4 Designing digital technologies and learning activities for different geometries

Keith Jones[1], Kate Mackrell[2], Ian Stevenson[3]

[1]University of Southampton, UK;

[2]Queen's University, Canada;

[3]King's College London, UK.

Abstract This chapter focuses on digital technologies and geometry education, a combination of topics that provides a suitable avenue for analysing closely the issues and challenges involved in designing and utilizing digital technologies for learning mathematics. In revealing these issues and challenges, the chapter examines the design of digital technologies and related forms of learning activities for a range of geometries, including Euclidean and co-ordinate geometries in two and three dimensions, and non-Euclidean geometries such as spherical, hyperbolic and fractal geometry. This analysis reveals the decisions that designers take when designing for different geometries on the flat computer screen. Such decisions are not only about the geometry but also about the learner in terms of supporting their perceptions of what are the key features of geometry.

Key words design, digital technologies, ICT, learning, geometry, geometries

4.1 Geometry, technology, and teaching and learning

While forms of algebra software (such as *Derive*, *Macsyma*, *Maple*, *Mathematica*, etc) were amongst the first mathematics software packages (pre-dating, in many cases, the graphical interface), it is software tools for geometry (beginning with *Logo* and followed by 'dynamic geometry' environments such as *Cabri* and *Sketchpad*) that have emerged as some of the most widely-used digital technologies in the mathematics classroom - and arguably amongst the best researched (for reviews, see Clements et al. 2008; Hollebrands et al. 2008; Laborde et al. 2006).

Our aim in this chapter is to consider the interplay between the design of digital technologies and activities that utilize those technologies for learning mathematics. We focus on geometry, covering both Euclidean and co-ordinate geometry in two and three dimensions, and non-Euclidean geometries such as spherical, hyperbolic and fractal geometry. The reason for considering this span of geometries is to capture key aspects of how the use of digital technologies can and does shape the mathematical activity of the user.

The chapter concludes by reflecting on how some of the key decisions that need to be taken regarding issues of geometry are handled by designers of digital technologies, and by designers of related learning activities, and on the implications for future users of educational digital technologies.

4.2 Working with different geometries on the flat screen

A distinctive, but perhaps somewhat neglected, characteristic of current digital technologies is 'flatness', both of the screen used as the visual medium in the classroom, and the 'computer mouse' operating on a flat mouse mat. As we demonstrate in this chapter, 'flatness' is problematic when representing and interacting with any geometry, and even introduces design issues when working with plane (two-dimensional) geometry.

That the flat screen presents some difficulties in handling representations of different geometries is nothing new. Artists and mapmakers have wrestled for centuries with trying to present the three-dimensional (3D) world on the two-dimensional (2D) canvas or atlas. In western art, beginning in the 15th Century with artists such as Brunelleschi, the use of perspective first found systematic presentation in Alberti's *Della Pittura* published in 1435. The most common method for representing 3D space on a surface, usually known as linear perspective, is illustrated by Albrecht Dürer in a famous engraving of 1525 reproduced in Figure 1. Here a hook on the wall takes the position of the eyes, and a taut string represents the straight line joining the eyes to a visible spot beyond the frame. This provides one solution to the problem of representing solid (3D) objects on a flat surface in a way that is compatible with human stereographic vision. As such, the idea of linear perspective is a result of taking account of human perceptual apparatus.

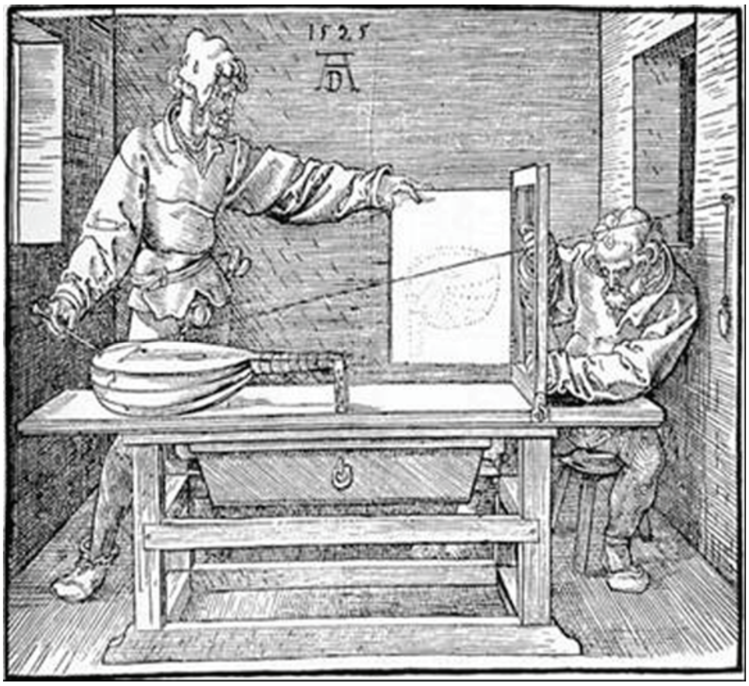

Fig. 1. one-point perspective, as illustrated by Albrecht Dürer in 1525.

