Human Centred Design Techniques Report

June 11, 2018

Group 14

1 HCD Techniques

1.1 Initial Research into Problem Statement

We decided to address the problem of a lack of quality programming education for Key Stage 3 students. After initial research we found that many schools are struggling to keep up with new the UK computing curriculum (figure 1), where students are now taught Computing from 11-14 as a compulsory course, as there is a lack of skilled programming teachers.

Many existing platforms either do not provide facilities for teachers to see student solutions or provide feedback, or don't give students any direction when attempting to teach challenging concepts.

We envisaged an engaging web application to help students learn difficult progamming concepts that are not taught by other platforms, without discouraging them with unintuitive syntax. This platform would allow teachers to set exercises for their classes to complete and track their students' progress through the app.

1.2 HCD Research Methodology

We started the project by creating: personas for our target audience (Persona section), a stakeholder map showing which groups of people our app will have an effect on (figure 2) and a mood-o-gram for a teacher's typical programming lesson without our app (figure 3).

Each week we employed an HCD methodology, roughly adhering to the following cycle:

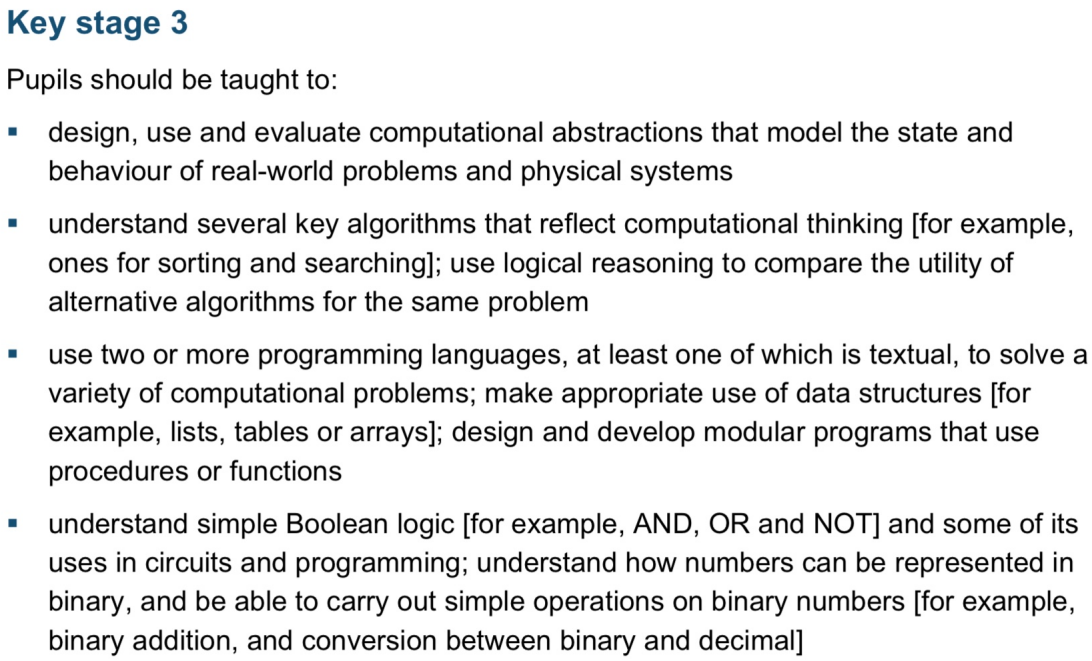
- Initially, we collate the previous week's feedback. We use it to determine what existing features have been most useful to the users and what new features would give them the most benefit in the coming week.
- After determining the key features, we develop a mocked up representation of the features.
- Our primary user tester is a teacher, Lucy Glanfield. We present our mocked up features each Tuesday and let her play with them. We note her verbal feedback and observe her interactions to gain valuable information about the current state of the feature. This feedback helps us decide which features are useful from the teacher's perspective. Lucy has also been able to demo the application with some members of her Key Stage 3 classes so that we can also get feedback on the student's user journey.
- We aim to implement/improve the most pressing desired features by Thursday morning. This micro-iteration allows us to receive more user feedback and to improve the functionality from the user's perspective.
- Thursday and Friday are spent refining new features in preparation for the following week.

