesson> (eg.Worksheet5.2.1)
- Support materials for each lesson – entitled Resource <year.unit.lesson> (eg. Resource5.2.1)

Links

Before you start, you may find these weblinks useful.

- Pyonkee can be downloaded at

  Pyonkee - Visual Programming for iPad

- Scratch Reference Guide: http://info.scratch.mit.edu/Support/Reference_Guide_1.4

Updates

If any links are not working, visit the iCompute website which contains up-to-date links:

http://www.icompute-uk.com/Links.html

Resources: Pyonkee app; iPads; Resource5.2.1 Folder

Objectives

- To understand that computer programs containing graphics use x y coordinates and turns are measured in degrees
- To use conditional (if) statements

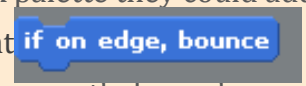
Success Criteria

- The children can move a sprite on stage by programming xy coordinates to change and make it turn
- The children's sprites bounce off the edge of the stage when they touch the stage boundaries

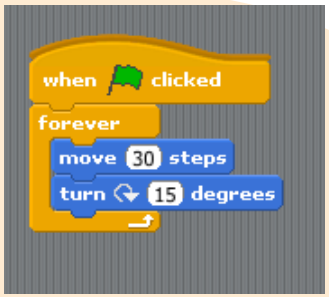
Vocabulary

Sprite; up; down; left; right; xy coordinates; condition; if

Step 1

- Remind the children of their work in Year 3 with Scratch or Pyonkee (much of this lesson repeats lesson 3.2.2 of Year 3 iProgram unit)
- Load Pyonkee and model how to add two sprites to the stage
- Add some code to move one of the sprites and draw attention to the forever loop
- Click on the second sprite and add some code to move the sprite using the up arrow key
- Move the sprite up to the top of the stage and ask the children which of the blocks from the motion block palette they could add to stop the sprite from disappearing from the stage
- Establish that 

could be added to the end of the block to achieve this
- Model how to save their work



Core

Step 2

- The children create two new sprites and name them ‘goodie’ and ‘baddie’.
- They program the baddie sprite to move around the screen
- They then program the goodie sprite to respond to arrow key input from the user (up, down, left and right)
- They use a conditional **if** statement to prevent the sprites from disappearing off stage.

Differentiation

Easier: Children program one sprite to move using arrow keys

Harder: Children could use xy coordinates and glide blocks to move the baddie sprite around the stage

Extension

Challenge the children to determine which xy coordinates (by hovering the mouse over the stage and recording them) they would like the baddie sprite to sequentially move to and add a block that programs the movements to be repeated.

Resources: iPads; Pyonkee app; Resource5.2.2; Resource5.2.1 Folder; Children's projects from 5.2.1

Objectives

- To understand that some variables can only be true of false (boolean)
- To understand that programs can do different things if the value of a boolean variable is true or false (conditional statements)

Success Criteria

- The children's 'goodie' sprites do something when they are touching the 'baddie' sprite

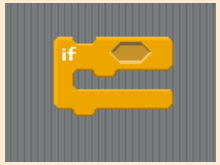
Vocabulary

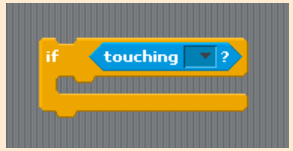
Condition; if; boolean; true; false; variable

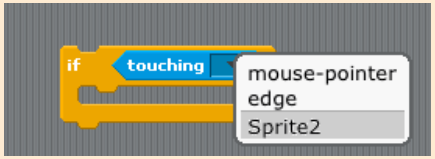
Step 1

- Remind the children of their work using variables in Year 4 (iRobot) where the value can only be **true** or **false**
- Explain that these are **boolean** variables. Get two children to stand up and ask child1 to put their hand on the shoulder of child2
- Tell child2, using only the answer **true** or **false**, is child1 touching you?
- Explain that child2 is sensing that child1 is touching him/her
- Repeat for the colour of child1's sweater or shirt – eg. is child2 touching blue?