In cartography, many forms of map projection have been developed as attempts to portray the surface (or a portion of the surface) of the earth (taken as a sphere) on a flat surface. Each of these projections maintains some geometrical properties (such as distance, area, or shape), but, by their very nature, such projections cannot maintain all such properties simultaneously. What is preserved, geometrically, in any particular cartographic projection, and what is not, is dependent on the purpose for which the 2D map is created (Kreyzig 1991).

A major revolution in geometry came in the 19th century with developments that led to consistent non-Euclidean geometries, and the emergence of curvature as a key idea. Work by Euler, Wolfgang and Janos Bolyai, and Lobachevskii, to name but a few, showed that Euclidean geometry was one of many possible geometries: its uniqueness lay with its 'flatness', not, as Kant would have it, because it is 'absolute'.

Curvature, as a geometric property, and because it characterizes more geometries than Euclidean, became an active area of research. Gauss and Riemann, for example, showed that curvature is an intrinsic property of surfaces, defined locally rather than globally. Hitherto, curvature had been defined by embedding a non-Euclidean surface in Euclidean (two and three-dimensional) space and using the associated global co-ordinate system. Riemann's introduction of a local description of geometry removed the need for projections, a technique which, as noted above, arose out of human (perceptual) need rather than mathematical necessity.

Yet the price of this advance into a range of geometries can be the loss of visual intuition that we need, as humans, to understand our experience of space.

In architecture and in many branches of engineering, prior to the development of computer-assisted design and manufacture (CAD/CAM), the ‘distorting’ nature of the forms of projective geometry used in cartography was circumvented through the use of orthographic projection (as developed by Monge in the late 18th century; see Bessot 1996) in which several 2D views of the object (often referred to as front, side, and plan elevation) are utilized instead of a single view. With the development of CAD/CAM, 3D modeling became possible – first through a 2D-to-3D paradigm (whereby the 3D object is built up from 2D objects) and more recently through the use of new geometric forms (including grid-like polygonal subdivisions of surfaces known as ‘meshes’ and curves in 3D space defined by control points known as ‘splines’), assisted, at times, by the use of a 3D input device (rather than the usual mouse on the 2D plane).

What this short historical introduction indicates is that projections of various kinds are the result of human needs, sometimes dependent on the available technological medium – such as the ‘flat screen’ of the canvas or atlas – and sometimes because of human stereographic vision. As such, projections need to be understood in relation to the problem that led to their creation.

Introducing digital technologies has enabled us to interact with more forms of geometrical objects, and this underlines the need to understand the conventions of the flat screen and how that medium alters our appreciation of the translated logical geometric structures (Euclidean or otherwise). What digital technologies may offer is a way of building, and developing, our visual intuition across a range of geometries. Yet we need to be much clearer as to the affordances and constraints of such technologies in the teaching/learning process. It is these issues that we turn to next.

4.3 Designing digital technologies for different geometries

In examining decisions about representations and interactions when designing for different geometries for the flat screen, we focus on three geometry technologies that are common to mathematics classrooms: 2D ‘dynamic geometry’ environments (such as *Cabri* and *Sketchpad*), software for 3D geometry (with 3D Euclidean geometry illustrated by *Cabri 3D*, and 3D coordinate geometry software illustrated by *Autograph*), and software suitable for various non-Euclidean geometries (illustrated by the use of *Logo*).

4.3.1 2D dynamic geometry environments

Over the years since the first ICMI study on technology (Howson and Kahane 1986) when users had to rely solely on text-based input via the keyboard, major innovations have involved the introduction of direct manipulation graphical capabilities that have become synonymous with contemporary computers (Norman and Draper 1986). Such changes have impacted particularly on geometry education with the development of 'dynamic geometry' environments (DGEs) such as *Cabri* and *Sketchpad* (and many others).

At first glance, a DGE is nothing more than a graphics editor enabling geometrical figures to be drawn on the computer screen. Yet there is more to it than this because with a DGE the user can utilize the mouse to 'grasp' an element of the on-screen figure and drag it about. As this 'dragging' takes place, the diagram on the screen changes in such a way that the geometrical relations specified (or implied) in its construction are maintained. Such digital environments are called 'dynamic' for this reason.

