MedienPädagogik

Zeitschrift für Theorie und Praxis der Medienbildung

www.medienpaed.com

ISSN 1424-3636

Jahrbuch Medienpädagogik 4.

Zweitveröffentlichung aus: Jahrbuch Medienpädagogik 4. (2005) Wiesbaden: VS Verlag für Sozialwissenschaften. Hrsg. v. Ben Bachmair, Peter Diepold und Claudia de Witt.

New Models of Learning for New Media:

Observations of Young People Learning Digital Design

Rebekah Willett

1. Introduction

There are numerous discourses that seek to define the relationships between young people and digital media. These discourses have different and sometimes contradictory ways of constructing learners and the learning environment (Facer et al. 2001). On the one hand there are panics around new media which position children and young people as being at risk from the dangers of digital technology. In this view children are in need of careful teaching and controlling, as they are unable to learn the correct and safe way to use digital technology on their own. In complete contrast, there are discourses around new technologies which position children as ready learners and technology as offering endless easy-to-use resources for worthwhile learning. This latter view of children as „natural cyberkids' overlooks many aspects of learning and digital technology, not least the socio-cultural aspects of learning or the possibility that there might be a developmental progression of skills related to learning new technologies.

These discourses are echoed in the various pedagogical approaches to children and digital technology. The model of learning known as „constructionism", developed by Papert and colleagues at MIT (Kafai/Resnick 1996) in relation to children's learning of a simple a computer programming language called Logo, has echoes of the „natural cyberkid" discourse mentioned above. According to the model, the more time children spend building on the computer (with the computer doing the teaching) the more the child will understand the programme and therefore develop logical thinking skills. In this model, children are employing new ways of learning via computers; ways of learning which are non-linear and contradict many models currently in practice. The data from the study discussed in this article include times when young people play around on a software package, such as Flash animation, and learn through trial and error guided by particular elements on the software, such as pull-down menus. This way of learning aligns with the constructionist model. On the other hand, popular pedagogies found in many schools are based on linear developmental models that define levels which children progress through given the appropriate environment. The work of Vygotsky (1962) is used with teachers to discuss the role of verbal interacti¬ons in learning. A model of learning based on Vygotsky emphasises the im-

Willett, Rebekah. 2017. «New Models of Learning for New Media: Observations of Young People Learning Digital Design». *MedienPädagogik*, MedienPäd.Retro: Jahrbuch Medienpädagogik 4 (2005), 127–144. doi:10.21240/mpaed/retro/2017.08.08.X.

portant role of a more able peer or teacher in assisting a learner's development. Using Vygotsky's terms, each child actively participates in learning within his or her ‘zone of proximal development', that is, the distance between the real and potential levels of development. According to Vygotsky, as the child engages in his or her learning, and with assistance, modelling of actions, and pointing out of discrepancies, particularly through verbal interactions, the child internalises desired actions. A Vygotskian model aligns more closely with the discourse which constructs children as at risk of time wasting (amongst other things) if not carefully instructed. In the data being discussed in this article, there were times when young people were unable to progress with their designs without assistance, and the assistance was most effective (in terms of internalising the learning) when the tutor modelled the skill, discussed what the problems were and engaged with the learner on his or her level. Finally, with the learning of digital technologies taking place in informal settings such as homes, there has been considerable interest in contextualising learning and looking at different styles and forms of learning (Coffield 2000; Lave/Wenger 1991). Lave and Wenger's concept of „situated learning“ focuses on looking at learning as a type of social interaction, rather than a cognitive activity. This concept can be applied to the learning of digital technology and cultures, with young people acting as „apprentices“ as they learn the language, skills and discourses from „masters“ of digital technology. This article will include descriptions of times when the tutor of a class acted as a master, using a variety of sophisticated concepts and language to immerse the young people into the world of digital graphic design.

The study discussed in this article, „Shared Spaces: Informal Learning and Digital Cultures“[1], was designed as a way of engaging with and examining some of these discourses surrounding learning and digital cultures by looking at how children and young people use digital technologies in more „informal“, out-of-school settings. Its aim was to develop ways for schools to engage with the new knowledge and experiences digital culture can offer young people, as well as drawing on the informal styles of learning which characterise young people's out-of-school experiences with technology. Our data was collected at an informal education and arts centre, WAC, in north London which runs various arts activities on weekends for the young people from low income families (www.wac.co.uk). This article will focus on one curriculum initiative established on the project, a games making class for children age 9-13. I will first give a brief description of the class and describe how software was taught and learned. Then I will discuss issues around technology and pedagogy which are highlighted by the study.

[1] The „Shared Spaces“ project ran from October 2001 - December 2002 and was funded by the Esmee Fairbairn Charitable Trust. The project was based at the Centre for the Study of Children, Youth and Media, Institute of Education, University of London. The directors of the project were David Buckingham and Julian Sefton-Green. My thanks to them for their contribution to this article. More information is available on the project website: www.wac.co.uk/sharedspaces.

2. The Games Making Class

The aim of this class was to see how computer game production could be used as a way of engaging with young people's experiences of playing computer games. This aim is founded on two principles: 1) the importance of drawing on young people's culture and 2) the need to shift away from the prioritisation of analysis over production in media studies. The incorporation of media culture in schools is seen as a way of allowing young people to express themselves, not just as students, but as social individuals; and it gives teachers more space to draw on varying cultures, personalities, and values. Furthermore, in a constructivist sense, teachers are seen as having the opportunity to build on pupils' previous experiences and knowledge, helping them to make sense of the culture surrounding them and extending what they already know. There are, however, a number debates about how to approach the study of media in schools. One of the debates in the field concerns the balance between analysis and production. In a conventional approach, there is a strong emphasis on the critical analysis of media. Analysis is sometimes seen as a way of „inoculating" young people against the dangers of media (for example, absorbing negative ideologies presented in media). However, as Buckingham et. al. (1995) write, „There is a fundamental difference between the "passive" knowledge that is developed through critical analysis and the "active" knowledge that derives from production" (p. 12). In our project, production is seen as a way of allowing young people's passive knowledge of media to be made active. The emphasis is on seeing the product as part of the process of analysis. Therefore, the process of making a game is seen as a way of allowing for a balance between playing games and developing conceptual understanding of games.