Step 2







- Load Pyonkee, add two sprites, name them ‘goodie’ and ‘baddie’
- Go to the control palette
- Drag out an **if** block and draw attention to the blank hexagon
- Explain that this is a space for a **condition**
- Go to the sensing palette, drag out a touching **variable** and snap into the blank space inside the **if** statement
- Click on the arrow for the drop-down menu and choose the name of the baddie sprite
- Choose something for the goodie to do when touching the second sprite and execute the program by clicking the green flag

Core

Step 3

- Children develop their projects from the last session
- They use conditional **if** statements and **sensing** blocks to make something happen when the goodie touches the baddie (nb: this could be a costume change, a sound being made, a background change or text display)

Differentiation

Easier: Some children could use projects where the sprite movement is already programmed

Harder: Children could program a number of different effects to execute when the goodie touches the baddie

Extension

Children add repetition to program the baddie to **always** react when the goodie touches them

iPad 5.2.3 iNavigate Year 5

Resources: iPads; Pyonkee app; Resource5.2.3.sb

Objectives

- To create a game that senses events on screen
- To program statements that make something happen in response to events on screen

Success Criteria

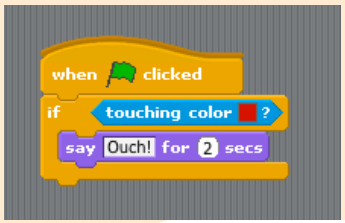
- The children create a simple maze game that senses when a sprite touches a wall
- The sprite is moved back to its starting position if it touches a wall

Vocabulary

Sense; boolean; true; false

Step 1

- Explain that Pyonkee can tell when a sprite comes into contact with a certain colour on the screen, in the same way as the sprite could tell (in the last session) that it had come into contact with a certain sprite.
- Load the app and open a project file with a maze background (resources). Add a sprite and model how to shrink it to fit the maze
- Move to the sensing palette and drag out a touching colour block.
- Model how to change the colour by clicking on the colour and selecting the red on stage with the colour picker



Core

Step 2

- The children load Resource5.2.3.sb and save it under their own names
- They add a sprite of their choice
- They program their sprite to navigate the maze using motion blocks
- They use conditional statements (touching colour) to move the sprite back to the beginning of the maze if it touches the colour of the maze wall

Differentiation

Easier: Children navigate their sprites using only quarter turns

Harder: Challenge the children to use smaller degree turns to navigate the sprite

Extension

Children create a maze background of their own and challenge another member of the class to navigate it

iPad 5.2.4 iVary Year 5

Resources: iPads; Pyonkee app; Resource5.2.4-Score.sb; Resource5.2.4-Timer.sb; Resource5.2.4-Enemies.sb; Resource5.2.4-Levels.sb; Resource5.2.4-Rewards.sb

Objectives

- To be able to understand what a variable is and why they are useful

Success Criteria

- The children can describe to others the possible uses of variables in a computer game

Vocabulary

Variable; value

Step 1

- Discuss last session's maze project and talk about what the children liked/disliked about it
- Take suggestions about how the maze game could be improved

Core

Differentiation

Step 2

- Divide the class into groups of three pupils
- Assign each of the groups of three a pre-programmed maze extension project (resources) to explore:

1. Score - Demonstrates how to set and change a score. Receive 10 points every time the cat is clicked.
2. Timer - Demonstrates how to use a timer. Use the mouse to navigate the cat to Gobo.
3. Enemies - Demonstrates how to add an enemy. Avoid the tennis ball using the up and down arrow keys.
4. Levels: Demonstrates how to change levels. Score increases by 1 every time the space bar is pressed. Level increases by 1 for every 10 points.
5. Rewards: Demonstrates how to collect items. Use the arrow keys to move the cat around to collect items for his quest.

Step 3

- One member of each group then moves to another group and teaches that group what they have learned
- They explain what their group's project was and how variables were used
- They then discuss as a group how that maze project could be extended/improved

Resources: iPads; Pyonkee app; Resource5.2.3.sb; Resource5.2.5-Lives.sb; Children's projects from 5.2.3

Objectives

- To understand that variables can be used in programming to keep track of values
- To program statements that make something happen in response to the value of a variable

Success Criteria

- The children use a 'lives' variable to keep track of how many times their sprite has touched the wall of a maze
- The children program a 'game over' script to execute when the sprite has touched the wall a number of times