Yet the way in which a DGE figure moves when it is dragged is not solely to do with geometry. Even though, as Goldenberg and Cuoco (1998) explain, an over-riding principle in DGE interface design has been to try to ensure that the behavior of geometrical objects constructed on-screen conform as closely as possible to how users would naively expect them to behave (in Euclidean 2D geometry), there is an unavoidable tension for DGE designers between the need for objects to move continuously when dragged, and the need for the position of the constructed elements to be uniquely determined (Gawlick 2004). The problem for DGE designers is that no DGE can be fully continuous and fully deterministic at the same time. For deterministic DGEs (and most currently available DGEs are deterministic) while on-screen figures are completely determined by the given points, the result is that some constructions can jump or behave unexpectedly when a particular point is dragged. With continuous DGEs (the minority at the moment), dragging any point does produce a continuous motion of the construction (through the use, usually, of a heuristic 'near-to' approach) but it can happen what when a dragged point is moved back to the original position, the resulting construction might be different from the original. Gawlick (2004) provides illustrations of both cases.

The result of such issues is that users of DGEs need to learn to distinguish between changes in the on-screen image (as objects are dragged) that are a consequence of geometry and those that are the result of decisions of the software designer. A seemingly trivial example is that, in some DGEs, objects that look the same may not act the same (for instance, some points may be dragged while others cannot). Yet even this apparently trivial issue can leave beginning DGE users wondering why not (Jones 1999). Another design decision involves deciding whether an arbitrary point on a line segment might maintain the ratio to the endpoints when either is dragged – or whether, for instance, the point jumps to another arbitrary position (since it is an arbitrary point), or whether it maintains a fixed distance to one or other of the endpoints. The common decision by DGE designers seems to be to maintain the ratio to the endpoints when either is dragged. Yet this is a decision of the DGE designers; it is not something governed completely by geometric theory. For more on the decisions of DGE designers, see Goldenberg et al. (2008); Laborde and Laborde (2008); Scher (2000).

In graphing software such as *Autograph* (which shares some aspects of a DGE), every object is defined relative to a coordinate system. This means that changing the relative scale of the axes changes the appearance of objects. The consequence is that, for example, lines which have been defined as perpendicular will no longer 'look' perpendicular when one axis scale is changed (though, of course, in the mathematical sense, the lines remain perpendicular). In contrast, in some DGEs (such as *Cabri* or *Sketchpad*), objects are not necessarily defined in relationship to coordinate axes. In such DGEs, a circle (defined, in effect, as the locus of points that are a fixed distance from a fixed point) retains the appearance of a circle on-screen even when either coordinate axis is changed. The impact of such decisions regarding the role of coordinate systems in the representation of objects on learners (especially beginners) is currently under-researched.

Whatever the DGE, another design decision relates to the provision of menu items (Goldenberg et al. 2008). Providing too few means that more things need to be constructed, something which becomes very tedious. Yet providing too many menu items produces undue complexity (rather than 'user-friendliness') and could mean that teaching opportunities are lost. Finding the balance between these two aspects is a key design decision in any educational application – and is something that simultaneously involves technical and pedagogical issues (Hoyles et al. 2002). In tackling the issue of too many, or too few, menu item, many DGEs, while necessarily prescribing a selection of provided constructions, also allow some menu items to be 'hidden' (thus allowing the software interface to be simplified) while, at the same time, featuring a macro or 'script' facility for user-defined constructions to be automated (thus allowing new idiosyncratic menu items to be added). How this adjusting of menus is used by teachers, and the impact on learners, is currently under-researched.

What this section illustrates is that there is a range of issues that add complexity for the technology user when it might be assumed that plane geometry on a flat computer screen would be the most straightforward case of doing geometry with digital technology.

4.3.2 Software for 3D geometry

From a purely mathematical perspective, it is perfectly possible to use common 2D geometry software to create '3D' objects, figures, and graphs. Yet it is complicated and time-consuming to do so. As a consequence, recent software development has provided a range of geometry environments in which learners can manipulate 3D objects directly on-screen. Such environments include *Cabri 3D* and *Autograph* (version 3).

The issue of representing 3D objects on a flat screen means that a number of design decisions, unique to 3D software, need to be made by software developers. One key decision is how the opening software screen both orients the user to 3D space, *and* provides a framework for the creation of 3D figures and structures. This has been tackled in different ways by different software developers. The opening screen for *Cabri 3D*, for example, shows part of a plane, with, at its center, three unit vectors representing the x, y and z directions (see Figure 2). This initial viewing angle was chosen so that the plane and vectors would have an appearance compatible with the usual textbook representation of 3D space, with the base (or reference) plane deliberately chosen so as metaphorically to represent the ground (in order to orient the user).

