Human Computer Interaction (HCI) Concepts Followed and Design Decisions Explained

A Report by Dylan Morroll

During the design, development and testing of the program I always came back to one core principle. That was that I followed a user centred design (UCD) approach to my project. My project was to create an interactive board game, the interaction in the form of questions proposed to the user, where they then select an answer and the state of the game was updated to reflect their choice. This update in the game state was then reflected back in the user interface, so that the process is completely interactive with the user. The project tested both my file handling capabilities and my knowledge of HCI as I had to design a program suitable for a specified target audience. My file handling capabilities were demonstrated in the project itself, along with my ability to follow HCI concepts, however I am now going to explain in depth what concepts I did follow during my project.

As I mentioned at the beginning, during the project I always came back to a UCD approach to this project. As it was aimed at a younger audience of eleven to twelve year olds, the GUI had to be specifically designed with that in mind and I had to continually assess my work against that criteria as opposed to what I would like to see in such a game. An example of this is my design decision to create and incorporate a board of my own design into the game. Whereas if it was aimed at an older audience I would probably not include such an aesthetically pleasing and interactive board, if there is even a market for learning games aimed at an older audience. However I know that most kids of the age of the target audience love playing video games, as I once did myself and my younger brother does now, and so I decided to try an approach that interacted with the user as if they were playing a video game.

Another reason for this decision was that younger kids usually don't have the motivation to do their compulsory homework, never mind willingly give up their free time of playing video games or with their friends to revise for exams, so I tried to recreate an experience that felt the least like revision as possible. This board introduces a level of competitiveness, where it feels like the users are playing a video game, with their character making progress across the land when they get it correct. I also included a path that wasn't a single straight line, so that their progress along the path like a journey or quest, as opposed to a simple measurement of their progress which I felt a single line across the screen would have been. This was designed this way so that children playing would have incentive to play and want to play for reasons other than revising. Also with the game supporting multiple players, it allows for kids to challenge their friends and see who's the best, as kids so often love to do.

I also focused a lot on usability throughout my design. I wanted to make a program that was very simple to use as kids don't want to be wasting time learning how to use something, they simply want to use it. More technical programs often utilise a complex variety of functions and so have potentially thousands of different customisation options which take time to get used to. However with my program I made the design choice to remove all menus from the program, so that the user

knows that if something isn't on screen, they don't have to go trawling through any menu system to find it, they know straight away that it isn't a function of the game.

When thinking about aspects of my program like this, I factored in to my consideration trying to avoid as many possible usability problems (UP) as possible. For all aspects I aimed for minor usability problems, so that when testing the program the only thought I ever had was along the lines of "this could be easier, but it works fine". An example of this is the buttons I used for selecting answers to a question, originally when the maximum number of answers had been selected the user could not select another button until they had deselected one of their choices. However when testing my project I often got frustrated when I clicked the wrong answer, then would try to select the right answer without success, before remembering I had to deselect one of my original choices before continuing. Part of the frustration arose from the lack of efficiency in performing the task, but also because this aspect of my interface didn't apply with the Jakob Nielsen's 4th heuristic that summarises to be there should be consistency across products of a similar nature. Most options that have multiple selections, deselect the first picked option when a new options is chosen that exceeds the limit (such as radio buttons on a form), so I decided to implement this in my program so it felt more intuitive. Another reason was that a common measure of efficiency across interfaces is task completion time and I thought that this would help speed up that process as the user wouldn't encounter the situation I had, which cost me time.

However, the heuristic I took into consideration the most whilst designing the game was Nielsen's 8th heuristic that summarises to be that the user interface should be aesthetically pleasing and minimalistic. This was taking into consideration considerably throughout my menu system, where I decided to only include a few key buttons that were essential to performing the functions and labelling them intuitively so that no further explanation was needed. Due to this the menu system is very easy, intuitive and quick to navigate. Furthermore I had a minimal, but professional and aesthetically pleasing background that contrasted the buttons with their thick borders, so that they stood out and were pleasing to look at. This also shows on the actual game screen, where I tried to keep each section fairly minimal and the layout straight forward, so that there weren't any distractions and you could figure out how to play the game as soon as you start.

