Game Making and Coding Fluency in Practice

Mick Chesterman https://orcid.org/0000-0002-8005-2390

## Introduction and Context

### Intro and contents

STILL IN DRAFT FORM

In this chapter I explore the potential of digital game making as an inclusive way of developing coding concepts and general coding fluency in the later stages of Primary Education.

This chapter contains (CHANGE THIS LATER TO PROSE not bullet points): - a summary of UK context of coding and inclusion - a brief summary of research relating to game making as an educational practice - a section on alignment of PBL, UDL and game making - two game making tools used in my Study - research vignette - evolution of research design - Overview of The 3M Game Making Learning Design - Summary of Supporting Resources

### Context

In the UK the influential report “Next Gen: Transforming the UK into the world’s leading talent hub for the video games and visual effects industries” was focused was on providing the UK games and animation industry with the talent needed to succeed <Do Zotero Refresh: [@livingstone\_next\_2011]>. The three top recommendations school level were to include computer science in core curriculum, introduce a new computing GCSE, bursaries for computing teachers and extensive and well supported use of games and visual animation in the school curriculum as a way to attract more young people to the subject. The After the Reboot report <Do Zotero Refresh: [@waite\_pedagogy\_2017]>, returned to the subject of game making as a way of increasing engagement in the process of coding. The review highlighted several area of promise which needed more research, namely; using games for engagement, use of game patterns and involvement of girls in coding and social and cultural aspects of coding. The After the Reboot report also had concerning observations. The report found that girls, ethnic minorities and students of lower socio-economic status were all less likely to take computing as a subject at GCSE level. Given this concern it is important that as educators we explore promising tactics for inclusion including the use of games and game making. Game making aligns well with the principles of inclusive practices and project-based learning (PBL) namely: that more learner choice in projects increases motivation, authentic and shareable project outcomes encourage peer feedback and reflection project iterative support and a mastery approach, supporting challenging goals encouraging self regulation and structured guidance for goal setting.

Research by the UK National Literacy Trust <Do Zotero Refresh: [-@picton\_video\_2020]> of 11-16 year olds found that 96% percent of boys and 65.2% of girls play video games. This study shows that while there is a disparity between genders, game playing is still extremely common. You are unlikely to be part of a household where no games are played. Thus, even if not all young people play games they are extremely likely have knowledge of the conventions and culture of video games. This allows educators to draw on these interests and experiences. Research on the exclusion from the culture of computing highlights the value of use of games and playful techniques to build connections to home interests and experiences <Do Zotero Refresh: [@kafai\_diversifying\_2017; @kafai\_beyond\_2014]>.

Some of the key areas of potential for learners an area explored in depth in a review by Kafia and Burke <Do Zotero Refresh: [-@kafai\_constructionist\_2015]>. The most prominent learning objective of making games in educational setting is to develop *coding and computing skills*. There are extensive studies on *game making to learn other subjects* common subjects include maths, biology and chemistry but diverse examples exist. The potential benefits of collaborative game making align closely with a concept called *21st Century Skills* which include social skills, self reflection, cultural awareness and a range of technical abilities that allow participation in information society <Do Zotero Refresh: [-@baytak\_case\_2010]>. An area that merits particular attention is that that of Games to explore systems and systems thinking. Games are in essence rule-based systems. *Games for change* is a concept that invites game makers to make games to explore social and environmental issues. Such issues often involve a systems based understanding of then world, and as games are themselves interactive systems themselves, they are a powerful vehicle for exploring a complex problems involving race, sex, social issues <Do Zotero Refresh: [@tekinbas\_quest\_2010]>.

While there has been a large body of research on the value and practice of game making for educational purposes, it is a dynamic landscape which has many areas which still deserve additional research. New software tools to make games emerge regularly which offer new pedagogical possibilities. Game playing practices also continue to evolve. For example, casual games played by both adults and children continue to become increasingly available via smart phones and home consoles. The cultural impact of nostalgic retro games which are played by young people and families is also significant. The aesthetics of cuteness and nostalgia around retro games creates a potential for connection between younger and older players and a possible entry point for more casual game players in to game making cultures. The reduced complexity of retro themed games also reduces technical barriers for enthusiasts to create games either for the Indy Game market or for free distribution via games enthusiast websites like itch.io <Do Zotero Refresh: [@garda\_nostalgia\_2013]>.

