



KODU * GAME LAB *

Game Making Workshop Manual Munsang College 8th May 2012



HONG KONG
DIGITAL GAME-BASED LEARNING
ASSOCIATION

香港數碼遊戲為本學習協會



KODU

GAME LAB

Course



Module 1 Introducing Kodu Basics

Introducing Kodu Game Lab

Kodu Game Lab is a visual programming language that allows anyone, even those without programming knowledge and experience, to create rich 3D games.

Kodu uses intuitive icons to control the characters and objects with character behaviours expressed in physical terms.

This Kodu Game Development Workshop is a designed primarily for teachers looking to introduce game design into their classroom program. This 9-hour intensive course will introduce Kodu Game Lab and demonstrate how anyone can create rich and exciting games.

The course will cover the skills need to design and create worlds and games with Kodu Game Lab and will also investigate the design process and how Kodu can used in the school curriculum.

Through designing and creating their own games, participants will learn how powerful and interesting games can be created with simple building blocks and techniques.

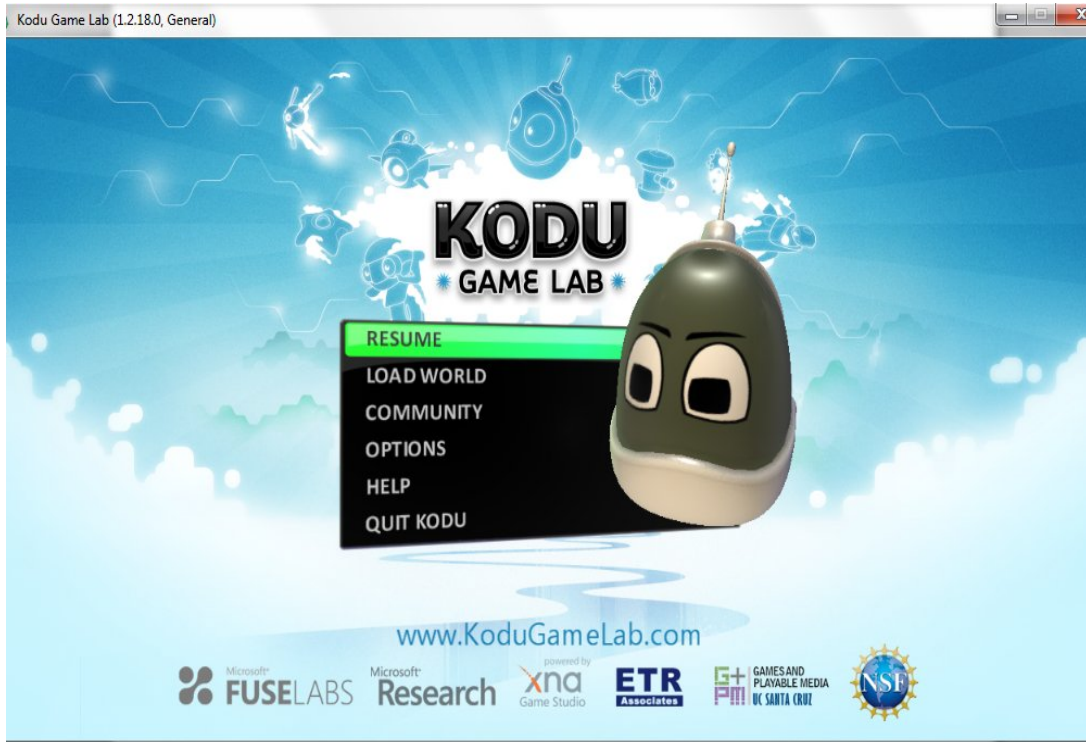
This course will:

- enable participants to program characters and objects
- enable participants to build 3D worlds
- introduce the key elements of games design
- provide a rich range of code examples
- provide a rich range of game play examples
- give plenty of hands on time so that participants are confident in using Kodu

Download Kodu Game Lab:

<http://fuse.microsoft.com/page/kodu>

Introducing Kodu's programming language




Programs for the Kodu Game Lab characters are created as a sequence of numbered rows. Each row has a WHEN part and a DO part. When the WHEN part side is true, then the action on the DO side is done. For example:




In this case when the character sees an apple the character will move toward it.