This was also considered when taking into account usability problems, a common measure of effectiveness is the number of errors encountered whilst using the program, also known as error rate. So for my program I decided to make it near zero. As said by Nielsen's 5th heuristic, even better than good error messages is a careful design which prevents a problem from occurring in the first place. So for my program the only error message I have display is when a user tries to save a file with a name that's already been taken by another save, apart from that there are no other error messages displayed to the user. I decided to set up my program in a way where it was impossible to enter information incorrectly, for example the submit answers, load saved game and delete saved game buttons are all greyed out until the valid requirements have been met.

This leads to an intuitive design, where a user won't attempt to use a button and have an error message displayed, rather pick up on the fact they have not yet fulfilled the requirements to use that button as it is greyed out and a greyed out button is commonly associated with an option that can't be selected. When designing this I took into account that kids might not have enough experience with

computers to be familiar with this feature, however I believe with the increasing amount the younger audience uses computers it works fine. Children are taught to use computers from an extremely young age and pick up on trends quick. Furthermore I believed this design would allow for children who don't read all the information on screen, such as the information below the question about it being multiple choice, so would try and submit a single answer.

This was evident when I asked my little brother Faron, aged 10, to test my program. When coming across his first multiple choice question, he selected an answer and moved his mouse to the submit button, however within a second he said "oh, there's more than one answer". It was too fast for him to have scanned the screen to find the relevant information, so I believe he inferred it from the greyed out button.

For each screen I also included a back/ cancel button, so that when a button is incorrectly chosen the user can go back to the previous screen and choose again. This complies with Nielsen's 3rd heuristic, user control and freedom. Kids are often quick to act without thinking a great deal, so they often make mistakes as they try to quickly perform tasks without taking in all the information on screen. These escape options allow the user to take back any rash decisions or mistaken clicks and return to the previous screen. My buttons also complied with Nielsen's 4th and 6th heuristics. Each button has an intuitive name of which the button's purpose can be easily inferred, alongside this where possible each button has a name common for its purpose, such as save, load, delete and cancel all do what you'd expect them to do.

All this makes the system not only usable, but also increases the user experience of the product. Its simple design allows the user to fly through the menu and from starting the program it can be as little as 3 clicks for a user to start playing the game. When choosing their character, the image of the character is right there in front of them, no names or numbers, they simply click the character they won't and that represents their position on the board for the rest of the game. This intuitiveness and lack of ability to make errors means the user doesn't feel like he is doing something wrong or gets frustrated, he feels competent with his abilities and aesthetically pleasing design encourages positive emotions in the user.

After some testing there were some features I changed. One of these was the design decision to remove the text "Question:" that originally preceded the question text on the game screen. I made this decision as I believed the question text alone with the list of answers to choose from was intuitive enough to allow the user to understand what the text represented, so I removed the text to free up space on screen.

If I had been able to, I would have incorporated more external resources when creating my project. During the design of my program I would have created paper prototypes of my project and showed them to groups of 5 children. Paper prototypes can be surprisingly effective and a visual aid allows kids to point and say what they don't like about my design of the product. Furthermore a group of 5 children will not get distracted from the task at hand which a larger group might. I wouldn't have created a implemented prototype of my design as I believe the children would be unlikely to able to appreciate that it is an unfinished product, so would be likely to report faults and things they don't like about the product that are a result of it being a prototype.

Once I had my program I would have tested it again on a group of roughly 5 children. A whole class would most likely result in children misbehaving and messing

with other children's computers, which would make the data unreliable. Furthermore kid's won't want to detail what they're going through by taking notes and would be unlikely to remember any faults they encountered during the process, so a testing technique where I can interact with the user's testing the product as they do so would have been most appropriate. Because of this I would have been most likely to choose a think aloud testing technique, where I could listen to the children document their thought process and problems they encountered as they went along. I believe this would have been particularly effective as kids are very liable to saying what they think, exactly how it is, so the feedback would have been very useful and informative.

I wouldn't have included any advanced data recording techniques or user experience evaluation methods during my testing phase. This is because this project was fairly small and a lot of these methods can be very costly, both in time and in money. Furthermore a lot of data recording techniques are large scale, such as questionnaires, which children would be unlikely to be willing to fill out. Although, however, I can appreciate that if it were possible at little cost, a face recognition technique could be particularly effective as once again, children's faces are often quite exaggerated in comparison to adults, so the data would have been rather informative. As I am the only developer on this project, there would have been no need to produce a report on the data received. I could perform changes to design and development in an agile manner as I progressed through creating the project, then testing my new design if I felt it necessary.