Activities

Before you start open go to the TLC website and open the online collaboration link for the day http://teachinglondoncomputing.org/ddpp (password tlcddpp) (ddpp diving deep into primary programming)

Contents

Activities 1

Review 2

Levels of Abstraction and Scaffolding approaches reminder 3

Activity: Session 4 – Selection Objectives 4

Activity: Session 4 PRIMM 1 5

Activity: Session 4 PRIMM 2 6

Activity : Session 4 Targeted Task - buggy code 7

Activity: Session 5 – Variables Objectives 8

Activity: Session 5 –Activity 1 Live coding – variables to gather and display a value 9

Activity: Session 5 –Activity 2 Guided discovery – variables for users to control aspects of a program 9

Activity: Session 5 –Activity 3a Remixing – variables that control a program to increase efficiency/reuse/understanding (1) 10

Activity: Session 5 –Activity 3a Remixing – variables that control a program to increase efficiency/reuse/understanding (2) - extension 11

Activity: Session 5 – Activity 4a – Targeted task – Predict and compare code - variables that control the flow of a program (4a) 12

Activity: Session 5 – Activity 4b – Targeted task – Predict and compare code - variables that control the flow of a program (4b) 13

Session 6 – Make your own blocks and lists objectives 14

Session 6 – Activity 1 compare 15

Session 6 – Exercise Game 16

Session 6 – Fortune teller 17

Session 6 –Job picker 18

Review

Objectives of this session

- Review work you did in school on design.
- In the reflective diary complete your section.
- Then read through other people's reviews.
- Can you spot any recurring themes?

Levels of Abstraction and Scaffolding approaches reminder

Levels of abstraction

1. Task
2. Design (including algorithms)
3. Code
4. Running the code

Copy code	Targeted tasks	Shared coding	Guided exploration	Project design and code	Tinker
				•Imitate •Innovate •Invent Vs •Remix	

Activity: Session 4 – Selection Objectives

Increase understanding of selection specifically:

- Different forms of selection
- Using design to represent selection
- Progression of selection
- Event based programming and other programming paradigms

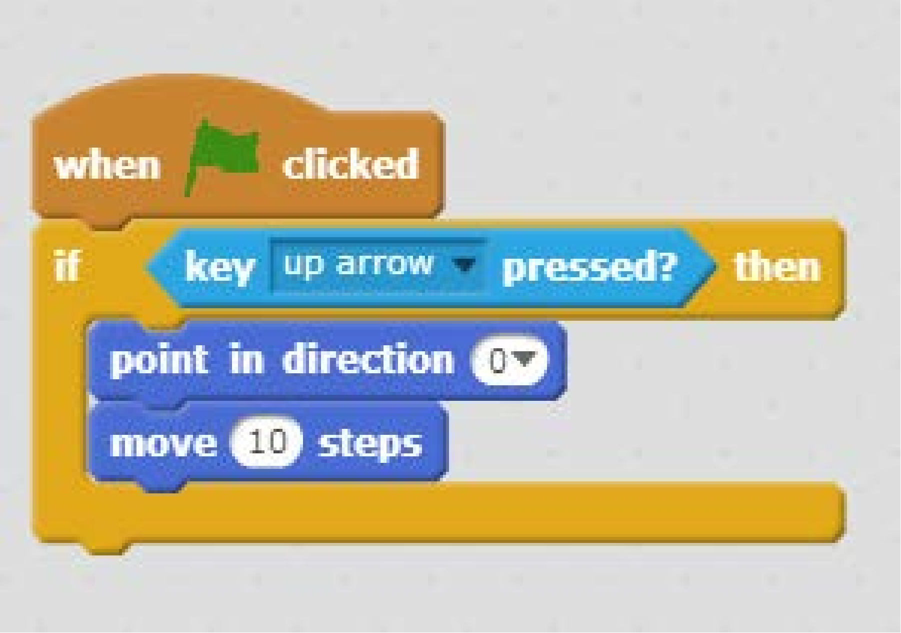
RAG your confidence on each of these

Activity: Session 4 PRIMM 1

- Predict (DO NOT RUN IT! – write down your prediction)
- Then run, then investigate, then modify.
- What would be good modifications? Why?
- What could you make?
- What unplugged activity could you add to help learning?

Selection – sample code 1
https://scratch.mit.edu/projects/167435363/#player

I predict this code will:



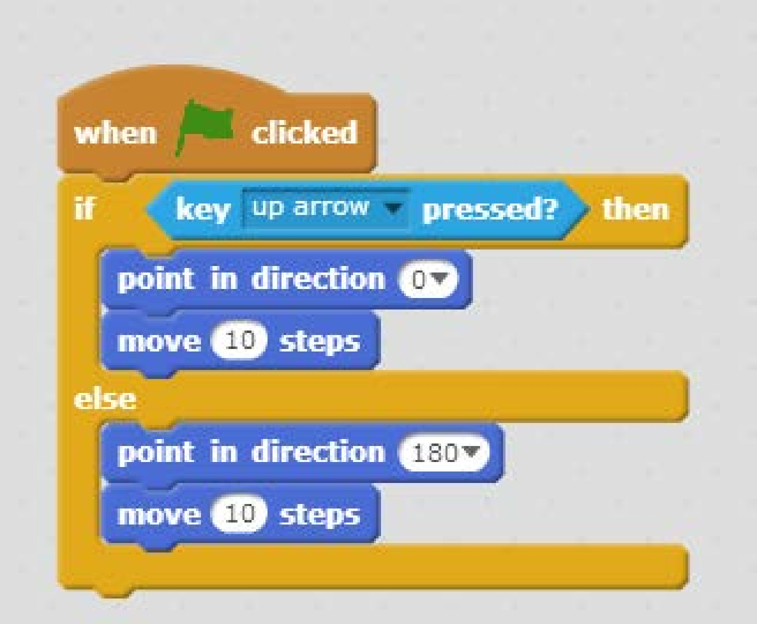
```
when 🏴 clicked
if <key [up arrow ▾] pressed?> then
    point in direction (0 ▾)
    move (10) steps
```

Activity: Session 4 PRIMM 2

- Predict (DO NOT RUN IT! – write down your prediction)
- Then run, then investigate, then modify.
- What would be good modifications? Why?
- What could you make?
- What unplugged activity could you add to help learning?