The games class had ten participants (all boys), and it ran from September 2001 to June 2002, meeting Saturday mornings for two hours. The class had four phases which aimed to reflect back on each other: analysis and critique of games, designing games and game products, learning software and creating games projects. During the analysis stage, the boys looked at the games they play and discussed them in terms of key media education concepts such as quality, genre, representation, narrative structure and characters. Having analysed games in context of their experience of playing, the boys started designing their own games. They first made „story boards" for their games on paper (mapping out a sequence of events) and then designed a character and environment for a game on the computers. After designing these components on the computer they created posters advertising their games (see figure 1).

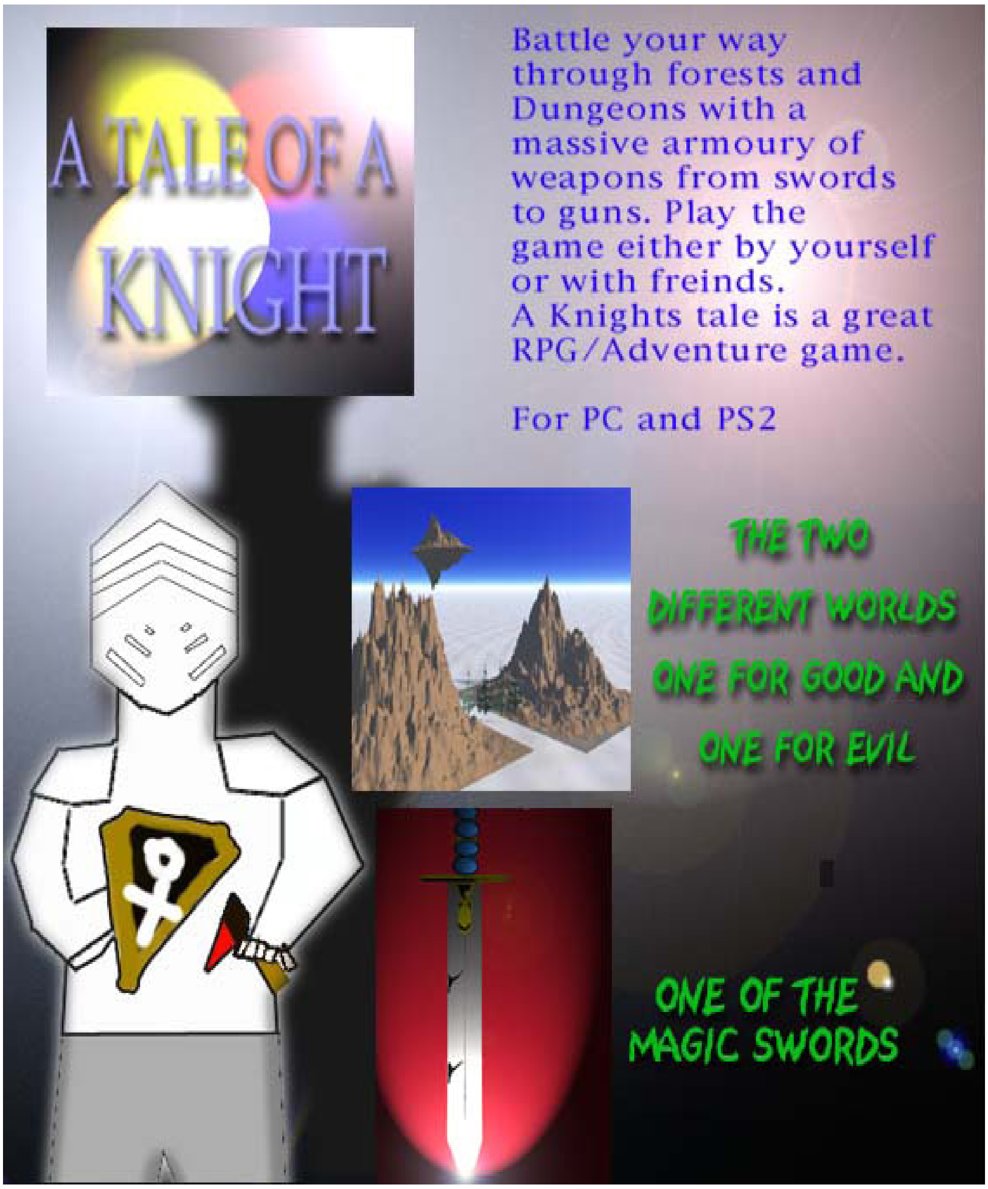

A TALE OF A KNIGHT
Battle your way through forests and Dungeons with a massive armoury of weapons from swords to guns. Play the game either by yourself or with freinds.
A Knights tale is a great RPG/Adventure game.
For PC and PS2
THE TWO DIFFERENT WORLDS ONE FOR GOOD AND ONE FOR EVIL
ONE OF THE MAGIC SWORDS

Abb. 1: POSTER MADE IN GAMES CLASS (November 2001).

