Chatbot Assessment Examples

Purpose
To help teachers assessing pupils programming to identify computational thinking and progression in programming concepts.

Limitations
This assessment goes alongside the module ‘Chatbot’ found in the book ‘How to teach primary programming using Scratch’ by Phil Bagge

Class Background
Year 6 Class
Previous programming experience
Prior to this module in Year 6, 10 hours creating ‘Times Table Game’, ‘Perimeter & ’Clock’ modules

Chatbot Investigation Aim
Identify where variables can be used to store information
Use the ask input to collect information
Respond to a user’s specific answer
Develop a style that mimics a human

Module Outline

Time Frame
4 hours
• Basic aims of the module were explained
• Teacher created a chatbot with the pupils explaining their thinking as they went along
• Pupils thought about information they could collect from the user and respond to
• Pupils created the project independently or generalise ideas from the support cards
Assessment Principles for this module

In this assessment all code is assessed but code that is exactly the same as the teacher example or the hint cards is only recorded on the bottom row of the infographic grid.

Computational thinking can be recorded. Mainly in this case (Ge) generalisation if a code example has been reused for a similar or different purpose. Pupils did think about what type of data could be collected which is evidence of algorithmic thinking but I have not added this to the infographic grid.

The infographic grid (below) used to evaluate the following pupils work is useful to help visualise complexity and independence.

The left column contains code that is the most simple, moving to the most complex code on the right. The bottom row is for ideas where the teacher or another student had a significant input. The middle row is where a lesser hint was provided and the top row is reserved for independent work.

A row of icons touching represent a contained section of code.

There is a rough guide to complexity at the bottom of each column below the infographic grid for this module.

In this module the teacher filled in the infographic grid after the pupils had finished but in some modules there is value in pupils reflecting on learning and filling in part or all of the grid themselves.

It is possible to assess this module without the infographic grid but as code becomes more complex it is easy to assume complexity when faced with lots of code.

The infographic grid doesn’t assess style and it is important to run programs as well as examine the programming elements.

1. found in the book ‘How to teach primary programming using Scratch’ by Phil Bagge
2. adapting an idea found elsewhere for a different purpose

The Infographic grid will be available for use soon and is the property of Phil Bagge & Les Carr
Pupil A

Has collected the users name and then used it within the program. If they had used it again later on in the program it would show even more understanding.

The age question is in the more complex column as it involves comparing the number to the variable using a < symbol. As it uses the same age example as the hint card it doesn't show fully independent reasoning.

There is a working broadcast that makes a dog appear and move around. Although broadcasts are not full procedures they are evidence of the user creating sub programs to cope with complexity.

Overall pupil A is a working within the accepted expectation for a child of her age. They are starting to experiment with more advanced concepts which will increase their attainment.
Pupil B

Bug

There is a clear bug where the pupil has set the variable before asking for information from the user to go into the variable. The pupil has identified this as an issue as they have attempted to set the same variable twice later in the code.

Bug

There is another bug here where the pupil has attempted to compare the value inside the variable before getting the user to place anything inside it.
Pupil B has two good working sections of code which compare the users answer with the authors answer and either action one set of events or another. The later errors show they are not fully secure with this and debugging these would be a good next step. Although they have attempted to use a list it has not been used correctly so no credit has been given for this. There are some really nice features when the user selects something contrary to the programmers choice such as the sprite changing to show a sad face.

Overall the user is working within the accepted progress for a child their age.
Pupil C

Has used simple sub-routines (broadcasts) to really good effect to mimic the main sprite talking. This is a lovely effect that shows, alongside the multiple use of the variable, that the student understands the universal, recallable nature of the variables they have created.

To expect the user to type in exactly 'joe sugg and casper lee' suggests that a next step would be to try this out on various users and find the limitations of the program.

The procedure icons refer to the successful use of simple sub routines (broadcasts)

Pupil C is working within average expected achievement and is ready for more complex ideas.
Pupil D

Pupil D has only attempted to use a variable once and has not attempted to reuse it.

