4. Exploration of the Formative Learning Design Process

Exploration of the Formative Learning Design Process

Research Questions - April 2024

~1. What pedagogical tools and processes are available to support novices to overcome barriers to participation in game coding processes?~ 1. What barriers contradictions arose in participation in this research's game coding processes and what pedagogical tools and processes are available to address these contradictions? 2. How can game design patterns support the development of coding practices with novices? 3. How can learners build agency in an evolving community of game makers?

A note on style

This chapter contains a large number of headings in draft form. These will be replaced with introductory sentences when structure is stable.

Chapter Introduction

Due to the novel nature of the learning design, a detailed outline of the specifics of the participants experience of the design helps readers situate the findings of the following two chapters.

This chapter contains a high level of detail on the technical elements of the learning design to both both helps situate the findings of the next chapter and to allow the design by other practitioners and researchers in this area. The motivation here is that the processes of this research should be understandable and replicable by other practitioners and researchers.

 NOTE

I originally structured the chapter around observations structured in the phases described in chapter three. This approach proved over-complicated, mixing the description of tensions which often spanned multiple phases.

In line with key techniques of third generation activity theory (3GAT), this chapter addresses the evolution of the key pedagogical features of the tools of learning design used in my research process using analysis of elements of nested activity systems and emerging tensions in learning design.

Firstly, I describe activity systems at different scopes, focusing on the objective of participants.

Following this, I explore emerging tensions in design in a way which communicates relevant

context through the analysis of three areas of emerging contradictions in activity. Here the term contradictions is used as a wrapper for more specific tensions, barriers, breakdowns, conflicts that indicate an "unfit with (system) elements (Kuutti, 1995). Three main areas of contradictions are explored in this chapter.

- Addressing contradictions arising in use of coding tool use and game asset creation - often experienced as more individual level
- Contradictions associated with project navigation and the use of supporting documentation - interpreted here through the lens of guided participation
- Responding to tensions and barriers in cultural aspects of the game making activity

Within the last area, I examine the impact of a process drama with an aim to establish and nurture community practice, exploring the value of introducing almost-real processes, near authentic tools and the possibility of learners developing learning identities within a drama frame. The drama process is an example of an introduced cultural tool. The process acts as wrapper to several supporting techniques. the side missions to and to sanction different approaches, the imagined audience encourages reflection and semi-authentic approaches. It is the most advanced tools of a range that was put into place to create a space suitable for creativity, collaboration and self expression.

This chapter ends with a synthesis to initial barriers to engaging in game making practices and initial interventions to address them.

Detailed description of dimensions of game making activity

The purpose of this vignette is to illustrate some of the key features of the overall game making activity in context. The following vignette, consisting of extracts of participant activity over twenty minutes of a session in P2 (see below), shows activity happening at different scopes of activity. The participant, a child Toby, has been working independently on the design of a platform game for the first fifteen minutes of the session. Some specialist terms are used in the vignette which are explained in more detail in this chapter and in the glossary.

Illustrative vignette one participants activity indicating general activity of participants

The vignette is included as a principle appendix.

https://docs.google.com/document/d/
1vYeVxYaRMTWPDOHwC4DEGYAkGPRIDX7wLiz0l8b7LWc/edit

Description of interacting systems of activity illustrated using the vignette

In line with the process outlined in chapter 3, the following sections outline different scopes of activity, at times using detail from the vignette above, to describe the activity of participants and to introduce tensions between system elements which are explored in this chapter.

Environmental factors and objectives

In socio-cultural approaches is it important to situate the context of the learning environment within wider systems of activity. In line with Engestrom's concept of shared object explored in the previous chapter (Engeström, 2001), I have selected several relevant systems and represented them in an illustration 4.x below.