A. Complete the First Tutorial

This tutorial will teach you how to:

- add  to your world
- programme Kodu to respond to user input


1. Choose **“Load World”**
2. Click **“First Tutorial”** and **“Play”**
3. Press Esc (keyboard) to edit the world.

4. Select the **“Add and Program Objects”** icon  from the toolbar.
5. Left click anywhere on the ground to add a new object
6. Choose “Add object” and click on Kodu
7. Customize Kodu’s colour, size and height

8. Click on the **“Object tool”**  to add or programme the character
9. Right click on Kodu and select **“Program”**




10. Press Esc to go back to the Edit Menu


11. Click on “Play Game”  to run the programme (control Kodu to move)


B. Complete the Programming Kodu to Find Apples Tutorial

This tutorial will teach you how to:

- add  to your world
- programme Kodu to find and eat the apples

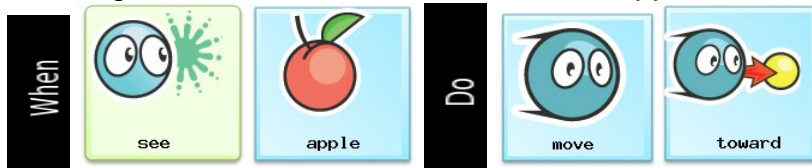
1. Choose **“Load World”**
2. Click **“Programming Kodu to find apples tutorial”**
3. Press Esc (keyboard) to edit the world.

4. Select the “Add and Program Objects” icon  from the toolbar
5. Left click anywhere on the ground to add a new object

6. Choose “Add object” and click on  (inside Objects)
7. Add a few apples and then add Kodu

8. Click on the **“Object tool”**  to add or programme the character
9. Right click on Kodu and select **“Program”**

10. Programme Kodu to move towards the apples when he sees them



11. Now programme Kodu to eat the apples when he bumps into them



12. Press Esc to go back to the Edit Menu

13. Click on "Play Game"  to run your game

Getting to Know the Bots and Objects

In this activity we will explore the various bots and objects that are available in Kodu Game Lab. We will investigate the various attributes and abilities of the different bots.

1. Different bots have different attributes and abilities.
2. Use colours to distinguish different character roles, e.g. red cycle, black cycle
3. All bots and objects have settings that can set as the game is created but not by using programming during the game

The Edit Menu:



1. Home Menu: 主目錄
2. Play Game: 執行遊戲
3. Move Camera: 移動鏡頭
4. Object Tool: Add or Edit Characters and Objects 物件工具 (添加或編輯人物或物件)
5. Path Tool: Add or Edit Paths 路徑工具(添加或編輯路徑)
6. Ground Brush: Paint, Add or Delete Ground 地面刷 (油漆, 添加或刪除地面)
7. Up/Down: Create Hills or Valleys 向上/向下 (製造山區和山谷)
8. Flatten: Make Ground Smooth or Level 弄平 (使地面光滑或水平)
9. Roughen: Create Spiky or Hilly Ground 粗糙 (創建高低不平地面)
10. Water Tool: Add, Remove, or Tin Water 水工具 (添加, 刪除或色調水源)
11. Delete Tool: Use a Brush to Clear Objects 刪除工具 (用刷子清除物件)
12. Change World Settings: 更改遊戲世界設定

The Bots:

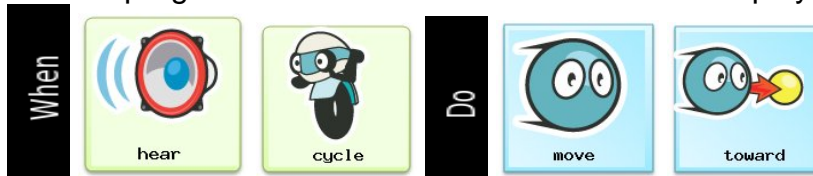
A. Player controlled bots

We can write simple programs in order to control a bot with the keyboard or xbox controller. We can also map actions such as jumping to specific keys on the keyboard or gamepad.