During the designing of the posters the boys had to consider the concepts covered in the analysis. They chose specific genres of game and thought about what elements constitute a particular genre. For example, the poster in figure 1 advertises a medieval role play game by using particular elements (knight, sword, different worlds), language (1st person address, descriptive words) and lighting (shadow of the knight, glinting sword). The choices the boys made when making the posters reflect their awareness of elements of gaming and game advertising, thereby making explicit some of the otherwise implicit knowledge. Also, through the production of the poster and the process of making their knowledge explicit, the boys were engaged in further discussion and analysis of their gaming experiences.

The posters as well as the final games projects required use of professional production software (Photoshop for editing images, Flash for animating and making interactive elements, and two 3D software packages). The 3D software was used for making sophisticated images such as the „worlds" in the poster in figure one. This software allowed the boys to produce images which are similar to the environments in the games they play (i.e. 3D characters in many role play games as opposed to 2D ones in platform games such as Mario). Because professional software was being used, a large section of the course was spent teaching the software skills (e.g. constructing layers, using various filters in Photoshop, doing frame-by-frame animation in Flash). As will be discussed in the following sections, the final projects were hampered by the complicated nature of the software. The final projects were not games, but instead were animated fly-through introductions to computer games which the boys had designed. This aspect of the course raised questions about our original aim of looking at informal learning, because the software skills were actually being taught in a formal way. Furthermore, the lack of a game as a final outcome raised questions about the feasibility of engaging with young people's knowledge of gaming through production, as we had originally intended.

This brief summary of the course indicates the different types of teaching and learning which were occurring. The premise of our research was that young people were learning digital cultures outside of formal school settings, possibly incorporating new forms and styles of learning. Our question now is how to describe and analyse the learning and teaching we observed in the games class. The next section of this article will look closer at how software packages were taught and learned in the games class. Issues related to pedagogy and technology will be discussed, including how our study relates to the various discourses and models of learning outlined in the introduction.

3. Software in situ

3.1 Photoshop – Drawing and editing images

The first piece of software taught in the class was Photoshop, used for drawing and editing images. Formal step-by-step instruction was given to the entire group, and then they boys experimented with the software as the tutor gave one-on-one help to the younger, less experienced users. (Of the ten boys in the class, the oldest three had done a class on web design the previous year using Photoshop and Flash – professional software packages used for editing and animating images.). In Photoshop the boys could use their previous knowledge from any basic drawing software (Paint, Kid Pix, etc.), and the boys found it easy to use tools such as filters to alter to images. They could use simple drawing skills to get started fairly quickly, and the filters gave the boys ways to make their hand-drawn images look more sophisticated. The boys used the pull down menu in Photoshop to experiment with different effects. For example, Lawrence (age 12) gave his hand-drawn sword dramatic lighting effects (the background, the glinting tip, and the sunspots) using simple filters (see figure 2).

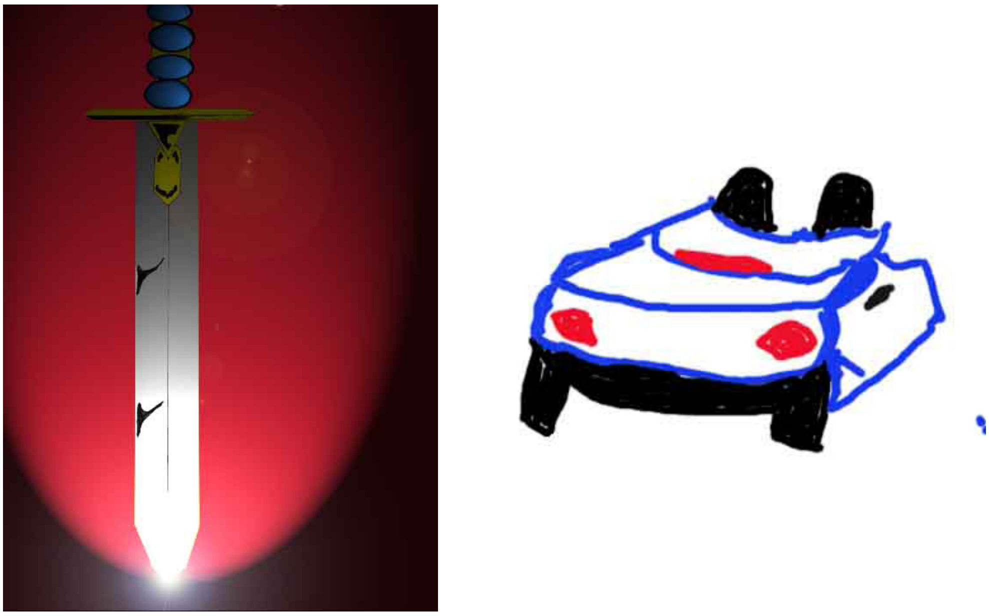


Abb. 2: LAWRENCE“S SWORD AND LUKE“S CAR (October 2001).

The younger boys (Jordan and Luke, age 9) struggled even with the basic drawing tools, as is evident in Luke's drawing of a car (figure 2, above). Luke found the concept of layers difficult, and he only used the pencil tool on the software, changing the colour but nothing else. Jordan similarly used simple pencil tools to draw a knight. When Luke and Jordan were taught to use other tools, they were not able to „take on board“ what they were being taught, indicating that perhaps the teaching and software were too advanced for them. For example, Jordan wanted his knight's armour to look shiny, so he was shown how to change the lighting and opacity. Jordan then forgot to save the changes that he had made, and when he tried to repeat what he had been taught (the same day) he was unable to proceed. Of course there were other circumstances which could have been affecting the boys' learning (for example, lack of practice time during the week and erratic attendance), and therefore it is difficult to say if the teaching or software were generally too advanced. If Jordan were shown again how to change the lighting, and if he practised it several times over the course of a week, then he probably would have had more success. However, there is a question about how Jordan would learn to conceptualise the armour as shiny and therefore know what tools to use. When Jordan said he wanted the armour to be shiny, the tutor explained that he could „give the appearance of shininess“ through lighting and opacity. It is unclear in our study how this element of production, being able to imagine and then conceptualise a particular image or effect, is learned.