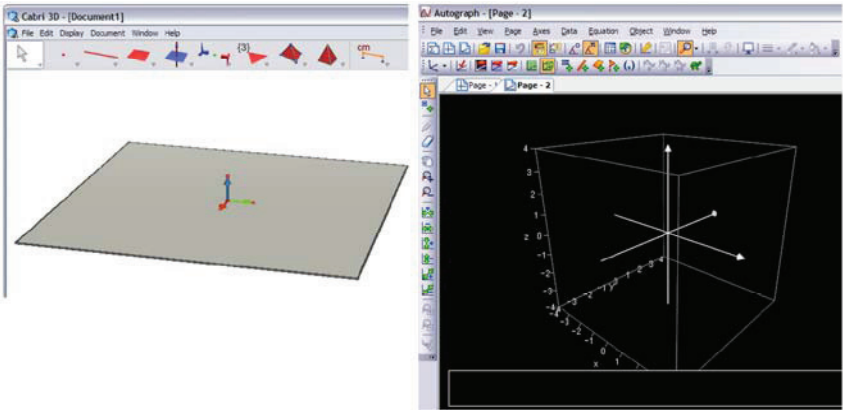
Cabri 3D - [Document1]
File Edit Display Document Window Help
Autograph - [Page - 2]
File Edit View Page Axes Data Equation Object Window Help
Page - 1
Page - 2

Fig. 2. opening screens of *Cabri 3D* (left) and *Autograph 3* (right).

The opening 3D screen of *Autograph* (version 3) shows a framework of a cube bounding 3D space from -4 to 4 on each axis (see Figure 2). This design was chosen as being likely to encompass most objects of interest at the relevant level of school mathematics. The scale and numbering of the axes is given along the edges of the framework so that labels do not ‘float’ through objects created within the cube. When objects are created, only the parts of the objects within this bounding box are displayed on the screen, the bounding box being chosen as a means of making this active area of the screen visible.

Even more so than with 2D software, the designers of 3D geometry software have to make a number of decisions about the ways in which objects are seen on-screen. For example, given that in 3D software a point in space is created by clicking in any empty screen location, a decision has to be made about the location in space of such a point, as a screen location does not define a unique point in space. In *Cabri 3D*, such a point is positioned on the base plane at the position of the cursor. *Autograph* locates such a point halfway through the bounding cube and along the observer’s inferred line of sight through the cursor position when the point is created.

Another set of decisions is about the way 3D objects ‘look’ on-screen. For an object and its surfaces to have a 3D appearance, use is made of perspective and ‘rendering’ (the computer graphics term for the ways in which the visual appearance of a 3D on-screen object depends not only upon its geometry but also upon the viewpoint by making use of lighting, shading, and, where appropriate, texture). In terms of perspective, the default for *Cabri 3D* and for *Autograph 3* is one-point perspective. In *Cabri 3D*, the default viewing distance is 50 cm, representing the screen at arm’s length from the viewer’s eye, chosen as it was thought to be ‘natural’. The viewing distance was more subjectively chosen for *Autograph 3* and is shorter. In terms of ‘rendering’, both *Cabri 3D* and *Autograph 3* use shading (by which the brightness of a surface is dependent on the direction in which it is facing relative to the inferred observer); *Cabri 3D* also uses ‘fogging’, a computer graphics terms for the effect by which objects ‘at a distance’ appear to be fainter than objects ‘close at hand’.

A further set of decisions relate to dragging objects using the mouse. Given that dragging on a flat screen can only give motion in two dimensions, in *Cabri 3D* a decision was made that ‘ordinary’ dragging would move a free point (or object) parallel to the base plane, while pressing ‘shift’ at the same time as dragging would move the point (or object) perpendicular to the base plane. In *Autograph* (version 3), dragging a free point continues to position it halfway through the bounding cube along the line of sight of the observer.

Given the centrality of ‘dragging’ in 2D DGE and its implications for developing different types of reasoning (Arzarello, Olivero, Paola, and Robutti, 2002), and as dragging is something which might make motion in 3D (on the 2D screen) more difficult to interpret by the user, the various aspects of dragging in 3D DGE are issues that could usefully be the focus for research.

4.3.3 Software for various non-Euclidean geometries

The ‘turtle geometry’ of *Logo* can give rise to several types of non-Euclidean geometry, each of which can be made available on the usual 2D computer screen. (Abelson and diSessa 1980). In *Logo*, a turtle’s ‘state’ is defined intrinsically (by reference to its own movement of forward-backward and its heading of turn left or right by so many degrees) and locally (since measures of steps and amount of turn are referred only to the turtle, not external coordinates). As a result, curvature in turtle geometry is turn per step, and is intrinsic to the turtle’s behavior.

While the turtle is ‘viewed’ through the Euclidean lens of the flat computer screen, if the screen’s metric is changed so that the turtle’s steps are lengthened or shortened in each step (with its turns unaffected), then there is the basis for non-Euclidean geometries. The turtle still responds to forward and right in the same way, irrespective of the geometry, but adjusting the screen metric alters its behavior as if the turtle were in spherical or hyperbolic space. The effect is that the screen can be thought of as having a variable ‘temperature’ (Gray 1989): from this perspective, spherical geometry has a screen that increases in ‘temperature’ as the turtle moves towards the screen’s edge, while hyperbolic geometries get ‘cooler’ towards the edge. By ‘dashing’ the turtle’s path (see Figure 3) so that the dashes grow longer or shorter according to the geometry, the turtle’s steps are expanded or contracted by the ‘temperature’ of the screen. A corresponding speeding up or slowing down of the turtle’s movement occurs as it leaves dashes as it is moved. Angles are preserved in these worlds, so that they sum appropriately to more (or less) than 180 degrees in a triangle, depending on whether the screen gets ‘hotter’ (spherical) or ‘colder’ (hyperbolic) at the screen’s edge, respectively.