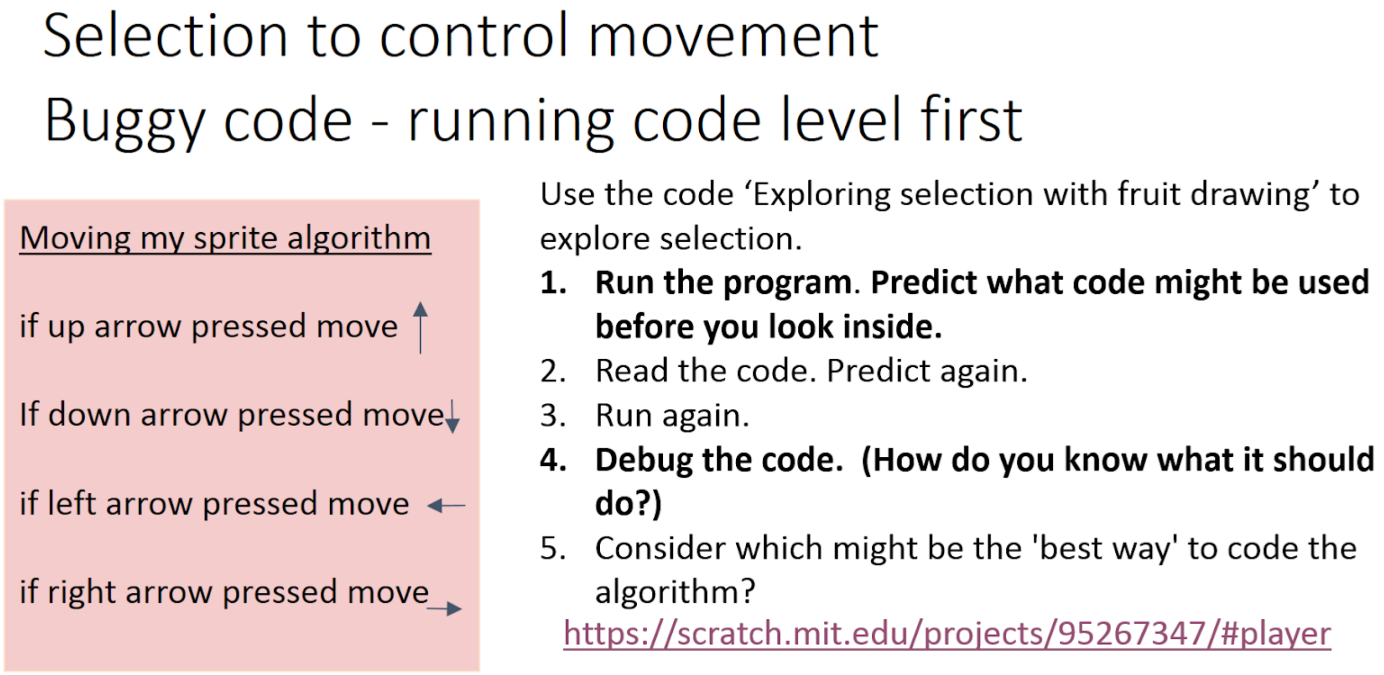
Selection – sample code 2
https://scratch.mit.edu/projects/167435909/

I predict this code will:



```
when 🏴 clicked
if <key [up arrow ▼] pressed?> then
    point in direction (0 ▼)
    move (10) steps
else
    point in direction (180 ▼)
    move (10) steps
```

Activity : Session 4 Targeted Task - buggy code



Selection to control movement

Buggy code - running code level first

Moving my sprite algorithm

if up arrow pressed move ↑

If down arrow pressed move ↓

if left arrow pressed move ←

if right arrow pressed move →

Use the code 'Exploring selection with fruit drawing' to explore selection.

1. **Run the program. Predict what code might be used before you look inside.**
2. Read the code. Predict again.
3. Run again.
4. **Debug the code. (How do you know what it should do?)**
5. Consider which might be the 'best way' to code the algorithm?

https://scratch.mit.edu/projects/95267347/#player

https://scratch.mit.edu/projects/95267347/#player

Activity: Session 5 – Variables Objectives

Increase understanding of variables specifically:

- What is a variable
  - analogy misconceptions
  - initialisation
  - data types
  - data structures
  - scope
- Using design to represent variables
- Progression of variables
- Roles of variables

RAG your confidence on each of these

Activity: Session 5 –Activity 1 Live coding – variables to gather and display a value