2.3 3-D software

On the whole, the boys were content with Photoshop when they were drawing objects and writing text. However, when the boys started drawing their characters and landscapes, they were dissatisfied with the simple drawing images they were producing. They wanted their images to appear more realistic and less hand-drawn. Therefore, two 3D packages (Bryce 3D and Poser) were introduced to supplement their work. The boys quickly dismissed their work in Photoshop in favour of the 3D look. Louis, for example, had been fairly successful at drawing a man and a mountain (using Photoshop). But when he repeated those images using the 3D software, not surprisingly, he did not return to his Photoshop work (see figure 3).

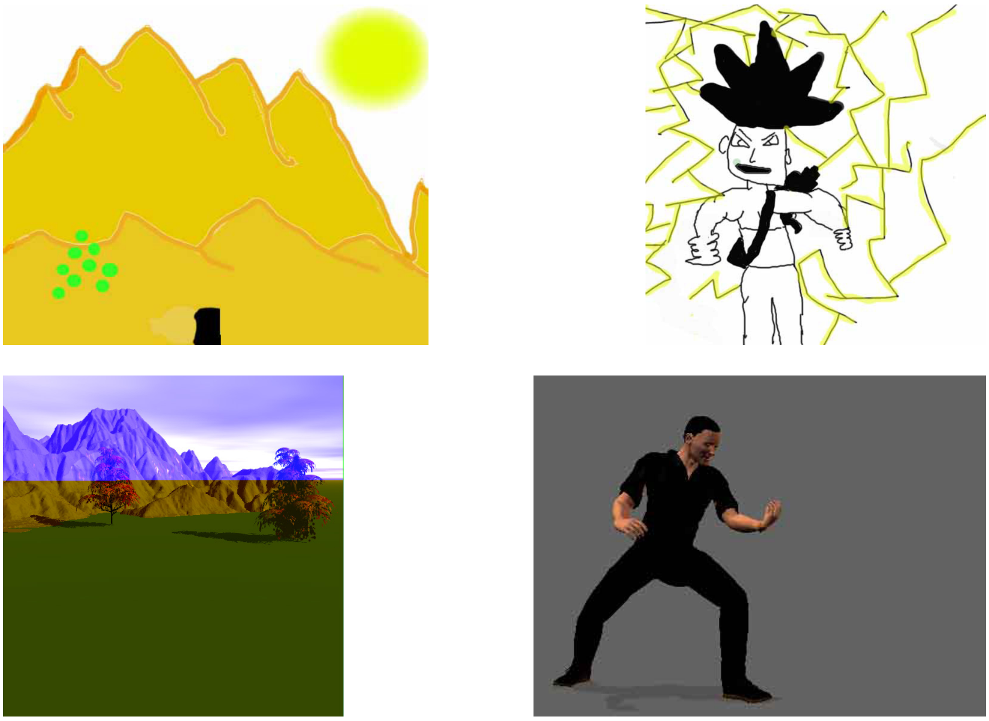


Abb. 3: LOUIS“ WORK IN PHOTOSHOP AND BRYCE 3D (Oct. and Nov. 2001).

The 3D software has several advantages, in addition to looking quite sophisticated. For most of the boys the software required almost no instruction, so they were able to explore the software, achieving instant satisfaction with very little effort. At this point in the course the boys had attended every Saturday morning for two months, and they expected to make a game by the end of the year. Although the boys were willing to give up their Saturday mornings to learn how to make a computer game, there was a balance between how much time they were willing to invest without some sort of noticeable progress. The 3D software offered what seemed to them to be major progress towards a game, and they could use the software independently to create images. However, because the images looked so sophisticated, the boys were too intimidated to use other programs for drawing. The most significant problem with using the 3D software was that it was hard to import and manipulate in other programs (Photoshop and Flash). The game production which the tutor had imagined involved creating images and then assembling, animating and making elements of the images interactive. Combining images from different programs complicates this production process. Even the posters (assembled in Photoshop) included many imports from various sources which needed to be formatted and arranged, and this assembling required one-on-one instruction. This instruction often did not result in them mastering the concept or procedure (for example, the boys were not able to do the formatting independently after the one-on-one help). Referring to the models of learning discussed at the beginning of this article, one could describe the instruction as faulty because it was not occurring within the boys' ‘zones of proximal development’. This description of the interactions reflects a linear model of learning whereby the boys need to learn certain skills and concepts in relation to digital graphic design before they can proceed.

However, although skills may not have been learned, the tutor was introducing the boys to the world of graphic work on professional software, much in the way Lave and Wenger (1991) discuss situated learning through „peripheral participation“. The tutor would regularly give general advice such as „try to leave as many windows open as you can“, „try to label each layer with a name that describes what's on it“. She also made general conceptual statements; for example „the machine allocates memory to every single application, so it will run much quicker if you close applications you're not using“. She used technical terms such as bitmaps, jpegs, tweening and megabites and discussed issues such as layering, different types of files and relevance of file sizes. There is an enormous body of skills, knowledge, concepts and discourse that needs to be learned here. As the tutor used the discourse the boys gradually developed an understanding of the field (especially the older boys). For example, when the tutor was showing the boys how to do the formatting and importing, she was using technical language which perhaps made more sense as they saw her do the formatting in other contexts throughout the year. I will return to a discussion of these models of learning in the final section of this article.