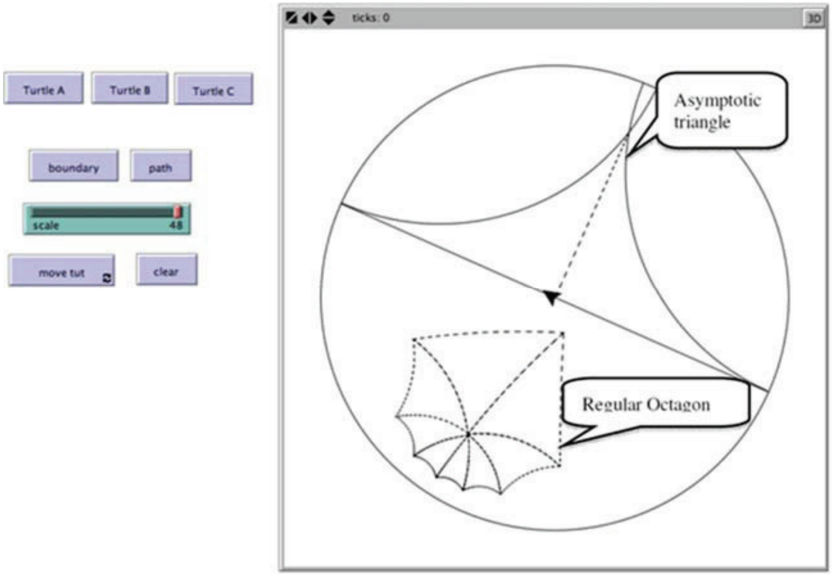

Fig. 3. Using *Logo* to create an asymptotic triangle in hyperbolic geometry.

The dynamic features provided via *Logo* are thought to play a significant part in helping learners to understand what is happening geometrically when exploring non-Euclidean geometries (Stevenson and Noss 1999; Stevenson 2000). Given that non-Euclidean models are obtainable through stereographic projection of a sphere or a hyperboloid onto the flat screen plane, this aspect of such models is thought to be critical in helping learners to understand the screen images. As illustrated in the next section of this chapter, such features can be used in the design of related learning activities.

Another form of non-Euclidean geometry that can be explored through utilizing the *Logo* turtle is fractal or ‘broken’ geometry (Mandelbrot 1975), formally defined as geometry in a space of a non-integer dimension and illustrated by objects such as the ‘tree’ and ‘snowflake’ in turtle geometry (Abelson and diSessa 1980). A snowflake, for instance, has an infinite perimeter, but a finite area – classic properties of fractal objects – leading to the scale-independent complexity of objects like the Mandelbrot set (Blanchard 1984; Mandelbrot 1980). Exploring fractals with Turtle geometry is thought to be powerful because such objects can be defined entirely in terms of four basic *Logo* commands (forward, back, right, left) and recursion.

Given the issues involved in the design of software for different geometries, we now turn to the issues in designing learning activities that attempt to realize the affordances, but take account of the constraints that are part and parcel of such software environments.

4.4 Designing learning activities to engage students with different geometries

As, when using 2D DGEs, dragging provides learners with an interactive way of validating their own constructions, much effort in task design has focused on encouraging learner conjecturing and on developing sequences of tasks that move pupils from conjectures to proofs (for examples, see Laborde et al. 2006). One interesting form of task is akin to a 'black box' (see Laborde 1998) by which learners are provided with a DGE figure for which they do not know the construction. The task is to construct a figure which has identical behaviour when dragged. Such a task is not possible with paper-and-pencil technology. This illustrates the powerful affordances of 2D DGEs. For more on task design for 2D DGEs, see, for example, Garry (1997) and Laborde (1995; 2001).

In designing learning activities for 3D geometry software (both Euclidean and co-ordinate), the complexity of the on-screen image, and the need for learners to orient themselves to a flat-screen representation of 3D, need to be taken into account. There may also be issues for users moving from 2D DGE to 3D software. For example, in 2D DGE the 'perpendicular' tool produces a line, while in *Cabri 3D* the 'perpendicular' does not produce a *line* perpendicular to a chosen line because the perpendicular to a line in 3D is a plane (and the perpendicular to a plane is a line).