Vocabulary

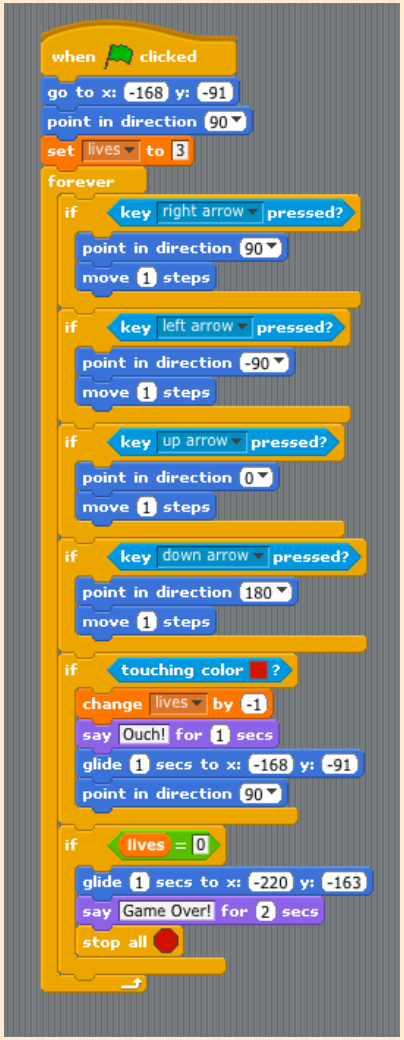
Variable; value; selection

Step 1

- Remind the children about the last session and the projects they explored involved using variables to keep track of events
- Load Resource5.2.5-Lives.sb that shows the script for keeping track of lives
- Explain that this variable allows you to keep track of how many times a sprite has touched the maze wall and can be used to give the sprite a certain number of attempts before the game ends

Step 2

- Discuss the script and draw attention to how the number of lives is set at the beginning
- Explain that each time the sprite touches the wall it loses a life
- When the sprite has run out of lives (lives = 0) it announces ‘Game Over’
- Also, that the game ends



Core

Step 3

- The children develop their maze projects from 5.2.3 (resources) to add a 'lives' variable that keeps track of the number of times their sprite has touched the maze wall
- They then program a 'Game Over' script that executes when their sprite has run out of lives (nb: they can be as creative as they like – it could be a background change; sound effect; costume change; hiding the sprite; text display etc.)

Differentiation

Easier

Children could create a simple text message, or make a sound when the game is over

Harder

The children could design a game over background and switch the background costume to this when the game is over

iPad 5.2.6 iDesign Year 5

Resources: Worksheet5.2.6 (photocopied and enlarged to A3)

Objectives

- To identify an appropriately scoped project
- To develop an outline of tasks and activities required to develop a project

Success Criteria

- The children create a storyboard of a computer game
- The storyboard details a main character, the objective of the game and how the game can be won or lost

Vocabulary

design; storyboard; sequence; input; output

Step 1

- Ask the children to talk about which games they have enjoyed playing the most in the past
- Discuss what features the games had that made the enjoyable – fast action/good graphics/story?
- Talk about what was required to win the game(s) – How did you control play (**input**) how did you know you'd won (**output**)
- Tell the children that in this session they will be designing their own computer game. They can try to replicate an existing game that they like or invent a completely original one.
- Show the children a selection of classic games on the internet (eg. Mario)
- Explain that computer programmers do not start with coding. They design their programs carefully and often produce storyboards first.
- Display a copy of Worksheet5.6 on the interactive whiteboard (Resources) and talk through the elements.

Step 1 (cont…)

- Tell the children that, when they are designing their game, they will need to think about:

1. Where is the game set?
2. Who is the main character?
3. How will the player control the character?
4. What is he aim of the game?
5. How does the player win or lose?

Core

Step 3

- Give out Worksheet5.2.6
- The children create a storyboard of their chosen computer game
- They identify a main character sprite, draw backgrounds and scenes
- They write about how play is controlled and how the game is won or lost

Tip: Any project created using Scratch 1.4 (i.e. projects ending in .sb) can be opened with Pyonkee. Locate a Scratch 1.4 project, tap it and choose 'Open In' Pyonkee. Pyonkee cannot open projects created with Scratch 2.0 but you can convert Scratch 2.0 to Scratch 1.04 here: https://kurt.herokuapp.com/20to14

Differentiation

Easier

Children could use existing projects available at http://www.scratch.mit.edu for inspiration

Extension

The children swap designs with another pupil and evaluate the designs of others. Has their partner considered all possibilities? Can they identify any potential problems? How could their design be improved?