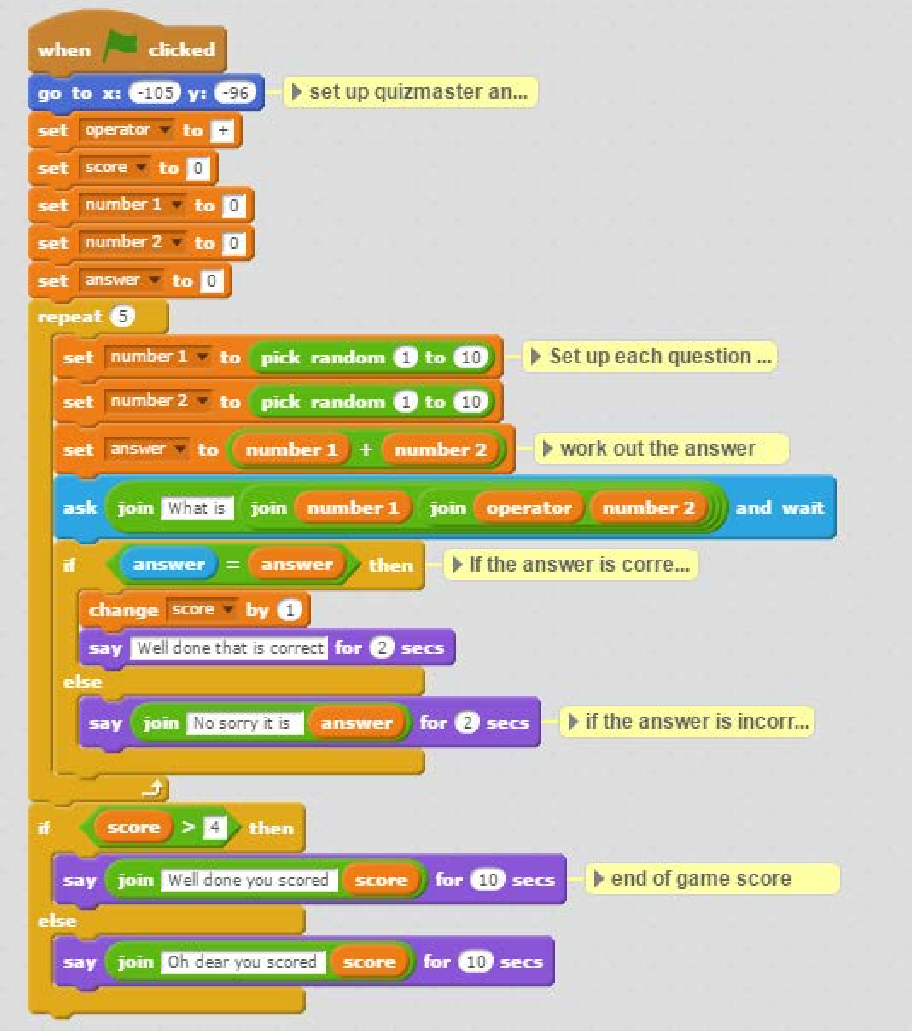
https://scratch.mit.edu/projects/167162449/

Activity: Session 5 –Activity 2 Guided discovery – variables for users to control aspects of a program

Explore this program.
https://scratch.mit.edu/projects/167543312/

- What have you discovered?
- Why did I include certain things?
- What did I want you to learn?

Activity: Session 5 –Activity 3a Remixing – variables that control a program to increase efficiency/reuse/understanding (1)



```
when [flag] clicked
go to x: -105 y: -96                              set up quizmaster an...
set operator to +
set score to 0
set number 1 to 0
set number 2 to 0
set answer to 0
repeat 5
    set number 1 to pick random 1 to 10            Set up each question ...
    set number 2 to pick random 1 to 10
    set answer to (number 1 + number 2)            work out the answer
    ask join What is join number 1 join operator number 2 and wait
    if answer = answer then                        If the answer is corre...
        change score by 1
        say Well done that is correct for 2 secs
    else
        say join No sorry it is answer for 2 secs  if the answer is incorr...
if score > 4 then
    say join Well done you scored score for 10 secs    end of game score
else
    say join Oh dear you scored score for 10 secs
```

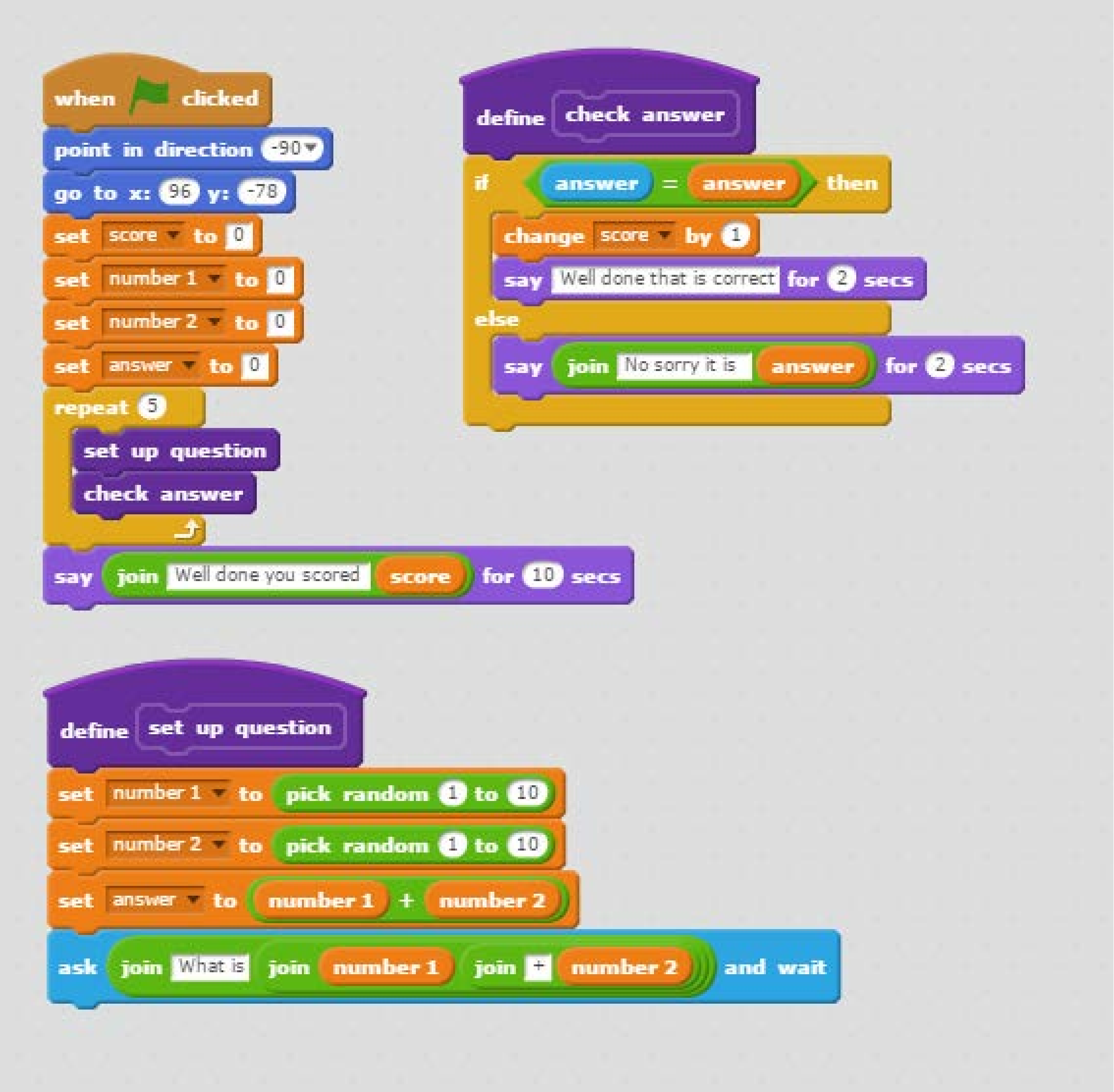
In this Maths quiz, how are variables used? Why?

