Teaching Approaches

Practical Thoughts

The WeeBee engine is all about learning to code, and doing this using a text-based approach rather than a graphical-based approach (such as Scratch). We also advocate its use in Story-Writing-Coding (SWC), using code to create animated stories. All children are familiar with stories, so this provides a 'natural' base for their learning to code. In effect, a lot of the cognitive effort in coding is therefore off-loaded, and their natural inclination to tell stories is leveraged in the SWC approach. Yet, as we mentioned above, there is the issue of complexity of choosing.

So we suggest that learning should be done in a staged way, where the complexity, the power of the engine is progressively 'unfolded'. Here is a suggested sequence of activities based on the four "bookies" we have created.

Book 1

This book introduces a lot of methods, but is restricted to a single character, so there is no need to consider *synchronisation* of the actions of multiple characters, which is the hardest concept for children to grasp. That comes in Book 3. Perhaps too much is introduced here?

Making a Scene	Using the add(…) method to compose a scene
Jumping	Using the add(…) method to add a character Getting the character to jump Introducing the parameter jump height
Spinning	Getting the character to spin Investigating the spin parameter
Flying Around	Introducing an investigating the **flyto(X,Y)** method
More Moving Around	Introducing the **leapto(X,Y), walkto(X), runto(X)** and **hopto(X)** methods
Changing Appearance	Introducing the **grow(…)** and **shrink(…)** methods, and also **rest()**
Talking and Thinking	Introducing the **says(…)** and **thinks(…)** methods
Emotions == Feelings	Introducing **feels(happy)** and other emotions.
Putting it all Together	Exploratory synthesis of methods learned so far

Book 2

This book stays with using a single character, but it introduces the concepts of "props", and how these can be picked-up, transported and put-down. In addition, some more "move-at" methods are introduced, such as flipping and squishing. One useful animation technique is introduced, making things exit and enter the scene. Finally, children are shown how to change scenes (the background) and they are encouraged to use this to create a short story. There is a movement from instruction to application and exploration.

Working with Props	Introducing **flyto(prop);** as a alternative to **flyto(X,Y);** then introducing **pickup(prop);** and **putdown(prop);**
Collecting props	Open-ended activity to apply the above
Flipping and Squishing	Introducing the vertical flip, **flipV();** and the horizontal **squishH(scale);**
More squishing	Introducing and applying the vertical squish **squish(scale);**
Things Flying Apart	Creative activity, applying **flyto(X,Y);** but making the target (X,Y) off-screen, e.g., (-100,30), (100,50), (10,100).
Changing Scenes	Use of **setScene(image);** Creative activity
A journey	A creative activity where a character moves through several scenes. The first coded story.

Book 3
This book focuses on coding two characters and introduces the concept of using “tuples” of lines of code (here “pairs”) to synchronise the actions of two characters. This is the most difficult concept the children will need to master. But they can do it!

Two Characters	Teaches how to use “tuples” (pairs) of lines of code to get characters to **jump** in sequence or together
Two Characters feel unhappy	Continues with programming in “pairs”, but with minimal guidance. Elements of a story here.
Sports Day – The High Jump	Continues with minimal guidance, uses a meaningful context of a sports day
The Collectors	Programming in “pairs” where the characters gather their favourite objects.
Strange Behaviour	A sequence of actions is presented, introducing Grog’s magical powers

Book 4
This book continues working with just two characters, but it is intended to help the children apply what they have learned.

A Night Scene	Here, children will be asked to code a story where Pip meets a flying saucer. The activity is quite guided.
Camping	Children create a camping scene in the woods, near a lake and cook an egg for their supper.
Terror on the Moon	Pip is visiting a friend who lives on the Moon. But the horrible dragon Drax appears and makes Pip feel really scared. Suddenly, a flying saucer swoops overhead which scares Drax. Let’s see what happens.
A Conversation	Two of our friends return to a place where they played when they were young. Not much has changed, and our friends are happy to return here. They start a conversation. At first they share memories. They notice things in the scene and they describe them. Then the conversation moves on.

Theoretical Ramblings

I have employed three approaches to learning and teaching, both to provide instruction and also space to allow the children to explore. These comprise (i) the use of ‘Bookies’ (Dutch word, diminutive, meaning ‘little books’), (ii) the use of Work Cards, which are essentially the same as Bookies, but presented to children as discrete steps, (iv) a “guided bricolage” approach. Teachers have indicated preference for various approaches. But before we get into details, it may be useful to take a bit of a theoretical approach to understand what we are trying to do.