A SUMMARY HERE AND LINK TO NEXT SECTION - POINT TOWARDS GAPS IN RESEARCH

## Game Making, Project Based Learning and Inclusion

IS THIS NEEDED - WORK ON OTHER STUFF FIRST - FOCUS ON MOTIVATION INSTEAD AND EXPLORE PBL AND INCLUSION IN RELATION TO THAT?

Recent studies study the use of games and playful techniques to overcome exclusion from the culture of computing <Do Zotero Refresh: [@kafai\_diversifying\_2017; @kafai\_beyond\_2014]>.

Contemporary understandings of inclusion go beyond SEN issues to include cultural exclusion. If students feel excluded from the school cultures then making bridges to home cultures is vital. One way to make those connections to home cultures is to allow for more choice of what can be incorporated into computing projects.

The inclusive possibilities of design and project-focused computing include; more learner choice in projects increases motivation, authentic and shareable project outcomes encourage peer feedback and reflection project iterative support and a mastery approach, supporting challenging goals encouraging self regulation and structured guidance for goal setting.

FIND SUPPORT IN THE RESEARCH HERE - DEVELOP IN MORE DETAIL? PERHAPS LINK TO UDL AS A FRAMEWORK ?

Certain characteristics of game making as an educational activity align well with inclusive and project-based approaches. Here are three examples.

**Family Game Experience as an inclusive Fund of Knowledge**

The concept of Funds of Knowledge emerged from research within US Latino communities. The term addresses the use of knowledge and skills from participation in activities outside of school that teachers can build on to help classroom work and to support the motivation of learners. Researchers found that Latino home cultures, skills and traditions were hardly visible in mainstream school cultures, resulting in a form of deficit thinking about the performance of these communities <Do Zotero Refresh: [@moll\_funds\_1992]>.

Games may allow children to draw on funds of knowledge in various ways, perhaps in the choice of the kind of game that is to be made, in the setting or subject matter or the style and aesthetics of audio and visual elements of the game. Teachers can also draw out attitudes and knowledge of game cultures and bring them into the learning environment in an inclusive way. In addition knowledge of game design conventions can be used by teachers to exemplify coding concepts. For example take conditional coding constructs. If Pac-Man touches a ghost then a player life is lost. Such structures are described as a game design patterns. Werner and colleagues <Do Zotero Refresh: [-@denner\_using\_2014]> found that the use of design patterns and game mechanics when teaching novice coders can increase accessibility for learners due the concrete and relatable approach.

**Game Making as an Authentic Activity**

Another important concept in both project and inclusive approaches to education is to make projects as authentic as possible to increase learner motivation. For game making this authenticity or realness can be seen in both to the tangible, shareable nature of resulting game created and in clear links to the domain of professional game design. When learners are designing with someone else in mind, this guides them to shape their game design effectively. The process of imagining the end user’s experience is a vital design skill that can be developed when making games. As teachers, it is helpful to redirect the attention of learners back to the imagined player of the game they are creating to help with motivation and prioritisation. The high-profile of the games industry helps learners recognise that their own game making skills can be applied outside of the classroom. Young people may not be able to create a technically commercial advanced game but other genuine audiences exist. For example, so-called Indy Games are made by enthusiasts and often released at low cost or for free on the internet. They often appeal to a retro-game aesthetic and are thus easier and quicker to make. Highlighting these communities and outlet may reduce student dissatisfaction at not being able to code more advanced games. As another way to increase authenticity schools sometimes enter online game making competitions or wider creative competitions like the Coolest Project. You may be able to add authenticity in a similar way by providing a low pressure competition or another frame for student’s game making.

**Coding and Computational Fluency**

Resnick and Rusk <Do Zotero Refresh: [-@resnick\_coding\_2020]> draw on literacy-for-all movements when using a concept of computational fluency to describe student’s creative expression through coding projects. Fluency in coding can be compared to fluency in spoken languages where the focus is less on accuracy and complexity of language use compared to how unhalting and comfortable the speaker is. Game making has a great potential to develop coding fluency if students are given flexibility over how they add in and adjust new features based on the motivation of designing for other players. Game making encourages small repeated changes to project variables and structures to get the game play feel of game mechanics just right. The process of adding different graphics and audio assets into games returns a high reward for students in terms of their efforts. These factors contribute to games being a good vehicle to encourage confidence in coding and computational fluency.