Resources: completed Worksheet5.2.6 (from previous session)

Teaching Notes: This session may span more than one session

Objectives

- To use the computational concepts of sequence, selection, repetition and variables to program a computer game

Success Criteria

- The children produce a simple computer game

Vocabulary

Condition; variable; boolean; true; false; repeat; loop; repetition; statement; algorithm; selection

Step 1

- Explain that, in this session the children will be using the designs they created in the last session (resources) to program their own computer games
- Discuss some of the children's designs and draw attention to any good designs where the objective and steps necessary towards achieving a win are clearly stated/drawn
- Remind the children about how variables can be used to keep track of events in computer games
- Discuss where variables could be used in some of the children's designs – eg. lives, scores, levels
- Get the children to discuss with a partner their idea for a computer game and to describe to them, who their main character is, how the player controls events and how their game is won/lost

Core

Step 2

- Give out Worksheet5.2.6 (completed in last session)
- The children work on computers to program the functionality they designed into a computer game
- Encourage the children, as they code, to test systematically to reduce bugs

Differentiation

Easier

Children could use Pyonkee to remix existing Scratch scripts downloaded from www.scratch.mit.edu

Extension

The children swap their projects with another pupil and test each other's programs

iPad 5.2.8 iTest Year 5

Resources: iPads; Pyonkee app; Resource 5.2.8; Completed Worksheet5.2.6 (from 5.2.6 and 5.2.7 session); Children's project files from previous sessions

Objectives

- To develop strategies for testing and debugging computer programs

Success Criteria

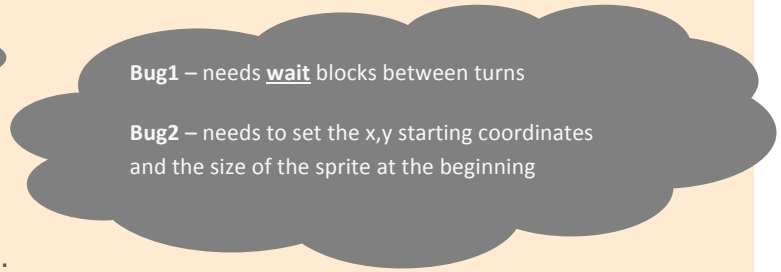
- The children test their computer games
- The children make changes to their code when things do not execute as expected

Vocabulary

test; bugs; debug; amend; systematically

Step 1

- Discuss with the children their programming experiences so far and ask what ways they have found helpful when they get stuck programming
- Display two of the sections of code in Resource5.8 which contain **bugs** (resources)
- Read out the intention of the programmer (the prompts) and the problems they have encountered when executing their code
- Ask the children to discuss in pairs:
  1. What is the problem?
  2. How did you identify it?
  3. How can it be fixed?
  4. Did anyone else come up with alternative ways of fixing the problem?



- Explain that the children can use the same strategies when testing their own programs.
- Tell the children that real programmers test their code systematically. They go through all possible ways of executing their code making sure that each statement is executed and deliberately try to make the code fail in order to establish that the code is bug free

Core

Step 2

- The children use their designs created in session 5.2.6 and continue to develop their programs for their own computer games (resources)
- They test their code to make sure it works as expected
- When they encounter problems with their programs they analyse their code and debug it

Extension

The children swap their projects with another pupil and test each other's programs. They then evaluate each other's work and discuss how they think their projects could be extended or improved

Assessment

Record of progress Write names in the appropriate box, with jottings on children on children whose attainment differs markedly from their group.	Expectations What children know, understand and can do
Some children will have not made as much progress and will: 	• Know that computer programs contain commands that achieve a specific action • Write or amend computer programs to produce specific actions with assistance
Most children will: 	• Write and amend computer programs • Program a number of algorithms that achieve a specific outcome • Use repetition, variables and conditional statements in computer programs • Test computer programs and correct any errors
Some children will have progressed further and will: 	• Write and amend more complex computer programs to create a variety of outcomes • Program algorithms that achieve a range of specified outcomes • Test, debug and refine computer programs
Guide - Working at Orange	