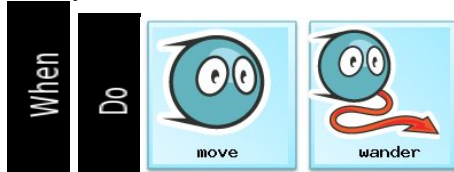


B. AI bots

We can program other bots that will interact with the player's bot.



If we just want the bots to wander rather than follow we can use:



C. The Bots in Kodu Game Lab:



.5.3 Sensors and actions:

- See, hear, got, bump, shot hit, health, held by, on land and on water are physical actions of the bot or object.
- Timer, scored, gamepad, keyboard and mouse are external actions that the bot or object can respond to.
- All of the actions are not available to all of the bots and objects.

A. When physical sensors:













B. Do physical actions:



Solving Problems 1 and 2









Problem 1

Objectives:

- To add Kodu  and apples  on the map
- To program Kodu to move  with keyboard Arrow keys
- To program Kodu to jump  with keyboard Space key
- To program Kodu to shot  with keyboard S key
- To program Kodu to eat  apples  when bump  into them
- To program Kodu to say  "Yummy!" when get the apples 

Problem 2

Objectives:

- To add Cycle  on the map
- To program Cycle to wander  on the map
- To program Cycle to vanish  apples  when bump  into them
- To program Cycle to stun  Kodu  when bump  into it

C. Game Play:

After all the programmes are set, try to test the game!

You should be able to control Kodu to eat all the apples on the map. Avoid bumping into Cycle, it will stun Kodu. Try to jump and shoot Cycle. Adjust the game camera with the mouse scroll wheel and try playing the game in first-person mode!

Module 2 Creating Game Worlds

Creating New World

- Choose “Load World”,
- Click “Empty World”
- Click “Play” and you will have a new empty world
- Click Esc to start editing your world
- Choose Home Menu to name your game and save your world

Designing Game Spaces

This activity creates understanding of resource considerations, world settings and cameras. Large worlds use a great deal of resources and is crucial to use as little land as possible.



A. Move Cameras

- Press F3 to turn on the **Snap to Grid** feature
- the cursor coordinates will be displayed in the bottom left hand corner of the screen.
- Choose “Move Camera” menu (the hand)
- Left click mouse to adjust the camera
- Zoom in/ zoom out with the mouse scroll wheel
- Right click mouse to change the direction to North

B. Ground Brush

- Left click mouse to build ground
- Right click mouse to erase ground
- Adjust brush size with keyboard left/right arrow keys
- Choose ground patterns and brush shapes
- Build your game map



C. Up/Down Brush

- Left click mouse to raise the selected land
- Right click mouse to lower the selected land
- The different shaped & sized brushes can be used to create different effects



D. Flatten Brush

- Left click mouse to level the selected land
- Right click mouse to smooth the selected land



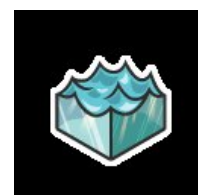
E. Roughen Brush

- Left click mouse to make the selected ground spiky
- Right click mouse to make it hilly



F. Water Tool

- Left click mouse to raise the water level
- Right click mouse to lower the water level



G. Delete Tool

- Use the Delete Tool to quickly remove bots and objects from your world



Game Play Design Patterns

A. Design Pattern: Time Limits

Time Limits require a player to complete an action or achieve a goal or alternatively sets a time that the player must survive in the game in order to win. Countdown clocks usually display the time remaining to give the game a sense of urgency. Some games feature time bonuses that are gained through achieving certain tasks.

Demonstration:

- Add an object (e.g. a tree) on the map
- “Program” the tree to set the timer
- Set the timer for 30 seconds, after 30 seconds the game is over
- We have to do it in three pages, for actions to be taken step-by-step
- Page One: the time 30 seconds will be shown in black once the game starts
- Page Two: one black score will be deducted each second, when only one second remains, turn to page three
- Page Three: one second later, the game ends