1. Predict what the code will do
2. Run it. Were you correct?
3. Remix the code to
    - Add 4 subtraction questions
    - Add 4 multiplication questions
4. What will you do with the end of game message?

https://scratch.mit.edu/projects/96144555/

I predict this code will:

Activity: Session 5 –Activity 3a Remixing – variables that control a program to increase efficiency/reuse/understanding (2) - extension

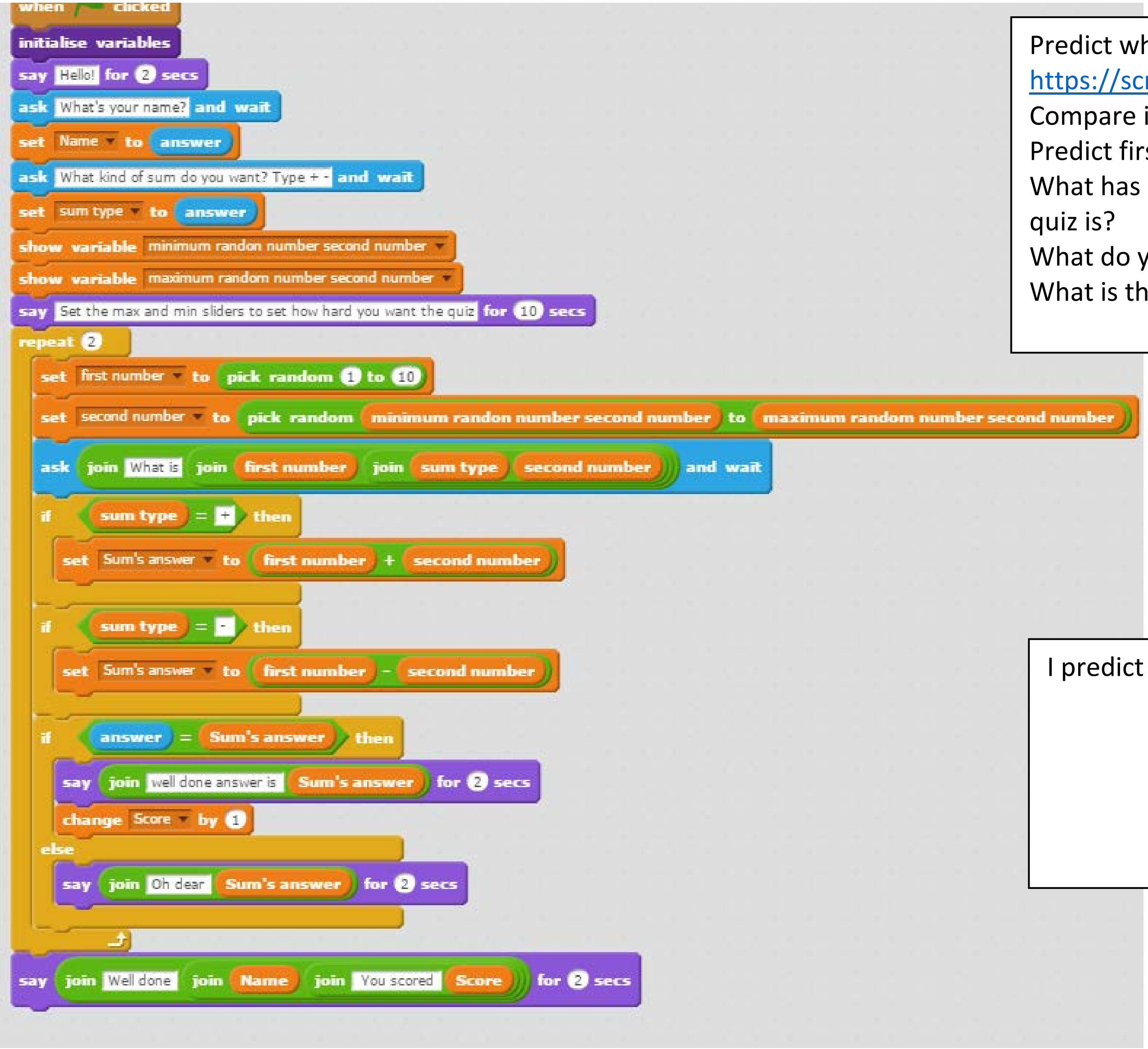


```
when ⚑ clicked
point in direction -90
go to x: 96 y: -78
set score to 0
set number 1 to 0
set number 2 to 0
set answer to 0
repeat 5
    set up question
    check answer
say join Well done you scored score for 10 secs

define check answer
if answer = answer then
    change score by 1
    say Well done that is correct for 2 secs
else
    say join No sorry it is answer for 2 secs

define set up question
set number 1 to pick random 1 to 10
set number 2 to pick random 1 to 10
set answer to number 1 + number 2
ask join What is join number 1 join + number 2 and wait
```