3.3 Flash animation software

After the boys made their posters, using the 3D software and Photoshop, four class sessions were devoted to formal instruction on Flash, a professional animation software package. In these sessions the boys did experimental projects which included scripting interactive elements (using buttons, for example), but in the end these skills were not used as part of their final games projects. The boys also learned „tweening“ which allows for objects to be animated without programming every frame, and they learned frame-by-frame animation. The final projects ended up being animated narrative introductions to the boys' invented computer games. The animation projects were assembled in Flash, and the tutor did most of the work of assembling the projects, due to the complexity of the task. Only one boy, Jake, used Flash independently in his final project (to make images of a newspaper spinning and Big Ben collapsing, see figure 4). Towards the end of the course, Lawrence, who was also able to use Flash independently, was making very rudimentary games in the first half hour of the class before the sessions started (for example, in one of Lawrence's games the player had to try to click on a moving target, see figure 5).

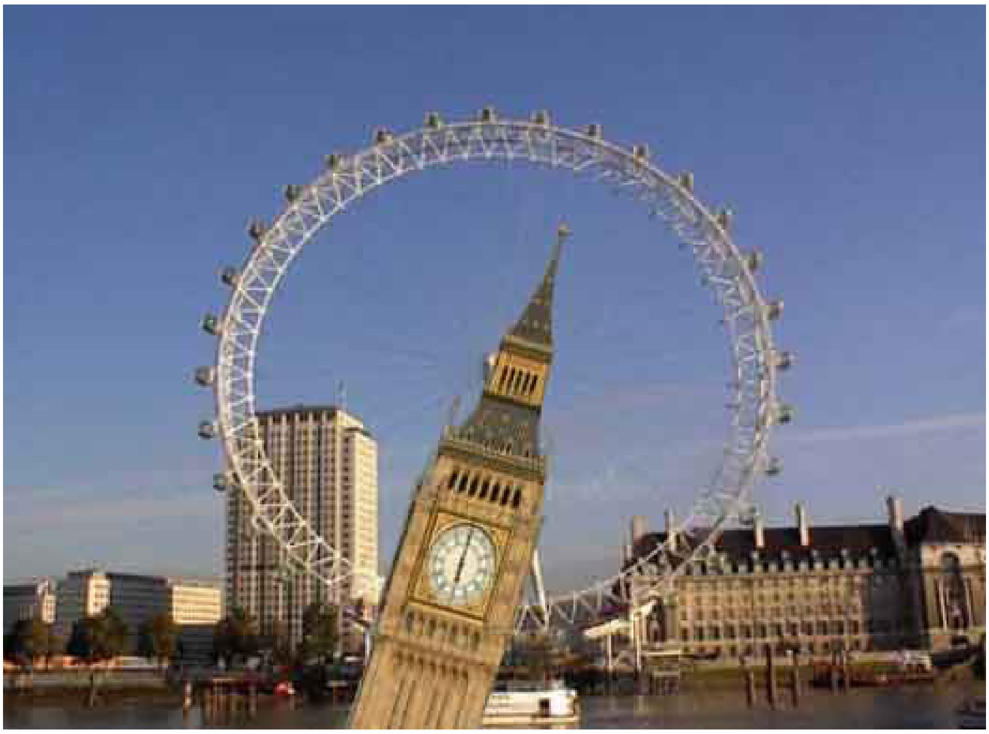
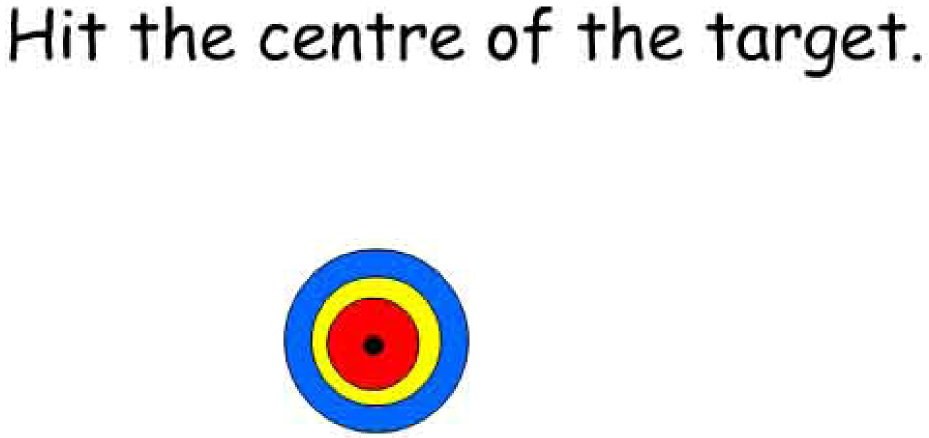


Abb. 4: SCREENSHOT OF BIG BEN COLLAPSING, JAKE, MAY 2002.



Hit the centre of the target.

Abb. 5: SCREENSHOT OF SIMPLE GAME, LAWRENCE, MAY 2002.

4. Graphic problems – issues raised by the use of professional software

One of the major issues highlighted by the course concerns the choice of software, especially with this age range. There are other packages available such as Kid Pix, HyperStudio or Stagecast which are produced for children. Using these packages would have resulted in different types of final projects. Stagecast, for example, is a game production software for children, but it can only produce platform games. The choice of software used in the games class was intended to leave open possibilities for the boys to produce the type of game they wanted, based on their knowledge and preferences as gamers. However, in practice the data from this project suggests that the professional software actually limited the production of games because the software was so advanced, particularly when the packages were combined (e.g. 3D images from Poser were animated in Flash). Because the 3D software was easy to use and produced images that are much more like the high-quality graphics in games that many kids like to play, it looked as if the software started leading the designs of the final products. Paradoxically then, although our aim was for the boys to design a game (without being restricted by a determined design built into specific software packages), the end result was still that the software led to a particular type of game.