These include: research goals on the part of the researcher/ participant; desire to create a useful learning experience on the part of student helpers; learning computing skills and concepts as a home education project; participating with other home educators in a creative environment and thus building team and communication skills; and finally a broad aim of fun on the part of the children participants.
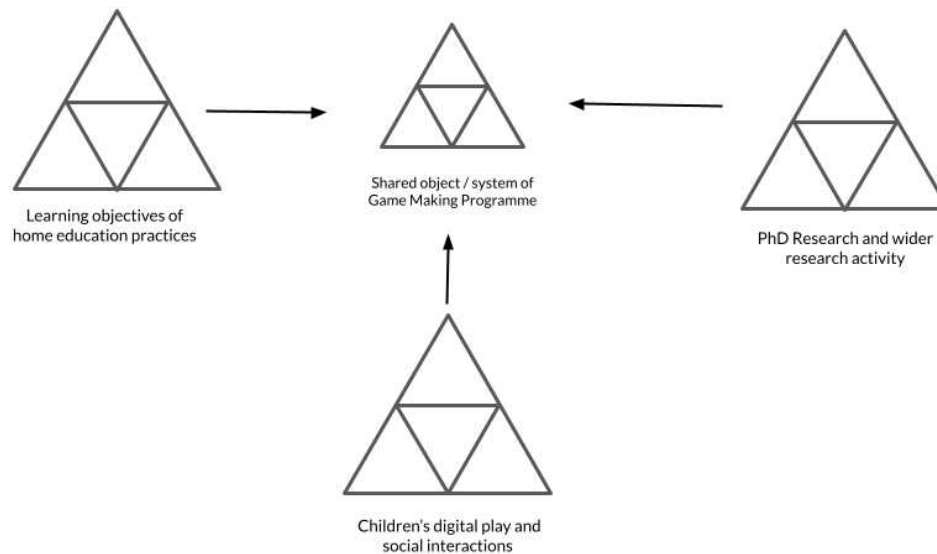
Illustration 4.x - Broad Environmental Activity Systems

In this research I take the approach of treating this shared object as an activity system in its own right.

Lecusay takes a similar approach (Lecusay, 2015) and describes the shared object as an ideocultural hybrid which develops into its own activity system.

Making games for an audience

The broader activity system whose object was the game making programme had a shared objective of making entertaining and educational games for an audience to play.

Tools: As above and: pixel art and sound editors, code examples, step-by-step tutorials, evolving concepts

Subject:
Pair or individual child or adult

Object -> Outcome:
Implement a game feature

Rules:
Choose achievable goals within technical scope of the project

Community:
Peer game makers
Student and researcher facilitators

Division of Labour:
Child as primary learner
Parent brokering access to help
Parent as active collaborator
Parent providing emotional support / encouragement

Illustration 4.x - Larger game making activity system

There are diverse end audiences for the games. The final target audience are students and staff in the Manchester Met Brooks building where the created games are shared in arcade cabinets at the end of the programme. An additional audience are friends and family who can be send the games to play online. The more immediate audience for the evolving games are peer game makers during the making sessions.
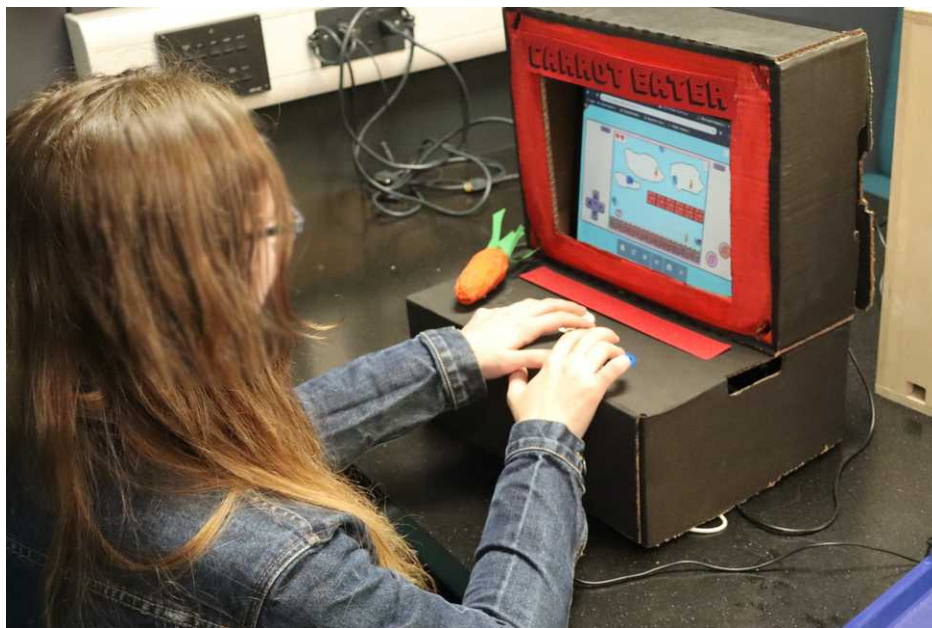
Carrot eater game with an self made arcade box made from cardboard