2 Appendix

2.1 Personas

- Nancy, 34, Chemistry teacher. Nancy has recently been asked to teach programming to Key Stage 3 students due to the new curriculum and a shortage of skilled Computer Science teachers. She has been given access to the IT suite each week and is looking for an intuitive digital tool to help her get the key programming concepts across to her students.
- Mark, 12, student. Mark is not particularly interested in technology (he enjoys Biology and wishes to study Medicine at university), however as part of the English curriculum he is required to attend a class teaching many of the core Computer Science concepts. Mark wants an engaging learning platform that does not require any prior knowledge of programming while still teaching him everything he needs to know.

2.2 Figures



Key stage 3

Pupils should be taught to:

- design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems
- understand several key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem
- use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions
- understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming; understand how numbers can be represented in binary, and be able to carry out simple operations on binary numbers [for example, binary addition, and conversion between binary and decimal]

Figure 1: An extract from Comuting curriculum

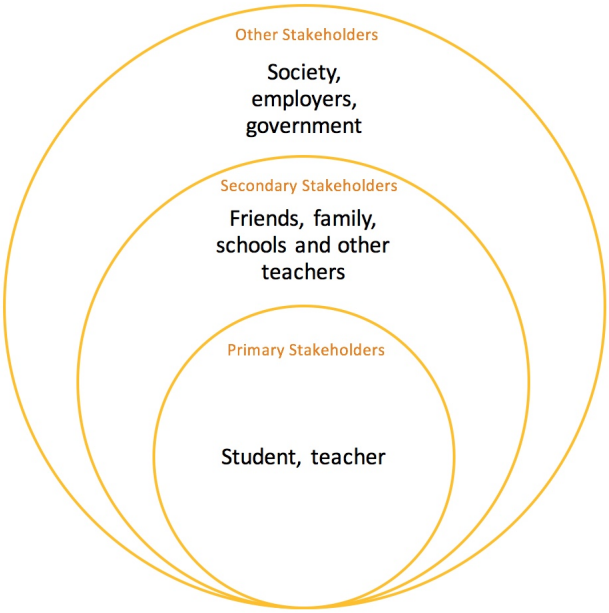


Figure 2: A stakeholder diagram

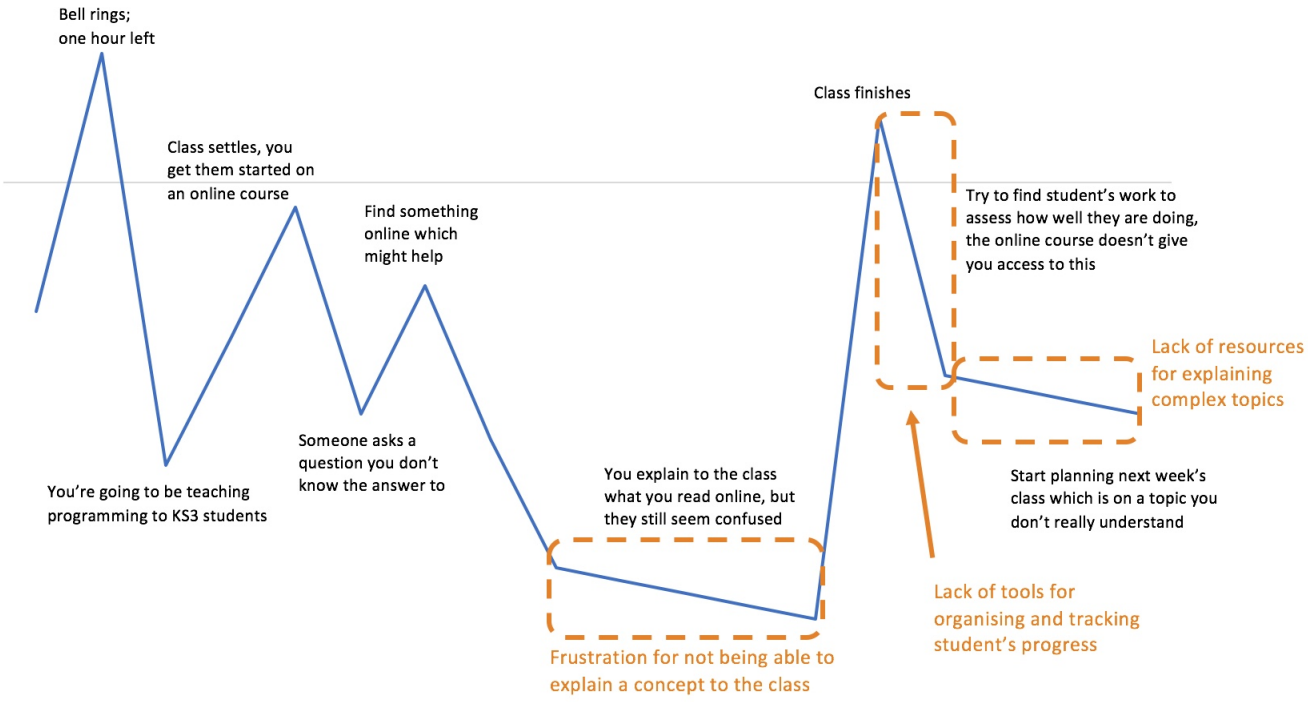


Figure 3: A mood-o-gram for a teacher's typical programming class

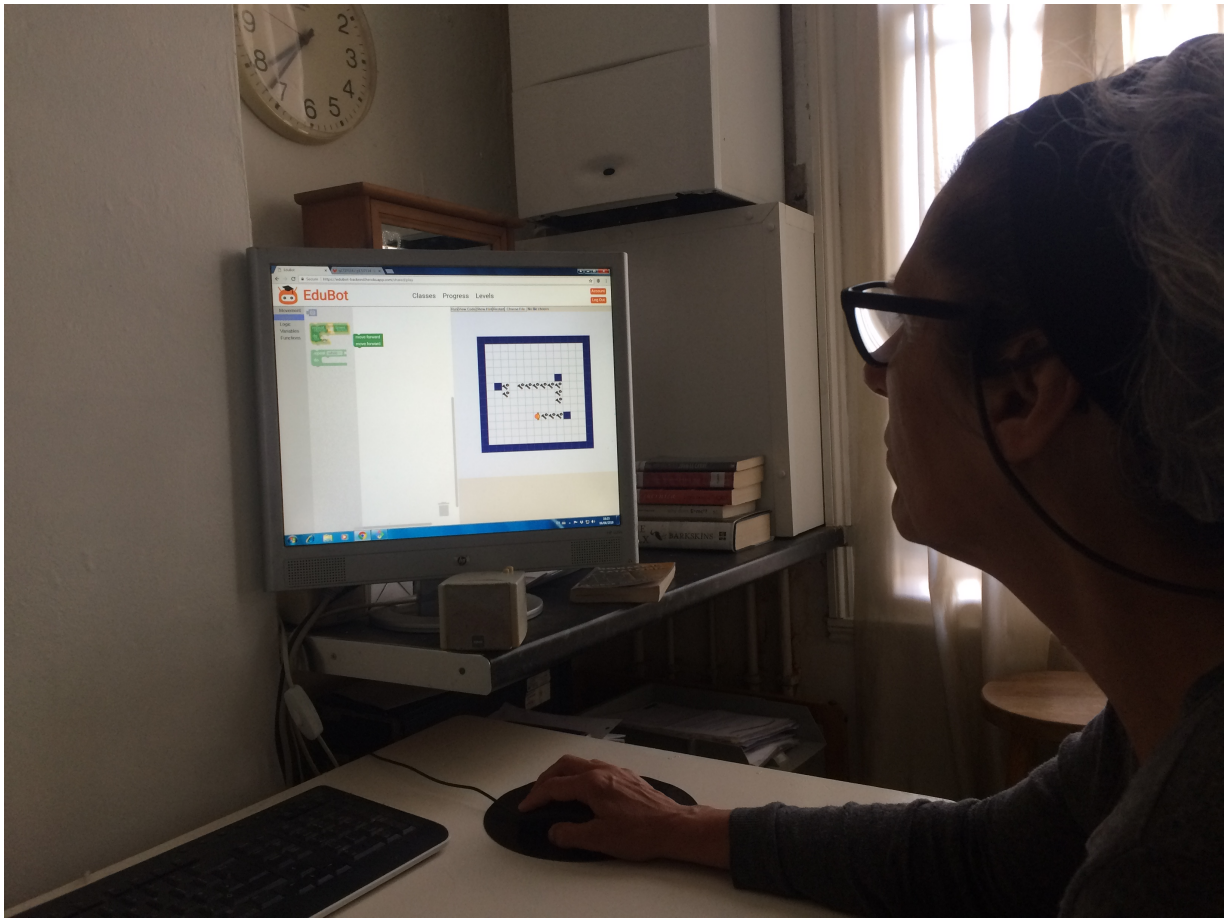


Figure 4: Getting feedback from a potential user

2.3 Diagrammatic Evidence of User Feedback

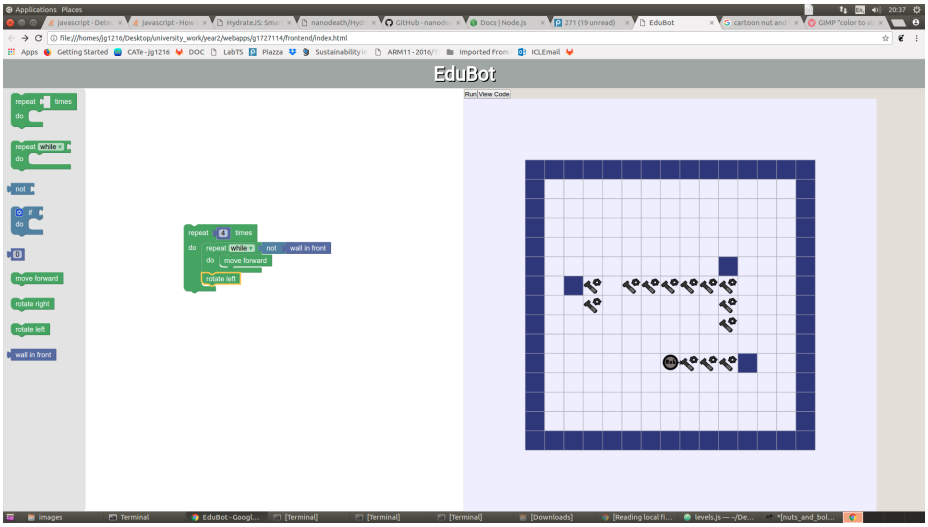


Figure 5: First iteration

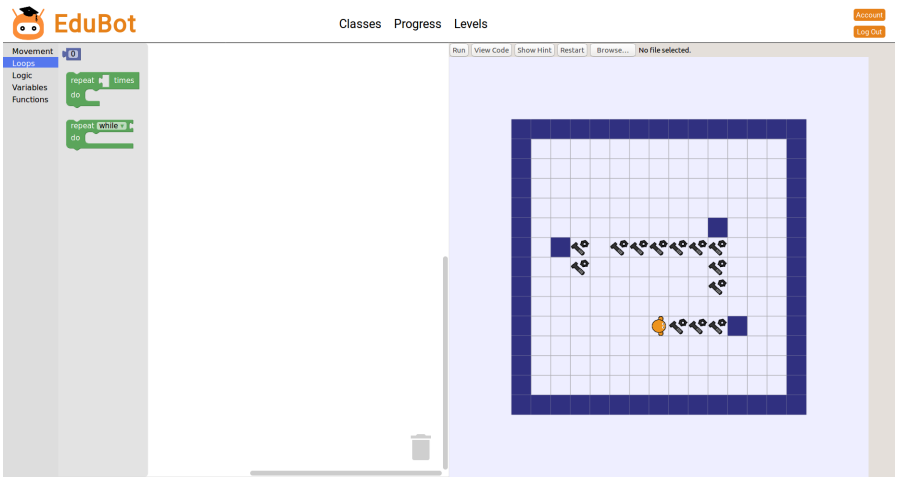


Figure 6: Second iteration