Both the tutor and the boys who were interviewed at the end of the course recognised that it would have been better to stick with Flash (which is two-dimensional) and drop the 3D software. Jake said, „Flash makes it simpler; with other software you have to keep putting it into different files, whereas in Flash you can just animate it and make it and just run it“. In completely independent interviews, Jake and Lawrence said that Flash was their favourite software because „it's quite easy to make cartoons, it's a really good drawing tool“ and „it's easy to muck about with it“. Jake said, „I think you need to be taught the basic stuff but then you can learn the rest“. This is an important learning outcome for the boys, and certainly the tutor tried to persuade the boys to use more Flash, but they seemed to need to get the 3D images out of their system and learn through experience. Furthermore, going through the difficult process of using 3D images led to an understanding of why games that they play cost so much and take so much time to produce. This is another important learning outcome when considering how to develop young people's understanding of digital media. It is also significant that the other boys (besides Jake) who had experience with Flash from the previous year did not use it in their projects, apart from when the tutor helped them. The skills they learned seemed difficult to apply to the task set (designing the animated introduction), and the software did not help to access their knowledge about high graphics games (which they were designing). The key finding here is that there was a mismatch between the task and the software available – an issue I will return to when I discuss pedagogy.

As I explained earlier, the tutor chose to use professional software as opposed to more child-orientated software such as Stagecast or HyperStudio. The choice was partly due to the framework guiding our research, but primarily it was the tutor's choice. The tutor is an artist, games player and professional software user, and as such she had personal preferences about the style of software she wanted to use. For example, she said she found an alternative software package „clunky“. However, several factors prevented the boys from reaching the point of being independent (they were not using the software apart from a couple hours a week, many were young and inexperienced with different software, they had little experiencing conceptualising graphics, and the tutor may have been using an approach that was not effective).

The course introduced the boys to the software, but (unlike software designed specifically for young people) the software is not scaffolded enough to allow young users to explore and learn independently. The software does not have a beginning level which introduces concepts and allows the user to gradually learn more technical aspects. Looking at learning of Logo-based software, Kafai and Resnick (1996) argue that learning depends on both the structure of the software and the developmental stage of the user. They describe the structures which scaffold learning on software as „training wheels“. According to Kafai and Resnik, those structures are based on observations of what experts do, however, novices are unable make use of them until they are at the appropriate stage in their learning process (i.e. when they realise that things need to be organised in a particular way). One of the questions our study raises is whether there should be a range of software which suits different developmental levels or whether software preference is more about the users“ learning style and mode of thinking. If we accept that software should have different developmental levels, this leads to questions about what cognitive skills are involved in using production software. Could we call the use of simple paint tools part of the first stage of development for young people using graphics programs, and if so what skills and concepts are being developed at this level? Is there a set of visual literacies that needs to be learned in order to use graphics programs? For example, in the image of the sword (Figure 2), how did Lawrence learn to conceptualise what he was imagining?

Another software related problem which held up the production of the games was the place of the software in group work. The tutor had decided to group the boys so that the boys who had previous experience with the software or caught on quicker were matched with the younger, less experienced boys. In theory, the tutor thought that a peer-tutoring situation would develop during game production. In practice, however, the times the boys were working in pairs usually involved working on their own computers, and then importing their individual work onto a joint document. The boys often had questions when they were working individually, but even more knowledgeable peers were not able to answer those questions. Furthermore, younger and less experienced Information and Communications Technology (ICT) users had a very hard time joining in group work with more advanced users. At this point, even the more experienced boys did not understand the software sufficiently to be able to help their peers. Furthermore, the boys were not familiar enough with the software to know how to divide the work so that less experienced users could do simpler tasks. For this type of interaction, the software was too advanced. Therefore, our study raises questions about the types of software used in group work and also about the role of group work in ICT production, which is a common practice in schools. For example, should we only set up peer tutoring situations if the tutor has gained a sufficient amount of conceptual understanding of a software program? If so how do we know what level of understanding is needed and how do we measure that understanding?

5. Technology and pedagogy

As described in the introduction, there is a particular discourse that constructs young people's relation to new technology as unproblematic (the „natural cyberkid“). This discourse claims that through access to digital technology (with or without instruction) young people will learn to use powerful software that will allow them to do many creative things (as well as develop various useful skills). The discourse implicitly assumes that pedagogy is not important, and therefore the role of teachers in learning new technology is not discussed. However, the difficulties encountered on the course outlined in previous sections point to a need to consider pedagogy. If software requires formal instruction, not just trial and error, in order for it to become a creative tool for young people, then pedagogical issues arise. If there is a series of developmental stages in relation to learning technology, then pedagogy needs to be considered. And finally our goal of accessing young people's knowledge as consumers of games through particular pedagogical means needs to be examined.