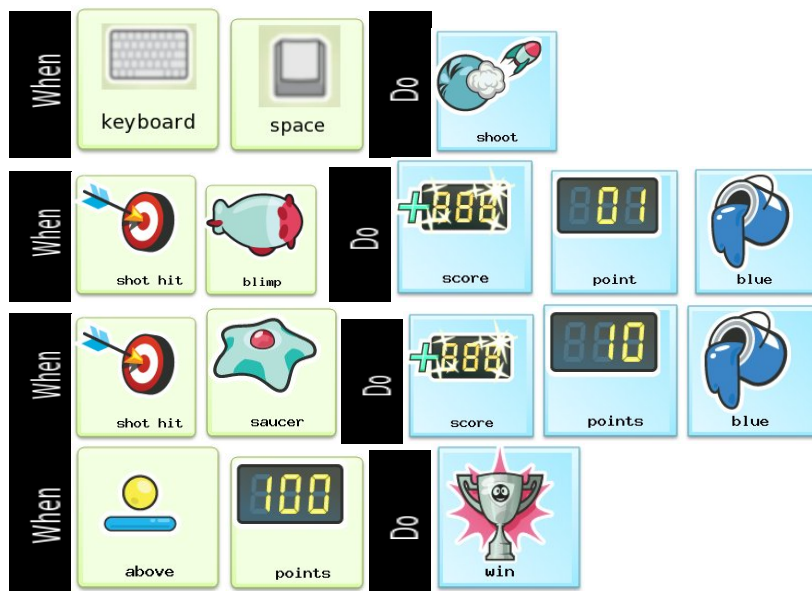


B. Design Pattern: Scores

Scores are a numerical representation of a player's success. Points can be added to a player's score for achieving certain goals or deducted for failed activities. A player's score is usually displayed at all times during a game, often points achieved are displayed as an overlay as the goal is achieved adding to the sense of achievement and progress.

Demonstration:

- Adds 1 point to the blue score each time a shot hits the blimp and 10 points each time a saucer is hit.
- When the score is more than 100 the player wins the game.

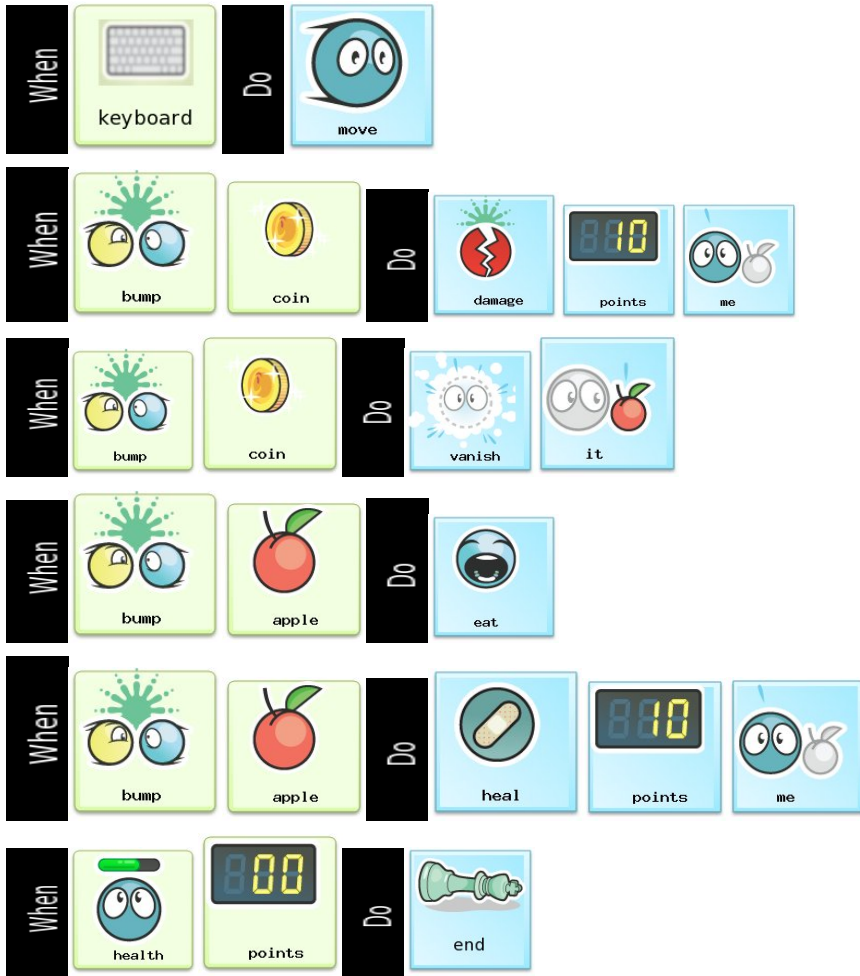


C. Design Pattern: Health

The health of the player's bot can also be used to indicate progress within the game. A health bar shows the current health of the character and provides immediate feedback to the player.