Predict what this code will do.
Compare this code to Using variables for other roles – example 1 (shown before).
What has been added? What is the impact?
https://scratch.mit.edu/projects/96144555/ (bat code)

I predict this code will:

Activity: Session 5 – Activity 4a – Targeted task – Predict and compare code - variables that control the flow of a program (4a)

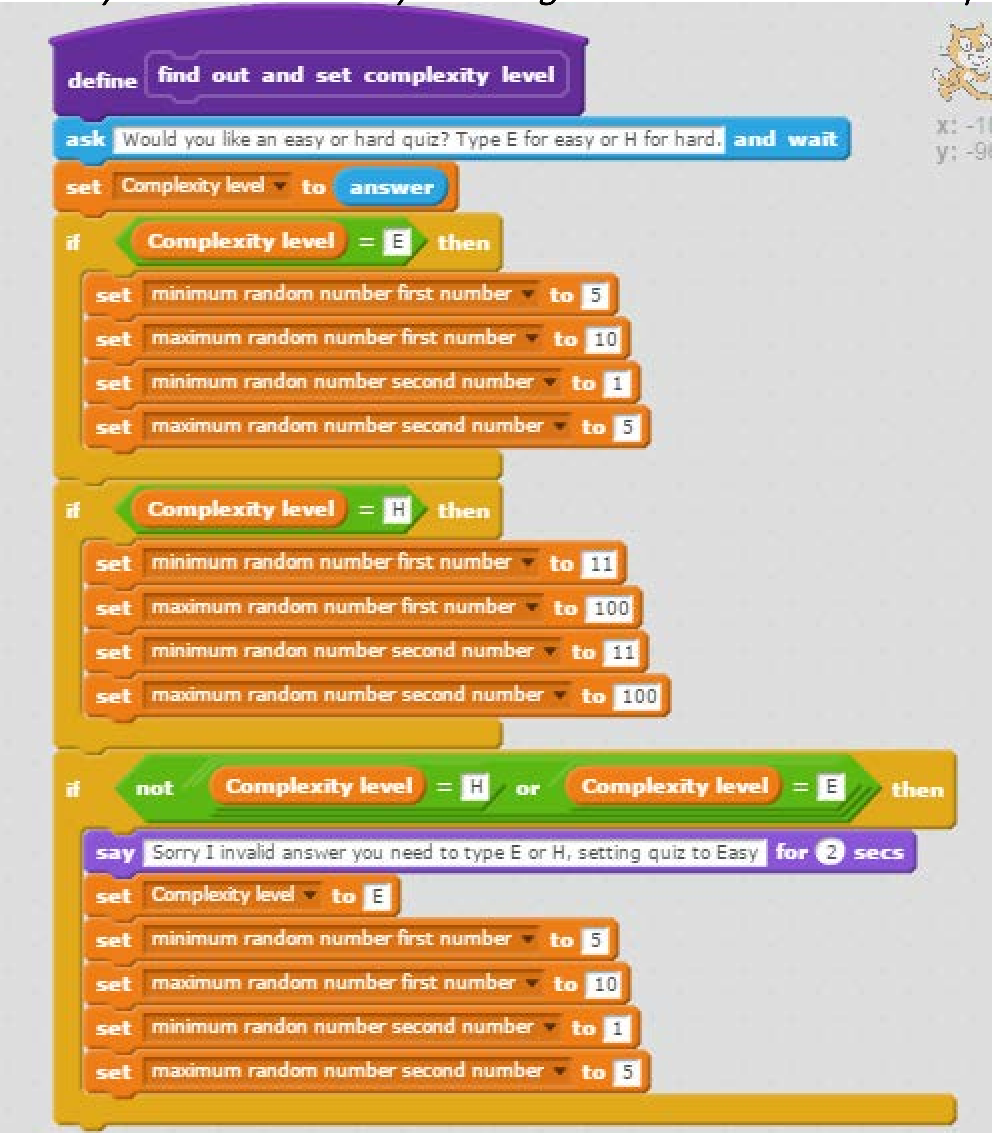


```
when clicked
initialise variables
say Hello! for 2 secs
ask What's your name? and wait
set Name to answer
ask What kind of sum do you want? Type + - and wait
set sum type to answer
show variable minimum randon number second number
show variable maximum random number second number
say Set the max and min sliders to set how hard you want the quiz for 10 secs
repeat 2
    set first number to pick random 1 to 10
    set second number to pick random minimum randon number second number to maximum random number second number
    ask join What is join first number join sum type second number and wait
    if sum type = + then
        set Sum's answer to first number + second number
    if sum type = - then
        set Sum's answer to first number - second number
    if answer = Sum's answer then
        say join well done answer is Sum's answer for 2 secs
        change Score by 1
    else
        say join Oh dear Sum's answer for 2 secs
say join Well done join Name join You scored Score for 2 secs
```

Predict what this code will do.
https://scratch.mit.edu/projects/167515468/
Compare it to example 4b on the next page.
Predict first what each program will do before you run them.
What has been added to each to control how easy or hard the quiz is?
What do you think to the differences?
What is the impact of make your own blocks?