5.1 Models of learning in the games making class

As I have described, the tutor had a basic plan which involved a series of sequential activities and constant application of theory and knowledge through practice. The tutor engaged the kids in critical analysis of computer games they play. She taught the software step-by-step to the whole group and then gave individual help. She used her own experience of gaming and using software to model and give advice, and she tried to make connections between practice and theory by referring to games when the boys were learning software. These are all characteristics of what one might describe as good teaching. However, the boys did not learn as much as the tutor thought they would, and so she modified her goals. The important question is why the boys did not learn as much as the tutor had planned. Some of the problems relate to the characteristics of informal educational settings. Unlike a school, the course had a looser structure in terms of attendance – a couple of boys joined the course half way through and one dropped out, two boys left early every week for other lessons, they were often late for class, and they sometimes had other commitments on Saturdays so their attendance was erratic. These factors made it hard for the tutor to plan lessons and teach them in a sequential, orderly way. However, the tutor thought the difficulties in learning were also due to the boys: they weren't working hard enough, they weren't committed to the projects and they didn't spend time practising using the software. In her opinion the boys would have learned if they had shown more interest – an opinion which strikes a chord with the „natural cyberkid“ discourse mentioned earlier. The constructionist model discussed at the beginning of this article would suggest that if the boys had more time with the software they would have developed their skills naturally. Using the constructionist model, it wasn't the lack of sequential lesson plans which hindered the learning, it was the lack of opportunity to explore the software. Certainly some of our data suggests that ICT skills are learned through repetition and over self-directed time which allows for learning through trial and error and exploration. Using a Vygotskian model of learning, however, the problem with the tutor's instruction was that she didn't find out where the boys were, in terms of software skills, and so she didn't build on their knowledge and engage with them within their zone of proximal development. The answer to the question of why the boys found it difficult to learn the software skills is not clear, and requires further theoretical reflection on the models of learning ICT we have taken for granted.

5.2 *Scaffolding*

In our study we collected data which can be explained through several different models of learning. Some of the instances of learning which almost „jumped“ out at us as significant moments happened at opportune moments when the learners where just at the level where they could take on new knowledge, and when the instruction was building on what they already knew. So, for example, when the boys tried to do something but it didn't work, then the value of the tutor's instruction was maximised. At these times there was a context for the instruction both in terms of the purpose and the tools (the boys had a goal and had already tried some tools to achieve their goal). This data suggests that „scaffolding“ by a teacher or more knowledgeable tutor/peer is crucial to the learning process (Bruner 1987). Bruner, whose work is based on Vygotsky's theories, used the term ‘scaffolding’ to describe the interactions between a learner and a teacher or more able peer whereby structures are put in place to support the learner in mastering a task. Effective scaffolding occurs within the learner's zone of proximal development and is gradually withdrawn as an action becomes internalised. A model of learning based on scaffolding perhaps indicates that a lot of time and energy (and ultimately enthusiasm) is wasted if instruction doesn't happen at the right time. In the games class the tutor spent significant amounts of class time giving one-on-one help, but in doing so she created a situation in which the boys were often waiting before they could receive help. During that waiting time they were repeatedly trying to figure out how to do something, to the point of frustration. This strategy was thus counter-productive.

Clearly, a teacher can not always be present at exactly the moment when a learner needs the next bit of scaffolding. However, the scaffolding which the teacher provides can also be „faulty“. In our data I could describe some situations as times when the tutor's instruction seemed too advanced (i.e. not in the boys' zone of proximal development, using Vygotsky's term). For example, the tutor would give advice about the size of files or about naming layers, but much of the advice went over the boys“ heads, given that they weren't even sure how to save files in correct places. However, using Lave and Wenger's situated learning model (1991) mentioned earlier, we could describe these occasions as times when the boys were learning the culture of digital production use by a master. Therefore, we have to ask if it matters whether or not the boys were „learning“ everything the tutor was trying to teach, or whether the boys should be expected to be achieving something all the time at all different levels. Pedagogic models found in many schools (at least in the U.S. and U.K.) based on Piaget, Bruner and Vygotsky (1971, 1987, 1962 respectively), are more linear than a situated learning model, so instruction has to happen in a particular way, time and place. Certainly educators believe that children can stay on one step for a while or progress backwards, but given the right environment all children are able to progress. This model of learning perhaps does not apply to the learning in all aspects of digital cultures.

5.3 Alternative models of learning

The games class was seen by the arts centre as a pilot project, and the class was repeated the following year with several revisions, raising further questions about the kinds of learning I have described. A significant fact is that almost all the boys returned for a second year, and several new children joined, including two girls. This simple fact perhaps indicates that my interpretation of the level of frustration which the boys were feeling was overstated. We may have been looking for „completed learning“ at too early a stage. Perhaps it was only me who was frustrated by the non-linear and non-sequential approach. If this is the case then we must look for other models and styles of learning to explain our findings.

Lave and Wenger (1991) argue that there needs to be a shift away from the concept of an individual learner and that notions of mastery and pedagogy must be decentred. They write, „(R)ather than learning by replicating the performance of others or by acquiring knowledge transmitted in instruction, we suggest that learning occurs through centripetal participation in the learning curriculum of the ambient community“ (p. 100). Therefore, instead of looking at the individual skills that each boy developed (or failed to develop) in the games class, we could look at their learning as a process of interacting in a (pseudo-) games making environment.

Researchers working with Logo and similar programming packages for children (Hoyles et al. 2001; Kafai/Resnick 1996; Papert 1993) also see learning as a process which is not as linear and sequential as strict developmental models would describe. On a recent research project called Playgrounds (www.ioe.ac.uk/playground), which looks at children building computer games using specially designed software, researchers describe how the learning develops as the children experiment with the software. Instead of having a tutor impart knowledge in an organised way, children learned to programme through their exploration of the software. This is a recursive process, as Goldstein and Pratt write, „As learners become familiar with the tools, they become aware of new opportunities and utilities of those tools. Through using the tools, the learners re-construct their understanding of them. This shapes the way that the learners think about their solution to the problem and the problem itself“ (2001, p. 2). Looking at this in relation to the games class, it is possible that the boys did not require the tutor to give them step-by-step instructions, but instead they were gaining familiarity and learning to use the software through a gradual process of experimentation.

Another non-sequential approach to learning is taken by researchers looking at computer game playing. In our research and, I would propose, in the experience of anyone watching a child learning to play a computer game, there are few times when children will sit down and be given step-by-step instructions by a tutor or instruction booklet. Children start playing a new game with little instruction, and they learn as they play. Toni Downes (1999) argues that playing games is producing new styles and ways of learning.