Given such issues, the ways that the tools available in 3D software mediate the learners' understanding of geometry are only just being researched (see, for example, Accascina and Rogora 2006). In designing learning tasks, Mackrell (2008), for example, has found that Grade 7 and 8 students can be highly motivated to use *Cabri 3D* to create their own structures. Such structures included models of 'real-world' objects and/or objects that moved, with the creation of such structures necessitating the use of a range of mathematics. The 'flat' representation on the screen appeared to have an influence on student use. For instance, in order for an object to have a particular visual property when viewed from all angles (such as a segment being perpendicular to the base plane) the object needs to be constructed using the mathematical tool which creates the desired relationship (in this case the Perpendicular tool). Animation also appeared to be important in that it is only points that can be animated and hence other moving objects need to be constructed in relationship to the points.

Research on the use of software such as *Autograph* appears to be more limited, though teaching ideas involving the intersections of planes, and volumes of revolution (in Calculus) are provided by Butler (2006). More systematic studies of the use of software packages such as *Autograph* are needed.

In terms of the research on constructing a Turtle-based microworld for non-Euclidean geometry, several principles illustrate the importance of the interplay between design and learning, especially the learner-centered development of tools and activities that mediate understanding in specific geometries (Stevenson and Noss 1999; Stevenson 2000). In Stevenson's research, Papert's (1980; 1991) principle of finding links to cognitive development was a central design feature. These links emerged by working with learners to find what engaged them with the structures of the new geometries. Three types of links were needed to help learners connect with non-Euclidean turtle geometry because of the complexity of the screen images: physical surfaces and their projection, metaphors, and on-screen structures. Through tracing paths on the physical surfaces with their fingers, learners were able to make sense of what they saw on screen by metaphorically linking their action with the screen turtle. Utilization of the metaphor 'turtles walk straight paths' helped learners identify 'straight lines' on curved surfaces with straight lines left by the turtle on the screen (Abelson and diSessa 1980). By 'dashing' the turtle's path so that the dashes grew longer or shorter according to the geometry, learners were provided with an on-screen structure that indicated that the turtle's steps were expanded or contracted by the 'temperature' of the screen. A corresponding speeding up or slowing down of the turtle's movement as it left dashes, coupled with a tool that drew the large-scale path which a turtle might take given a particular position and heading, provided a dynamic structure for learners to build up their understanding. The key point here is that these physical, conceptual, and virtual resources emerged through looking for cognitive 'hooks' in these specific geometrical contexts.

Overall, the principle of iterative design (see, for example, van den Akker et al. 2006) is a feature of much work on learner activities as such a perspective pays careful and systematic attention to learners' needs. For example, in Stevenson's research on non-Euclidean geometries, the non-Euclidean microworld emerged through analysis of a series of structured activities and observations based on the relationship between the roles, tools and organization of resources over three cycles of development. It used a combination of didactic intervention, reflective discussions, task-based interviews and non-participatory observation of learners. Each of these roles was applied consciously in designing activities to achieve particular design objectives.

In this section, and in terms of switching attention to how learning activities are designed, what also needs to be acknowledged is how the activities are transformed *in use* by learners and teachers, and that feedback from task design can lead to further modifications of software design. As Harel (1991) points out, learning and designing are intimately connected, both for 'learners' and 'designers'. As a field, mathematics education has benefited from some useful connections between technology designers and users, perhaps no more so than in the area of geometry education.

4.5 Shaping, and being shaped by, digital technologies

In this chapter we have shown how key decisions taken by designers of digital technologies for mathematics are influenced both by the mathematics involved (in the case of this chapter by geometrical ideas of projection, curvature, local and global co-ordinates, and so on), and by the affordances of the available flat-screen technology. For more examples of the design process see Battista (2008), where a case study of the design of a 2D geometry microworld is presented, and Christou et al. (2006), where the theoretical considerations in the design of a form of 3D geometry software are revealed.

We have also examined the ways in which the design of learning activities is affected by, but also affects, the design of the digital technology. As we have illustrated, the software packages featured in this chapter exemplify how mathematics *and* learner needs influence the design of the digital technology, while, at the same time, the use of these digital technologies undoubtedly shapes the mathematical activity of the user. It is this symbiotic beneficial relationship that is continuing to offer so much – not only in the area of geometry education, but also as fruitful ways are being developed of linking geometry and algebra (Jones, in press).

Given that the book in which this chapter appears follows on from the very first ICMI study (Howson and Kahane 1986), it is appropriate to conclude by looking forward to the follow-up to this present study. It may be that, in another twenty years, we will have moved beyond flat screen technology, perhaps to a spherical screen for spherical geometry, and perhaps to ‘virtual reality’ (VR) environments which embed the user in space, something that is already being tested (see, for example, Kaufmann et al. 2000; Moustakas et al. 2005). In June 2007, *Flatland the movie*, an animated film inspired by Edwin A. Abbott’s classic novel, *Flatland* (originally published as Abbott 1884) was released. Perhaps, in due course, we can look forward to the release of *Flatland the VR game* in which the learner might take part as one of the ‘creatures’ in *Flatland* and experience (in ‘virtual reality’) what it is like to ‘live’ in a flat land.