I predict this code will:

Activity: Session 5 – Activity 4b – Targeted task – Predict and compare code - variables that control the flow of a program (4b)



```
define find out and set complexity level
ask Would you like an easy or hard quiz? Type E for easy or H for hard. and wait
set Complexity level to answer
if Complexity level = E then
    set minimum random number first number to 5
    set maximum random number first number to 10
    set minimum randon number second number to 1
    set maximum random number second number to 5
if Complexity level = H then
    set minimum random number first number to 11
    set maximum random number first number to 100
    set minimum randon number second number to 11
    set maximum random number second number to 100
if not Complexity level = H or Complexity level = E then
    say Sorry I invalid answer you need to type E or H, setting quiz to Easy for 2 secs
    set Complexity level to E
    set minimum random number first number to 5
    set maximum random number first number to 10
    set minimum randon number second number to 1
    set maximum random number second number to 5
```

Predict what this code will do.
https://scratch.mit.edu/projects/167515528/#player
Compare it to example 4a on the previous page.
Predict first what each program will do before you run them.
What has been added to each to control how easy or hard the quiz is?
What do you think to the differences?
What is the impact of make your own blocks?

I predict this code will:

Session 6 – Make your own blocks and *lists objectives*

Increase understanding of make your own blocks and lists specifically:

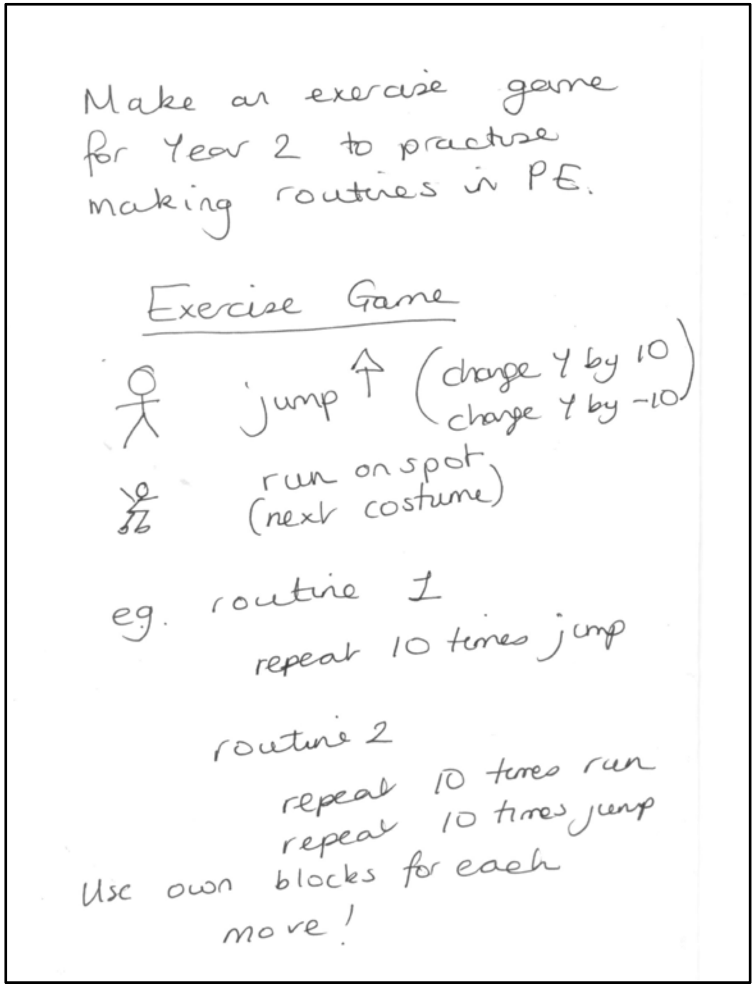
- What is a procedure – make your own block
- What is a list
- Using design to represent procedures/make your own block and lists
- Progression of procedures and lists

Would you like to Rag your confidence? How does it feel not to be asked now?

Session 6 – Activity 1 compare

Code to predict/read and modify	Code to predict/ read and fix - buggy code
https://scratch.mit.edu/projects/98664570/ Predict what the code does Run it Investigate Modify 1. Change a question/answer 2. Add a new question and answer	https://scratch.mit.edu/projects/98664691 Predict what the code does Run it Investigate 1. Fix it
https://scratch.mit.edu/projects/98664744/ Predict what the code does Run it Investigate Modify 1. Create an initialisation block 2. Create a choose 'a' block 3. Create a choose 'b' block	https://scratch.mit.edu/projects/98664866/ Predict what the code does Run it Investigate 1. Fix it
https://scratch.mit.edu/projects/98664903/ Predict what the code does Run it Investigate Modify 1. Remove all make your own blocks so that it is a single main thread 2. Remove the use of lists – so hardcode the questions and answers in if- then statements.	https://scratch.mit.edu/projects/98664969/ Predict what the code does Run it Investigate 1. Fix it

Session 6 – Exercise Game



Make an exercise game for Year 2 to practise making routines in P.E.