First, it seems to me that learning and teaching should be situated on a cline of **instruction – exploration.** In **instruction**, we set up various activities for our students: Activities that (i) *introduce* new concepts, (ii) *re-inforce* these concepts, (iii) *apply* these concepts and (iv) *extend* these concepts. The trick is how to sequence these different flavours of activity. In **exploration** we allow the children to take control of what they do; this involves setting their own goals and working to achieve these. From the teachers’ point of view, an instruction-based approach allows the teachers to prepare materials in advance, and to control their expectations of what the children can achieve, and the problems they are likely to have. In effect, these teachers are crystallising their knowledge and understanding of the material into the instructional materials, so they do not have to draw it down from memory, in the class, on the fly. This is a very safe approach, but is constrained by the prepared materials. The exploration-based approach requires minimal preparation, and effects minimal constraints, but puts a heavy load on the teacher who must work in real time, providing guidance of “what next”, solving problems and managing a very busy class.

Whichever approach we take or a mix of the two, situated at some place on the cline, we do face a real cognitive issue; this is the question of how much choice we give the children. The engine contains a lot of choice, there are many backgrounds, elements of scenery, many Props and several Actors, and children can introduce their own created assets. Each time a child writes a line of code, they must make a choice; do I use an Actor or a Prop, or add some more scenery? Which method do I use? Which characters do I get to interact with each other? Too much choice can lead to frustration; a child may simply not see how to start coding an animation because there is too much choice. I have seen this on several occasions, and have firmly come to believe that one important aspect of pedagogy (at any level) is the management of choice. I suggest that this should be the guiding principle behind any learning and teaching approach we take, whether instructional or exploratory.

Bookies

These are given out to the class and children work through them at their own pace. They comprise activities which children ‘tick off’ on completion, and include a linear sequence of the four activity types mentioned above. At the end of each bookie, children complete a sheet where they say ‘I can do …’. Teachers’ time is spent on helping the children, and providing focus points during the lesson to address common issues. Several teachers like this approach, since they can easily monitor children’s progress, when the class is over, by recording the ‘tick off’. Also, the learning and teaching is sequential, which these teachers prefer. Most of their cognitive effort is spent in preparing the materials, and they can focus, in class, on helping the children correct their errors.

Work Cards

These contain essentially the same material presented in the bookies. Each child is given a work card which can be completed within a few minutes; the child requests the next card from the teacher. Some teachers prefer this since they can record the progress of each child during the class, and they can check each child has really understood the task presented on the card through a short conversation. Others, who tried this approach, reported that it is not feasible, even disruptive, since their time is divided between working with the children and managing the cards. Interestingly, one teacher preferred this option, since it encouraged children to be physically active during a class. Only one class I worked with was organized so that the TA managed the flow of cards and the teacher and I helped individual children, but this class worked extremely productively. The main advantage of this approach is that it *can be non-sequential* where teachers can choose which card is appropriate for which child.

Guided Bricolage

Guided bricolage as an initial learning and teaching approach has good results, but not for all children, especially boys who seem to benefit from direct instruction.

Coda

Over the past four years I have worked in four local primary schools, teaching mixed classes of around 30 children; Yrs1&2, Yrs3&4, Yrs5&6. The class teacher was always present, though there was variable level of engagement. I have also used the engine with secondary school students, during visits to the University Computing Labs, both years 9 and 12. Then there are my own 3rd-year undergraduates who have used the engine over the past three years. So what I write here is a mixture of all of my experiences. One over-arching experience is that children and students of all ages really enjoy working with the engine; they remain on task, are being creative, and are happy to get things working. From my time in Primary, I think Yrs3&4 have produced the best work. While Yrs1&2 enjoyed working with the engine, they often had poor keyboard skills which slowed them down. This lack seems to be an emerging trend, and recently a secondary teacher reported that children coming into Yr7 had impoverished keyboard (and general IT skills). Yrs5&6 also enjoyed working with the engine, but gender issues seemed to surface here; boys were less interested in Story-Writing-Coding, and spent a lot of time trying to ‘break’ the engine, e.g. by using silly values for parameters, and often referred to video games they had played. Perhaps a more boy-targeted approach may be needed, providing more boy-oriented assets (such as transport)? While I am by no means a primary expert (and do not have QTS at any level) I understand that girls will learn just as well with materials targeted to boys.