Downes writes, „Within game playing the continued success of using the „learning by doing“ and trial and error approaches alter children's predisposition to learning and performing in similar environments, particularly other computing environments such as word processing or using information data bases. Importantly these computing environments, through their interactivity readily afford these approaches and therefore reinforce this pre-disposition towards exploratory modes of learning“ (p. 77). Looking at this description of learning, we could say that the boys on the games class did not need the sequential instruction of the tutor, especially as they were all avid game players who were accustomed to learning through trial and error. We were asking the boys to apply their skills and knowledge of playing as they engaged in digital production, but in expecting a linear model of learning we overlooking an important gaming skill – learning by doing.

A brief conversation about hand-outs which I had with the tutor who taught the class in the second year exemplified the non-linear approach to learning which is used by the young people in the class. For example, the tutor gave the children hand-outs when instructing on Flash, listing all the steps involved to achieve a particular effect (e.g. shape tweening or masking). The tutor described how the children didn't even glance at the hand-out after she had shown them the steps on the computer. They felt comfortable relying on their memory to go through the steps, and they were not anxious about remembering the exact sequence of steps. According to the tutor, the children were more concerned with the overall effect that they were trying to achieve, and they were comfortable exploring the software in order to achieve the effect, rather than following the specific steps. This approach taken by the young people contrasts greatly with the one taken by adults whom the tutor has worked with on the same concept. Adults are anxious about missing out steps, and they glue themselves to the hand-outs, taking additional notes during the instruction. These two contrasting approaches to learning are explained by the theories I have outlined, and help explain why I, as an adult, am less comfortable with non-linear learning and perhaps misinterpret learning situations, particularly in relation to new technologies.

One obvious question raised by the games course both during the pilot and the following year is whether it is possible to produce games with young people. Although the children were more successful the following year, for various reasons including the fact that Flash was the only software used, the projects during the pilot and the follow-up year can not be classified as games. By the same token, the fly-through introductions from the pilot year are actually animated narratives, not games. The second year the children made interactive animations, but they were not games either. Our aim was to engage with kids' knowledge of the games they play at home through the production of games, but we ended up focusing on learning of software and discussing graphics. A course using alternative software (Logo or Stagecast) would focus on learning the logics of programming, and similar to our games class would miss out on engaging with kids' gaming experiences.

6. Conclusion

The study, perhaps, raises questions rather than providing answers. It is clear, however, that children and young people are experiencing various ways of learning through their consumption and production of digital cultures. Computer games, for example, can involve endless repetition, trial and error and risk taking in their consumption, as well as in their production, as this article has described. However, gaming can also involve careful scaffolding. When playing a computer game, the first level is easier than the other levels and sometimes includes auditory or visual hints on how to progress. There is an economic advantage to scaffolding the learning in this way so that players will continue to play the game. Similarly, as described in this article, digital production requires some degree of scaffolding in order for users to make progress and avoid frustration.

The problems encountered in the computer games making class as described in this article raise questions about the relationship between knowledge and production. For example, what kinds of knowledge are needed in order to engage in production, and how is that knowledge developed? Is it possible for young people to produce the games they themselves play, and if so would that production process engage in the critical analysis we are hoping for? As explained earlier, our aim was to use production as a way of accessing the boys' knowledge of games which they acquired through game playing, to make that knowledge visible and somehow to involve a critical framing of that knowledge. Instead, what ended up happening was that the production tools disempowered the boys, making their knowledge of games fairly useless. Although the tutor continually reminded the boys that the high graphics videogames they play (Tomb Raider, Grand Theft Auto) involve many years of development and expansive budgets, the boys still had quite high expectations about what they could produce. This is an important contribution to the debate about the role of production in media studies. We need to find the tools which will allow us to empower students; tools which will make visible the embedded knowledge of their media culture. Furthermore, we also need to consider when to use a model of learning based on a developmental progression of skills related to production, or when to see young people as learning technology through immersion into the digital culture.

References

Bruner, J.: Actual minds, possible worlds. Cambridge 1987.

Buckingham, D./ Grahame, J./ Sefton-Green, J.: Making Media: Practical Production in Media Education. London 1995.

Coffield, F. (ed.): The necessity of informal learning. Bristol 2000.

Downes, T.: Playing with computing technologies in the home. Education and Information Technologies 4. 1999, S.65–79.

Facer, K./ Furlong, J./ Furlong, R./ Sutherland, R.: Constructing the child computer user: from public policy to private practices. British Journal of Sociology of Education. 22:1 2001, S. 91-108.

Goldstein, R. & Pratt, D.: „Michael's Computer Game: A Case of Open Modelling" in M. van der Heuval-Panhuizen (Ed), Proceedings of the Twenty Fifth Annual Conference of the International Group for the Psychology of Mathematics, Vol 3, Utrecht 2001, p. 49-56 (www.ioe.ac.uk/playground/RESEARCH/papers/open_modelling.pdf).

Hoyles, C./ Noss, R./ Adamson, R./ Lowe, S.: „Programming rules: what do children understand?" Proceedings of the Twenty Fifth Annual Conference of the International Group for the Psychology of Mathematics. Utrecht 2001. *[*www.ioe.ac.uk/playground/RESEARCH/papers/programming_rules.pdf].

Kafai, Y./ Resnick, M. (eds.): Constructionism in Practice: Designing, thinking and learning in a digital world. Mahwah 1996.

Lave, J. and Wenger, E.: Situated Learning: Legitimate peripheral participation. Cambridge 1991.

Papert, S.: The Children"s Machine: Rethinking School in the Age of the Computer. New York 1993.

Piaget, J.: Science of Education and the Psychology of the Child. Harlow 1971.

Vygotsky, L.S.: Thought and Language. Cambridge 1962.