Perhaps it is fitting to finish with raising the issue of just how ‘direct’ is what is often called ‘direct interaction’ when interacting with different geometries using digital technologies. As digital technologies for geometry develop, will users feel that they are interacting directly with geometrical theory; or will rapidly moving dynamic on-screen images seem more like computer-generated imagery (CGI) of the form commonly found in contemporary movies? How, we ask, can interaction with different geometries be facilitated through different digital technologies in a way which successfully builds the visual intuition that we need, as humans, to understand our experience of physical and mathematical space? We look forward to further research on such issues.

Coda

This chapter examines the design of digital technologies and associated forms of learning activities for a range of geometries. The purpose is analyzing how design is influenced by the mathematics involved, by the affordances (and constraints) of the available technology, and by the needs of the learner. If space had permitted an even longer chapter title, then the borrowing of E. A. Abbott's 1884 subtitle *a romance of many dimension*s could well be appropriate. While there is no space in this particular chapter for analyses focusing on other areas of mathematics (such as algebra or statistics), such analyses would usefully complement this chapter and are to be encouraged.

Notes

The main geometry software mentioned in this chapter (with publisher or contact in brackets) are as follows:

- *Autograph* (Autograph Maths)
- *Cabri* (Cabrilog)
- *The Geometer's Sketchpad* (Key Curriculum Press)
- *Logo* (Logo Foundation)

Source for Figure 1: *The Complete Woodcuts of Albrecht Dürer*. Edited by Dr. Willi Kurth 1963, Dover Publication, New York (illustration 338)

4.6 References

Abbott, E. A. (1884). *Flatland: a romance of many dimensions*. London: Seeley.

Accascina, G., & Rogora, E. (2006). Using 3D diagrams for teaching geometry, *International Journal for Technology in Mathematics Education*, 13(1), 11–22.

Abelson, H., & diSessa, A. (1980). *Turtle Geometry: the computer as a medium for exploring mathematics*. Cambridge, MA: MIT Press.

Arzarello, F., Olivero, F., Paola, D., & Robutti, O., (2002). A cognitive analysis of dragging practices in Cabri environments, *ZDM: the International Journal on Mathematics Education*, 34(3), 66–72.

Battista, M. T. (2008). Development of the *Shape Makers* geometry microworld: design principles and research. In G. Blume & M. K. Heid (Eds.) *Research on Technology in the Learning and Teaching of Mathematics, volume 2: cases and perspectives* (pp. 131–156) Greenwich CT: Information Age.

Bessot, A. (1996). Geometry and work: examples from the building industry. Iin A. Bessott and J. Ridgeway (Eds.) *Education for Mathematics in the Workplace* (pp. 143–158). Dordrecht: Kluwer.

Blanchard, P. (1984). Complex analytic dynamics on the Riemann sphere, *Bulletin of the American Mathematical Society*, 11, 85–141.

Butler, D. (2006). Migrating from 2D to 3D in 'Autograph', *Mathematics Teaching (incorporating Micromath)*, 197, 23–26.

Christou, C., Jones, K., Mousoulides, N., & Pittalis, M. (2006). Developing the 3DMath dynamic geometry software: theoretical perspectives on design, *International Journal of Technology in Mathematics Education*, 13(4), 168–174.

Clements, D. H., Sarama, J., Yelland, N. J., & Glass, B. (2008). Learning and teaching geometry with computers in the elementary and middle school. In M. K. Heid & G. Blume (Eds.) *Research on Technology in the Learning and Teaching of Mathematic, volume 1: research syntheses* (pp. 109–154) Greenwich CT: Information Age.

Garry, T. (1997). Geometer's Sketchpad in the Classroom. In J. R. King and D. Schattschneider (Eds.) *Geometry Turned On!: Dynamic software in learning, teaching, and research* (pp. 55–62). Washington, D.C.: The Mathematical Association of America.

Gawlick, T. (2004). Towards a theory of visualization by dynamic geometry software. Paper presented at the *10th International Congress on Mathematical Education* (ICME-10), Copenhagen, Denmark, 4-11 July 2004.

Goldenberg, E. P., & Cuoco, A. (1998). What is dynamic geometry? In R. Lehrer and D. Chazan (Eds), *Designing Learning Environments for Developing Understanding of Geometry and Space*. Hilldale, NJ: LEA

Goldenberg, E. P., Scher, D., & Feurzeig, N. (2008). What lies behind dynamic interactive geometry software? In G. Blume & M. K. Heid (Eds.) *Research on Technology in the Learning and Teaching of Mathematic, volume 2: cases and perspectives* (pp. 53–87) Greenwich CT: Information Age.