Exercise Game

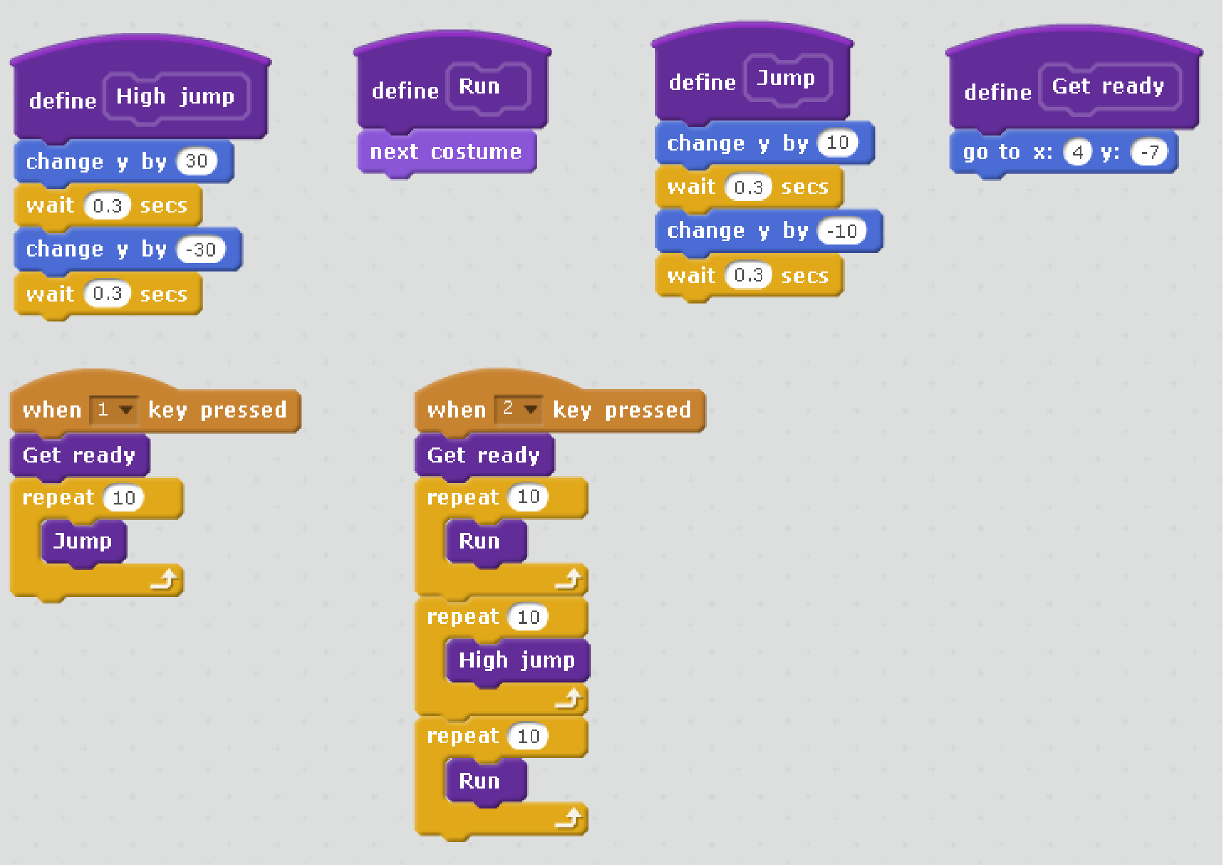
jump ↑ (change Y by 10
change Y by -10)

run on spot
(next costume)

eg. routine 1
repeat 10 times jump

routine 2
repeat 10 times run
repeat 10 times jump

Use own blocks for each move!

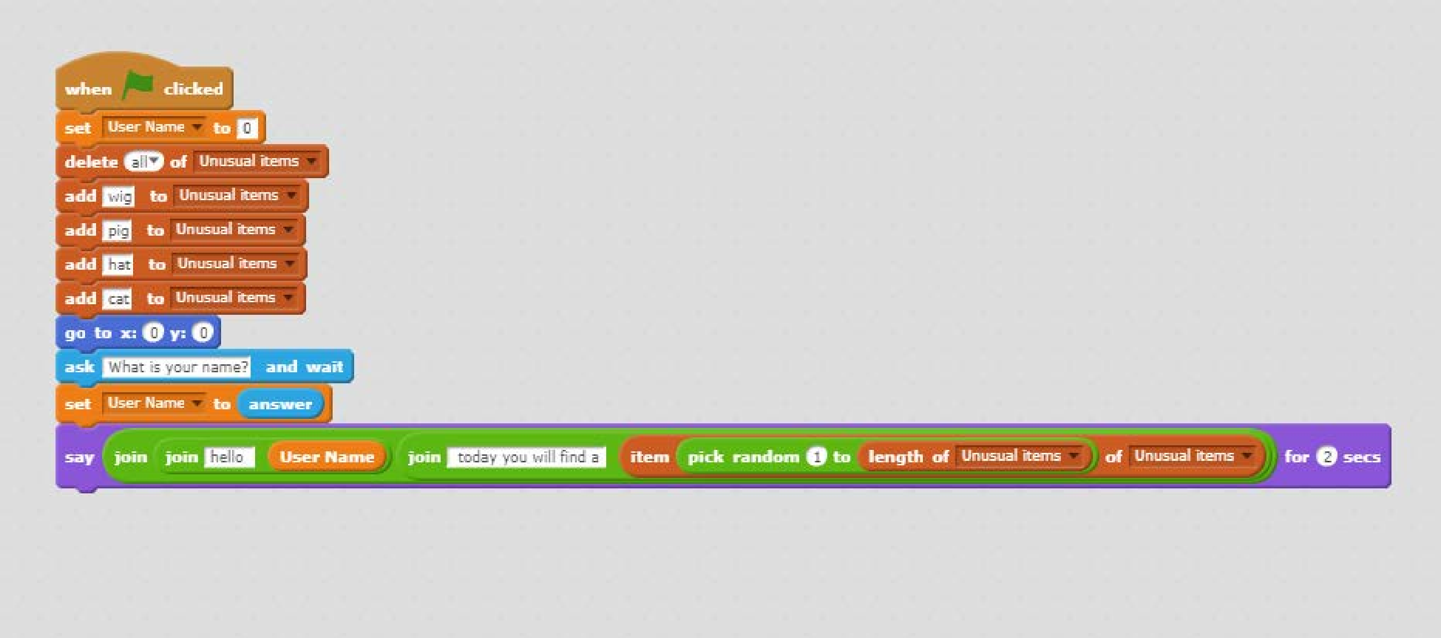


```
define High jump
change y by 30
wait 0.3 secs
change y by -30
wait 0.3 secs

define Run
next costume

define Jump
change y by 10
wait 0.3 secs
change y by -10
wait 0.3 secs

define Get ready
go to x: 4 y: -7

when 1 key pressed
Get ready
repeat 10
    Jump

when 2 key pressed
Get ready
repeat 10
    Run
repeat 10
    High jump
repeat 10
    Run
```