## An Overview of Game Coding Tools

The field of game making is extremely dynamic as new tools frequently emerge with novel approaches and features. In this section I outline the key features of selected game making tools. Some game making tools for beginners, for example Kodu or the Game Maker series, use a Graphical User Interface (GUI) to abstract away the detail of the underlying code complexity. While those may be useful to learn game design without coding, my study addressed tools that allow users to code the game directly.

After research two tools emerged as significant based on the opportunities provided. These were MakeCode Arcade a specialised block based programming and Phaser.js an authentic web game making framework used in a browser-based programming environment (code playground)

**Phaser.js in a Code Playground - an authentic web game making framework:** Phaser is a javascript game making library. It is my own tool of choice when it comes to game-making using text code. To teach it I ask learners to code games a web coding environment called a code playground. Code playgrounds are a tool used by both expert and novice coders to share examples of code that can be edited and preview online. A key feature is the ability to make changes in code and quickly see the new results appear in the live game. The concept is particularly useful for novice coders. Many text-based code playgrounds exist online. I chose Glitch.com although the process also works well in Trinket which is promoted by Code Club resources. Using this kind of web playground is a particularly authentic choice of tool. Phaser is used by professional game makers and Glitch is the test bed of choice of many code developers. If learners do take to this way of working they can easily progress to creating genuine Indy Games, dynamic websites and flexible web applications.

*phaser simple game to edit* <Game\_Figure1.1.png> Figure 1.1. Phaser in Glitch.com playground with code and game side by side. Screenshot by author.

The drawback of this approach is the potential complexity of using real web technology. While it is possible to hide certain elements of html and css away from the user, many mistakes are possible which break the game completely. Luckily Glitch has the ability rewind and undo your changes via an easy to use timeline of your project.

**MakeCode Arcade - specialised block based programming:** MakeCode Arcade is a block based programming environment similar to Scratch but with some interesting features which are tailored to game making like gravity, lives and a game over block. In addition, the multi-media making abilities are very stripped down, you can download the games to hand held devices or run them easily. Another advantage is that the MakeCode system is also used to code the popular Microbit micro controller. So this may be familiar to you as a teacher or to your students.

<Game\_Figure1.2.png> *MakeCode* Figure 1.2. MakeCode Arcade with code and game side by side. Screenshot by author.

The MakeCode Arcade interface, like Scratch has the ability to edit sprite characters. However, compared to scratch it is more limited in terms of what can be created. The simplicity of a tool like MakeCode Arcade can help address issues around learners trying to attempt projects that are too ambitious in scope. The designers have intentionally limited the tools provided to work on audio and graphical assets and the screen size of the game. This simplicity reduces the amount of time learners spend creating assets and thus support students to spend time build familiarity and fluency with coding constructs. There some engaging, diverse and comprehensive example games and tutorials at the project website.[[1]](#footnote-2)

## Research Vignette - Evolution of Design

RESTATE / DESCRIBE RESEARCH GOALS CONCISELY

My own research in game making is an experimental approach to create a new learning design. I have worked with young learners, local families and undergraduate student helpers to evolve a game making design. A key driver of my research was to explore the potential to draw on family experience in learning activities by working with families to make games together is a fertile research base to jointly create learning activities with a wider potential application.

To facilitate this goal I have taken a design-based approach. Design based research It was originally developed as a response to the importance of context in educational research as part of the social turn of social science research <Do Zotero Refresh: [@brown\_design\_1992]> .

DPR is a varied discipline which can take a multitude of forms. The core elements include: research as an intervention, iteration, involvement of participants in the evolution of designs, a flexibility of research outcome based on how events unfold <Do Zotero Refresh: [@easterday\_design-based\_2014]>. One of the key motivations of this approach is to produce educational research that has a high utility for practitioners in that it which develops theory that is rooted in contextual practice.

Barab and Squire <Do Zotero Refresh: [-@barab\_design-based\_2004]> describe the messiness of design based research and that this creates a challenge to the researcher of how to present results in a coherent way which is of use to other practitioners. There is a tension between sticking closely to the context of the research and the concrete specifics or stepping back to generalise and being lost in abstraction. As this is early presentation of results I try to strike a balance which stays concrete but which also pulls from my observations a framework which may translatable in to other game-making and creative project based approaches.