Gray, J. (1989). *Ideas of Space: Euclidean, Non-Euclidean and Relativistic*. Oxford: Clarendon Press.

Harel, I. (Ed.) (1991). *Children Designers: interdisciplinary constructions for learning and knowing mathematics in a computer-rich school*. Norwood, NJ: Ablex Publishing.

Hollebrands, K., Laborde, C., & Strasser, R. (2008) – Technology and the learning of geometry at the secondary level. In M. K. Heid & G. Blume (Eds.), *Research on Technology in the Learning and Teaching of Mathematic, volume 1: research syntheses* (pp. 155–205) Greenwich CT: Information Age.

Howson, A. G., & Kahane, J.-P. (Eds) (1986). *The Influence of Computers and Informatics on Mathematics and its Teaching*. Cambridge: Cambridge University Press.

Hoyles, C., Noss, R., & Adamson, R. (2002). Rethinking the Microworld Idea, *Journal of Educational Computing Research*, 27(1-2), 29–53.

Jones, K. (1999). Student interpretations of a dynamic geometry environment. In I. Schwank (Ed.), *European Research in Mathematics Education* (pp. 245–258). Osnabrueck: Forschungsinstitut für Mathematikdidaktik.

Jones, K. (in press). Linking geometry and algebra in the school mathematics curriculum. In Z. Usiskin (Ed), *Future Curricular Trends in School Algebra and Geometry*, Greenwich CT: Information Age.

Kaufmann, H., Schmalstieg, D., & Wagner, M. (2000). Construct3D: a virtual reality application for mathematics and geometry education. *Education and Information Technologies*, 5(4), 263–276.

Kreyzig, E. (1991). *Differential Geometry*. New York: Dover Publications.

Laborde, C. (1995). Designing tasks for learning geometry in a computer-based environment. In L. Burton & B. Jaworski (Eds.), *Technology in Mathematics Teaching: a bridge between teaching and learning* (pp. 35–68). London: Chartwell-Bratt.

Laborde, C. (1998). Visual phenomena in the teaching/learning of geometry in a computer-based environment. In C. Mammana & V. Villani (Eds.), *Perspectives on the Teaching of Geometry for the 21st Century* (pp. 121–128). Dordrecht: Kluwer

Laborde, C. (2001). Integration of technology in the design of geometry tasks with *Cabri-Geometry*, *International Journal of Computers for Mathematical Learning*, 6(3), 283–317.

Laborde, C., & Laborde, J.-M. (2008). The development of a dynamical geometry environment: Cabri-géomètre. In G. Blume & M. K. Heid (Eds.), *Research on Technology in the Learning and Teaching of Mathematic, volume 2: cases and perspectives* (pp. 31–52) Greenwich CT: Information Age.

Laborde, C., Kynigos, C., Hollebrands, K., & Strasser, R. (2006). Teaching and learning geometry with technology. In A. Gutiérrez & P. Boero (Eds.), *Handbook of Research on the Psychology of Mathematics Education: Past, Present and Future* (pp. 275–304). Rotterdam: Sense Publishers.

Mackrell, K. (2008). Cabri 3D: An environment for creative mathematical design. In P. Liljedahl (Ed), *Canadian Mathematics Education Study Group Proceedings 2007 Annual Meeting*. Frederickton: University of Frederickton.

Mandelbrot, B. (1975). *Les Objets Fractals: forme, hasard et dimension*. Paris: Flammarion.

Mandelbrot, B. (1980). Fractal aspects of the iteration of $z \rightarrow \lambda z (1-z)$ for complex λ and z, *Annals of the New York Academy of Sciences*, 357, 249–259.

Moustakas, K., Nikolakis, G., Tzovaras, D., & Strintzis, M. G. (2005). A geometry education haptic VR application based on a new virtual hand representation. In *Virtual Reality 2005 Proceedings IEEE*, Bonn, Germany, March 2005

Norman, D. A., & Draper, S. W. (1986). *User Centered System Design: New perspectives on the human-computer interaction*. Ablex: New Jersey.

Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. New York: Basic Books.

Papert, S. (1991). *Constructionism: research reports and essays 1985-1990*. Norwood, NJ: Ablex [edited by I. Harel & S. Papert]

Scher, D. (2000). Lifting the curtain: the evolution of *The Geometer's Sketchpad*, *The Mathematics Educator*, 10(2), 42–48.

Stevenson, I. (2000). Modelling hyperbolic space: designing a computational context for learning non-Euclidean Geometry, *International Journal of Computers for Mathematical Learning*, 5(2), 143–167.

Stevenson, I., & Noss, R. (1999). Supporting the evolution of mathematical meanings: the case of non-Euclidean geometry, *International Journal of Computers for Mathematical Learning*, 3(3), 229–254.

Van den Akker, J., Gravemeijer, K., McKenney, S., & Nieveen, N. (Eds.). (2006). *Educational Design Research*. London: Routledge.