Session 6 – Fortune teller

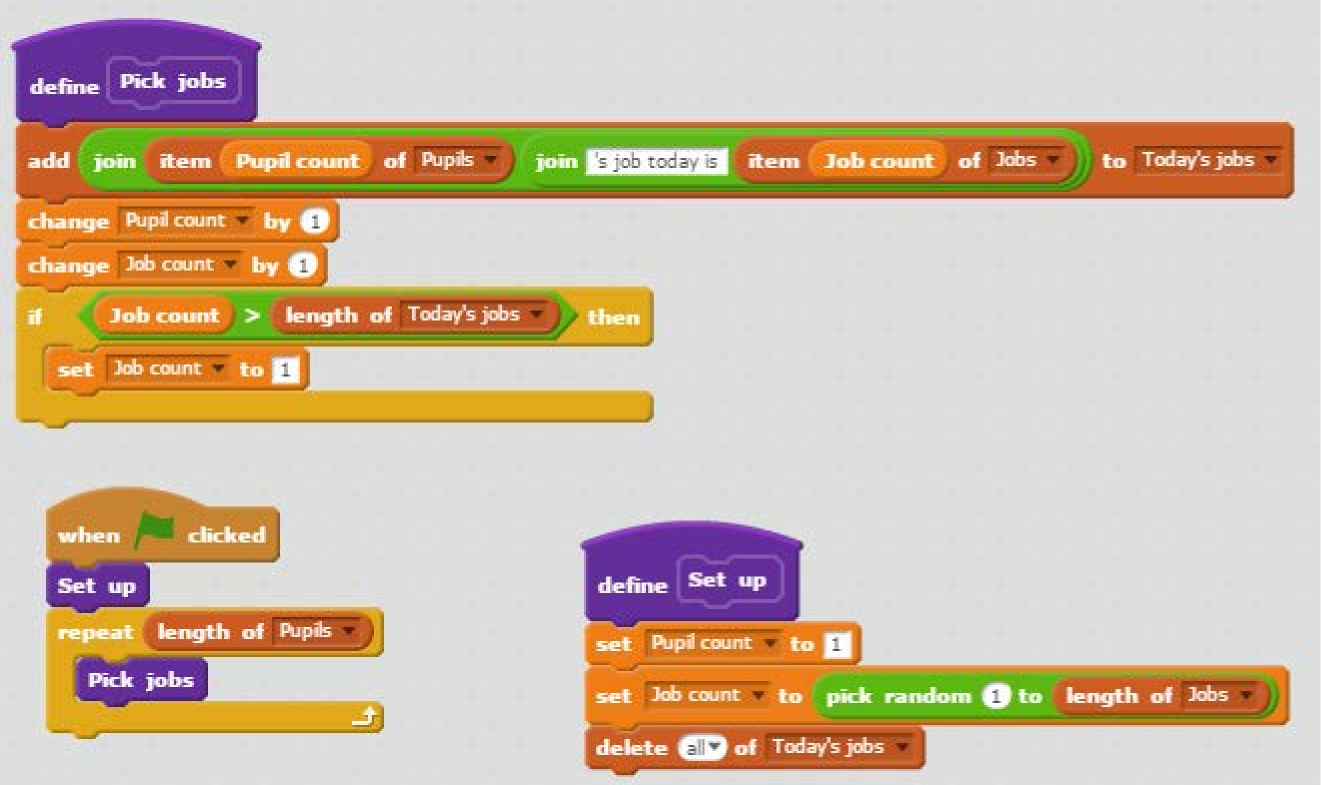
Task: Make a simple fortune teller that predicts you will find something unusual!
Let's create the design together.
Note
I am going to cover
•A gradually building up complexity and testing as I go.
•Making mistakes – resiliency (use of list block)
•Adding to a list
•Not deleting from the list for initialisation

I already wrote the code… but will show you how I did it. https://scratch.mit.edu/projects/168123593/



```
when ⚑ clicked
set User Name to 0
delete all of Unusual items
add wig to Unusual items
add pig to Unusual items
add hat to Unusual items
add cat to Unusual items
go to x: 0 y: 0
ask What is your name? and wait
set User Name to answer
say join join hello User Name join today you will find a item pick random 1 to length of Unusual items of Unusual items for 2 secs
```

Session 6 –Job picker



```
define Pick jobs
add join item Pupil count of Pupils join 's job today is item Job count of Jobs to Today's jobs
change Pupil count by 1
change Job count by 1
if Job count > length of Today's jobs then
    set Job count to 1

when ⚑ clicked
Set up
repeat length of Pupils
    Pick jobs

define Set up
set Pupil count to 1
set Job count to pick random 1 to length of Jobs
delete all of Today's jobs
```

Job picker

Allocate jobs to pupils each day

Algorithm

For each pupil name allocate a job randomly.

Make three lists
One for the names, one for the jobs and one for the job list.
Add the names and jobs by hand.
Write code so each pupil in the list has a job picked for them.

https://scratch.mit.edu/projects/167562917/#player