To address my research interests of engagement, autonomy in the design of the experiment I started from a very open starting position. The main intervention was to lead a workshop with participants challenged to make a simple computer game. Beyond this little was decided in in advance with the goal being to respond to the context of the learning environment.

My current game making designs have emerged from several iterations of experimentation. The main experimental team consisted of Home Educating families. The designs that emerged where then also trialled in a local primary classroom of Year 6 (10-11 year old). For some iterations of the design I also worked with undergraduate students who acted to support the game-making workshops. Through informal responses, direct feedback, and structured evaluation all participants acted as research partners as I acted on their reactions to design the next iteration of the game making program.

## Overview of The 3M Game Making Learning Design

The section outlines the main features of the 3M model which comprises of Missions, Maps and Motivational Methods. This learning design could can be applied using any of the software tools above. The resources I have created for MakeCode[[2]](#footnote-3) and Phaser[[3]](#footnote-4) are free and open source and available online. I invite other educators to adopt this approach and share resources for Pygame, p5.play and other suitable platforms. I will explain how the methods involved in the model are informed by inclusive pedagogy principles contained in Universal Design for Learning (UDL) and project-based learning (PBL).

### Missions

Many commercial open world games offer a central challenge consisting of small incremental missions and then optional side missions. The goal is to of increase user choice and give players more agency. To mirror this approach, the main challenge of the 3M model is to create a playable game around a theme for a real or imagined audience. The incremental missions of the my implementations of 3M model involve adding game features. This approach steers students towards developing the use of coding structures, debugging practices and systems concepts. In addition, side missions encourage social and playful coding approaches which help develop a community of coders.

**Side Missions:** Bartle proposed that online gamers play games for different reasons (**???**). You can find out what kind of game player you are with an online test.[[4]](#footnote-5) There are also different styles of game makers. Some like to study to get a full knowledge of the tools and what is possible before they build up their game step-by-step. Some are happy to borrow code, images and sound from anywhere for quick results. More social makers like to find out about the games of others or tell stories within games while others mess around with the code to break it interesting ways. To encourage these valuable social coding practices I created extra missions which are available online.[[5]](#footnote-6)

**Game Design Patterns as Main Missions:** Design patterns are most commonly used for computing students at higher education to teach object oriented computing but they are also useful for all levels of learners. Design patterns are rooted in real-life incidences of problems that are often solved in a particular way. They are concrete examples of coding principles in context. Design patterns can help the development of coding communities if more experiences coders take the time to document the patterns they use in an accessible way for novice coders. For educators the use of design patterns can help support learners develop coding proficiency by providing scaffolding and modelling good design decisions. However, one of the challenges for teachers of using worked examples and design patterns is how to integrate them into student-led design challenges. In the 3M model rather than following a step-by-step tutorial learners start with an incomplete game template and add new features as they choose. Each feature is described as a mission. This approach follows the Use-Modify-Create model to limit learner anxiety for novice coders and to scaffold the acquisition of coding and computational thinking concepts <Do Zotero Refresh: [@lee\_computational\_2011]>. I worked with learners to create a wish list of game features to create a 2D platform game. These features included moving hazards, jumping on enemies, finding a door or flag to progress to the next level. We can describe these features as game design patterns. I developed tutorials to support students implement these patterns and grouped them to aid navigation of the resources. This approach aligns with inclusive education principles in that it increases the choices of students, scaffolds the way they can access resources and allows them to keep a track of their own progress. In my research I found that providing a set of limited options but allowing student to choose their own pathway had a immediate positive effect on the motivation and general energy level of learners.

### Game Design Pattern as Mission for Students

In my version of the 3M model students picked missions from a choice of printed cards. There were four colour themed categories of missions. *Game mechanics* are features to do with the actions of the game. *Game space* patterns address the layout of the game. *Game polish* patterns involve adding music, backgrounds, graphics and story elements. Finally *System and Challenge* patterns look at how different elements interact to create challenge in the game. An example of a game mechanic design pattern follows.

BOX BEGINS

Your mission is to apply the following pattern to your game. There are supporting step-by-step resources available if you need them. When you finish be sure to reflect on how adding this pattern helps your understanding of the computing concepts and similar patterns listed. This concludes your mission.