They have a good understanding of how a basic if or else conditional selection choice works but have not attempted to generalise any of the ideas they were shown by the teacher or provided by hint cards.

There is some understanding of iteration (looping) but lots of extra show and hide blocks have been used suggesting that this understanding is limited.

The next steps will be to challenge the pupil to store their answers in variables and re-use those and find ways to make their simple loop more efficient.

In Year 6 this pupil is working below the average of their peers.
There is a bug here as mums name variable is compared to Ciara but no opportunity was given to assign anything to the variable.

However the variable is assigned later and is reused well later on in the programme.

This use of the holiday place variable really doesn't work with the next question.

There is some nice simple story telling using broadcasts to set different backgrounds.
Pupil E

Before I filled in the infographic grid I was convinced pupil E had done enough to be within the average band. However if a variable has only been assigned and reused elsewhere I have only classed this as simple use as nothing has been compared. The first selection block doesn't work and the second doesn’t use a variable in the comparison so is only a simple use. There is generalisation of the teachers reuse of the variable.

After reviewing the infographic grid I place this pupil below the average however it will not take much extra work to raise pupil E to within. Stepping through Pupil E’s program and recording what is in the variables at each stage will help Pupil E determine what is going wrong.
Pupil F

Pupil F sets all their variables back to nothing which indicated good logical thinking.

Pupil F uses the age variable to good effect both comparing it to see if it is the same as 10 and then if it is not the same checking to see if it is greater than or less than and responding to both.

Pupil F has used broadcast sub programs for the user to make a choice between different chatbots a happy script and a sad one.
Pupil F stores favourite subjects in a list although this was the example given so no generalisation in this block.

Pupil F uses multiple variables to summarise what the chatbot has learnt from the conversation.

Pupil F is the master of using multiple conditional selection blocks to provide more choices than just two.
Pupil F is clearly working above the expected average and is demonstrating a real command of conditional selection. Before completing the infographic grid I was convinced, because of the amount of code, that his work was far above his peers but after the grid I think he is above and working towards being far above.

Only a 1/3rd of his code has been copied here
Pupil G represents a unique problem. On the one hand there are very few questions, no complex use of variables and very little data has been collected or reused. On the other hand the pupil was determined to make a sarcastic remark if none of their favourite You Tubers were chosen and worked tirelessly to come up with a solution. The pupil decided independently to use multiple and blocks. The teacher only hinted about the addition of the not block at the end.

I would assess this as **within** the expected average for the age group due to the Boolean operatives used correctly and the level of independent perseverance shown by the pupil.
Pupil H has assigned a 1 to the variable sad and a 0 to the variable happy. The following code should always choose 1 or 0 and always say I feel ok today. However it actually picks 1 or 2 which makes the conditional selection block work.

However they show in this section that they do understand the use of the or Boolean choice block to check if the user is equal to or greater than 21.

Again they show a good understanding of the or block to check if either team is a user choice.

Multiple or conditions are checked here.
Another good use of the or Boolean block to check if either conditions are present.

The broadcast triggers a simple sub program which shows and later hides a bowl of cheesy puffs. (code bottom right)

This subject example is the one on the hint card but adapted enough to suggest the pupil understands how it works.
Pupil H is clearly working above the average of their peers. They show an excellent understanding of ‘or ‘and how to use a variable multiple times. The next challenge I will give would be to see if they can appreciate how and works to check that two or more conditions have all been met.
What to record

Once the work of assessing the modules is complete the teacher assesses them into

Outside (This pupil shouldn’t have attempted this module)
Below
Within
Above
Far Above

The teacher considers any pupil who falls into the outside category to be a failing on the teacher’s part as the pupil shouldn’t have attempted the module.

For this class
Below 7
Within 16
Above 8

(This class had a higher amount of above average scripts that I am used to)