Demonstration:

- When the bot touches a coin 10 points are deducted from the bot's health and the coin vanishes.
- When the bot touches and eats an apple 10 points are added to the bot's health.
- When the bot's health is zero the game ends.
- The number of Hit Points (health) a bot starts with is set in the bot's settings.
- Values between 0 and 1000 in increments of 5 can be set.



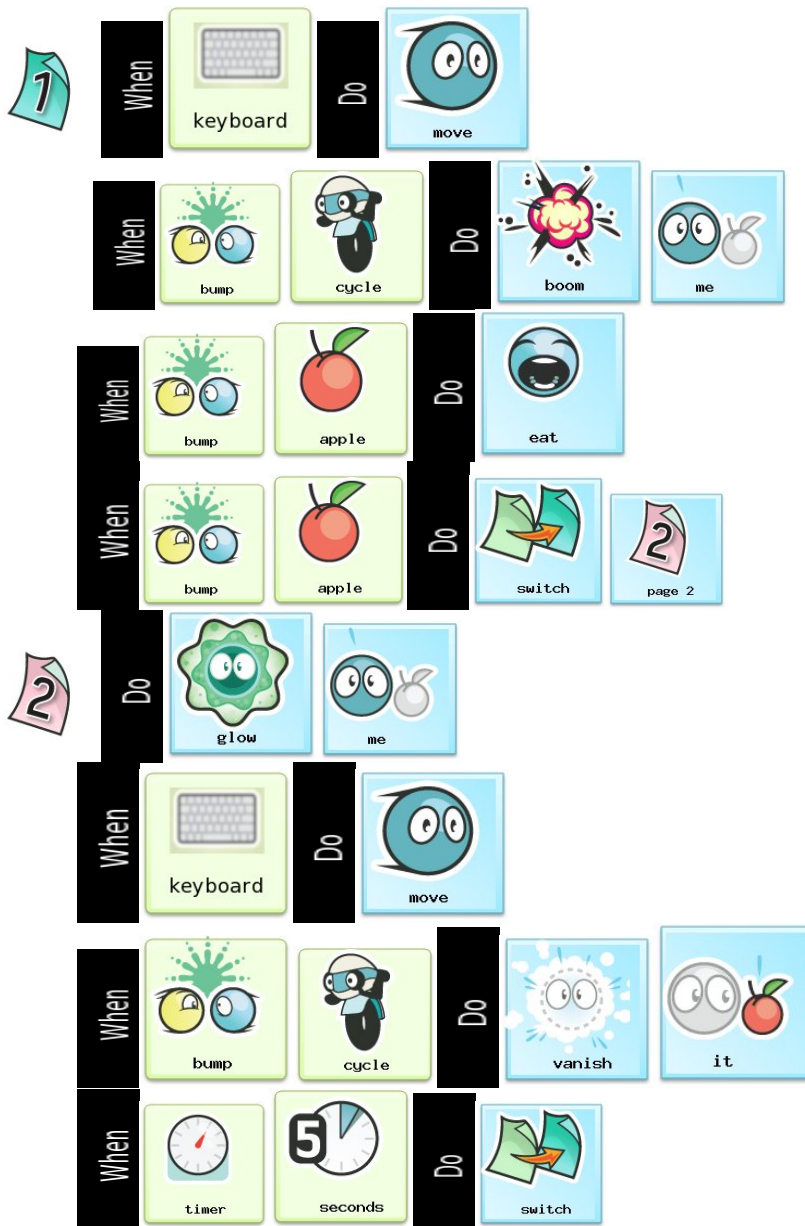
D. Design Pattern: Power Up

Power-Ups give a time limited advantage to the player that activates them. Power Ups require players to make strategic decisions about when to use them as power up need to be earned or require time to recharge. Power Ups solve the problem of having to complete a variety of tasks of varying difficulty as they let the user have increased abilities to face harder challenges without making the tasks (without power ups) too easy.