* **Name of Game Design Pattern:** Jumping on Enemies to Zap them
* **Type Pattern:** Game Mechanic
* **Description:** If the player is descending from a jump when they touch the enemy the player is zapped and in this case disappears.
* **Need for Pattern:** Enemies create challenge and being able to jump on an enemy is a good way of clearing the area you want to explore or to be able to jump up to a high platform.
* **Coding Concepts involved:** Arrays[[6]](#footnote-7), Change Listener[[7]](#footnote-8)
* **Links to other Computing Patterns:** Systems Dynamics[[8]](#footnote-9)
* **Related Game Patterns:** You’ll need to have added the **Add Enemies** pattern to your game before you can add this one.

BOX ENDS

In addition to outlines of game patterns, print-outs or on-line documents to support learners to them implement the code needed are provided. While on-line documents allow learners to copy and paste code thus avoid many syntax errors, printed or incomplete code examples provide a greater level of challenge. Supporting resources help resolve tensions around learners getting stuck and needing a lot of facilitator help. You can vary how much detail you provide in supporting documents to suit the challenge level for students. I work with young coders, thus I normally provide significant coding scaffolding. Once learners have built familiarity with code structures, processes and the coding environment, I provide less complete code examples.

### Maps

**Learning Dimensions Map:** In learning environments where there is a lot of choice assessing learners via observation is beneficial. Rather than deciding what you want to teach and testing on that area, you can map the learning happening in an authentic activity. When researching hands-on tinkering in Science museums Bevan and Petrich <Do Zotero Refresh: @petrich\_it\_2013> worked with educators to examine video footage of families interacting with exhibits to make a structured list of the learning they observed. The resulting map of learning dimensions included underlying science concepts but also contained more general skills and helping behaviours involved in exploratory learning processes. Another chapter in this collection identified concept maps and observation as methods for teachers and researchers to identify key learning suited to particular computing projects. One of the outcomes of my research was to extract from hours of recorded material some of the concepts and practices that learners engaged with when making games. While some are common to existing Computational Thinking frameworks others, including systems thinking concepts, are more unique to game making. Table 1.1. shows my resulting map of learning dimensions for the 3M game making model.

| Coding Concepts | Systems Patterns | Design Practices |
| --- | --- | --- |
| Sequences | Systems Elements | Goal Setting |
| Variables | Systems Dynamics | Being Incremental and Iterative |
| Logic | Reinforcing Feedback Loops | Developing Vocabulary |
| Loops | Balancing Feedback Loops | Web Navigation |
| Arrays |  | Problem Solving |
| Creating Functions |  | Version Control |
| Change Listener |  | Debugging |
| Input Event |  | Reusing and Remixing |

Table 1.1. Learning Dimensions of the 3M Game Making Model

This process of mapping such frameworks may be overly time-consuming for many full-time teachers. However, teachers may also use and adapt existing maps and frameworks based on their own classroom experience to help their observation of students. Because these frameworks can also help students to navigate their own learning journey the effort serves a double purpose.

**Physical Maps of Missions:** To support younger coders unsure what to do next or who struggle to stay on task, I sought to create another kind of map to help them navigate their game making journey. I printed out a large-scale colour map of a coastal landscape stylised in a way that mirrored a map used for navigation in an quest-themed computer game. The game pattern missions where represented as different islands . Learners took time to create and personalise a movable marker representing themselves. When learners selected their next mission, they moved their counter to the relevant island. Thus learners had to be intentional about their next goal and were implicitly encouraged to stick to it. They also kept a track of the missions that they had completed by tracing a trail as they progressed. In addition the colourful, physical and visual representation served to encourage a sense of community and peer learning. When moving the counter on the map I prompted them to reflect on the coding concepts or other learning dimensions that they have been working with. As learners traced a trail between the different island/missions they had visited, the map provoked learners to reflect on their journey and progress. This approach may be too labour and time intensive for many class environments. I am investigating replicating this process using online tools to reduce complexity and preparation time.

### Motivational Methods

The final M of the 3M framework stands for motivational methods. There are many ways to motivate and sustain the learners involvement PBL. In this section I share two methods that I have found valuable in the context of game making.

**Physical Computing and Game Making:** The use of physical computing to create concrete and tangible activities can increase the engagement and motivation of learners <Do Zotero Refresh: [@kaloti-hallak\_students\_2015]>. Making the digital concepts physical, and thus allowing exploration via diverse means, also aligns with inclusive learning principles. To support my game making projects, I created simple arcade cabinets out of wood with retro arcade buttons. Connecting arcade buttons to the computer via simple electronics is a project which can be completed quickly. The process of students building their own arcade cabinets for a games showcase increased their perception of the authenticity of their end goal. Some families created low-tech, customised arcade cabinets using cardboard. Although my studies have been small-scale the self-reported effects on learner engagement and motivation of this part of the program were significant.

**Drama / Fictional Frameworks:** Another method I use to increase learner engagement in game making is the concept of using a fictional scenario or simulation. A fictional community while less authentic than a professional community, can still provide some of the associated benefits of authenticity. I have worked with practitioners of Drama Education department at Manchester Met to develop such fictional dramas, but you don’t have to be a trained drama practitioner draw on key techniques to increase learner engagement. For example, I asked trainee teachers to devise a scenario to support a series of sessions and they used a fiction of making games for a alien race coming to destroy the earth. The process of using a fictional situation can help with the motivation and reflection of learners in the following ways;

* Asking learners to step into a role can increase identification with participation in the project. For example you may say “As game designers, we will make this game for a particular audience”.
* Fictional situations can help create a sense of imagined jeopardy which can help learners stay on track with their creative timescale and may increase their commitment to the process.
* When learners share their games with their real or imagined audience, they can talk through their design decisions and challenges, thus creating an opportunity for reflection.
* Drama processes can help explore identification with or hostility to gaming cultures.

## Supporting Resources

This section contains links and descriptions of supporting resources that illustrate and research model.

### MakeCode Arcade Resources

THIS SECTIONS NEEDS FLESHING OUT.

The following resources are available and freely adaptable under a creative commons BY-SA licence.

* An overview of resources - https://mickfuzz.github.io/makecode-platformer-101
* An interactive starting template and patterns to add is here - https://mca-platformer-examples.glitch.me/
* A five week course adaptable by teachers - https://mickfuzz.github.io/makecode-platformer-101/groupCourse

### Phaser and Glitch.com Resources

Similar resources exist for text based coding in Phaser. These include:

* A splash page for those Phaser / Glitch resources is here - https://glitch-game-makers-manual.glitch.me/
* An interactive starting template and patterns to add is here - https://ggc-examples.glitch.me/

## Conclusion

In this chapter we have examined how game making fits an inclusive and project-based approach to computing. I outlined some of the potential that make game making provides in to be an authentic activity and how it allows students to incorporate their own interests and home experience into projects. The chapter then listed the diverse motivations and key concepts involved in undertaking game making projects in education. Finally, to explore some of the particular tensions and opportunities involved in undertaking project-based game making with young people I used a detailed case study of the game making 3M model. Each of the three methods helps to align game making project work with inclusive pedagogical approaches. For example, the use of game design patterns as missions helps scaffold the process of goal setting and project navigation. The use of maps helps learners to navigate their progress and can help teachers to facilitate a learner-led processes thus increasing student autonomy. Finally the motivational methods of using a fictional frame and the incorporation on physical computing techniques can help engage learners and to sustain their continued investment in the project work.

<Bibliography: Do Zotero Refresh>

1. https://arcade.makecode.com/ [↑](#footnote-ref-2)
2. https://mickfuzz.github.io/makecode-platformer-101/ [↑](#footnote-ref-3)
3. https://glitch-game-makers-manual.glitch.me/ [↑](#footnote-ref-4)
4. https://matthewbarr.co.uk/bartle/ [↑](#footnote-ref-5)
5. https://mickfuzz.github.io/makecode-platformer-101/missions [↑](#footnote-ref-6)
6. https://mickfuzz.github.io/makecode-platformer-101/learningDimensions#arrays [↑](#footnote-ref-7)
7. https://mickfuzz.github.io/makecode-platformer-101/learningDimensions#change-listener [↑](#footnote-ref-8)
8. https://mickfuzz.github.io/makecode-platformer-101/learningDimensions#systems-dynamics [↑](#footnote-ref-9)