For example, a character may become invulnerable or be able to move faster for a period after obtaining an item or after a certain key is pressed. Sounds or visual effects can be used to signal that a power up is being used and/or about to end.

Demonstration:

This recipe programs a bot which is controlled by a player to lose the game when it is bumped into by a Cycle bot. When the player's bot bumps and eats an apple it is powered up so that the Cycle bots are now destroyed when they are touched by the player's character. The power up lasts for 5 seconds and then the player's bot returns to its original state.



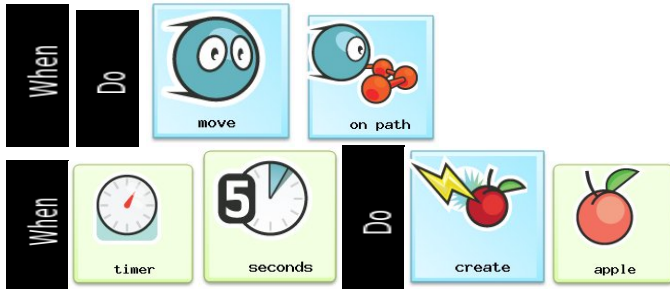
E. Design Pattern: Creates

Some objects can create other objects in the game at the default time.

Demonstration:

This example shows that the cloud will create an apple in every five seconds

- Add a Cloud
- Programme the cloud to wander on the map
- Set the timers, create an apple every five seconds



Game Information Design Patterns

It is important to communicate with players so that they clearly understand the state of the game and their progress within it. Game information can be communicated with text, sounds, scores and change to the game space. Clear communication helps prevent frustration and unexpected outcomes for the player.

A. Pattern: Alarm

Alarms can be used in games to warn players of danger. They can be used indicate that a phase of the game is about to begin or end. Different sounding alarms can be used to notify the player of different events

Demonstration:

This recipe shows how an alarm can be used to indicate that cycle bot is close by.



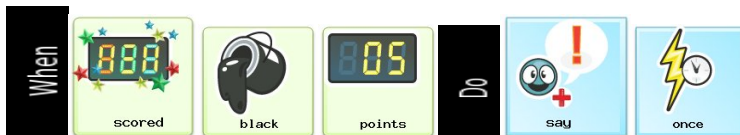
B. Pattern: In Game Information

Dialog boxes can be used to display information to the user, this information may simply notify the user of their progress in the game or it may provide information that the player needs to successfully complete the game.

There are three kinds of dialog boxes in Kodu Game Lab: Full Screen and two Thought Balloons, one in which the lines are displayed sequentially and one in which lines are displayed randomly. Full screen is used to give general messages to the player, while speech dialog boxes are used when specific characters are used provide information to the player.

Demonstration:

In this example a full screen dialog will be displayed when the black score reaches five points.



Module 3 Game Design & design process

What is a game?

- Games are fun
- Games have rules
- Games have an objective / Games have winners and losers
- Games require players to make decisions

**“A game is a problem-solving activity, approached with a playful attitude.”
Jesse Schell**

Generating Ideas

Game Name: _____

Objective: _____

Rules: _____

Bots used: _____

Design process

A. Designing your game

- Name and descriptions
- Game play
- One or multiple players Characters/ Objects involved
- Scoring
- Timer

B. Working together

- Communicate and collaborate with your teammates
- Shift roles regularly (to make the game or to monitor the process)

C. Testing and debugging

- Is the game running smoothly?
- What can be improved?

D. Trying other's game

- Walk around and play others' game and study the game design
- Give comments and suggestions (what do you like about their games?)

E. Evaluating the games

- Vote for the best game
- Why is it the best?

F. Introducing your game

- Game description and game play?
- How to win? Any time limit?



香港數碼遊戲為本學習協會

More:

info@digitalgameslearning.org
www.digitalgameslearning.org
www.facebook.com/HKDGBLA