While the broad aim of making a game to share is tacitly agreed through participation, there are also diverse motivations stemming from environmental factors in the interrelated actors in this activity which bleed into the tensions which emerge at this level. In the vignette and surrounding data, Toby pays close attention to the challenge and variety of the game playing experience for his projected audience of players. In the vignette above, I draw the participants attention to the showcasing of their games to an external audience. Toby begins the session playing his own game. During the course of this session he invites many other group members to play his game. He initiates and responds to conversations around the difficultly of the game he has designed.

While the programme had a shared objective of making a game, there were tensions brought by the cultural factors and other objectives imported from other activity systems. This chapter explores those tensions.

Constructing a game feature by feature

Within the activity system of making a game

There is a strong case to make for the sub-projects of implementing GDP as a key unit of analysis. Barab and colleagues justify smaller units in their study observing emerging practices in a technology rich learning environment (Barab et al., 2002).

At this scope of activity, participants work on the game code and and audio and graphical assets to achieve the broad objective is to alter and add new features to the game using tools available in the learning environment.
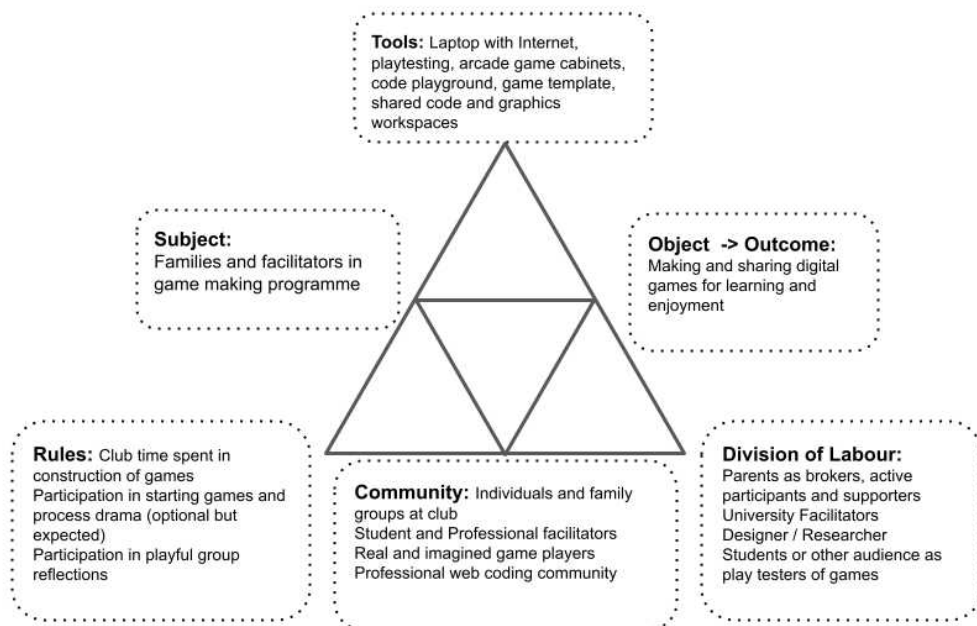
Mid-level activity system of game making

In addition game design patterns emerged as a shared object. DEVELOP THIS IDEA HERE OR

IN DISCUSSION.

Individual or Collaborative
Game Pattern Implementation

Shared object of concept of
game design pattern

Individual or Collaborative
Game Pattern Implementation

Illustration 4.x - Shared object of implementing GDP

Overall activity at this level was driven by learners setting goals and requesting help to implement game features. By this stage of the process, Toby is able to browse a collection of game design patterns and use diverse of tools, resources and processes. Participants developed preferred approaches and tool choices especially in accessing help when adding a new feature to their game. In the vignette above I outline the two main sources of help at the level of adding a new feature to the game: "Here's the tutorial and there's the examples of code". In the course of the vignette above, Toby chose to access only a code example of the desired game behaviour. He did not, as some in the group were doing in this session, use the tutor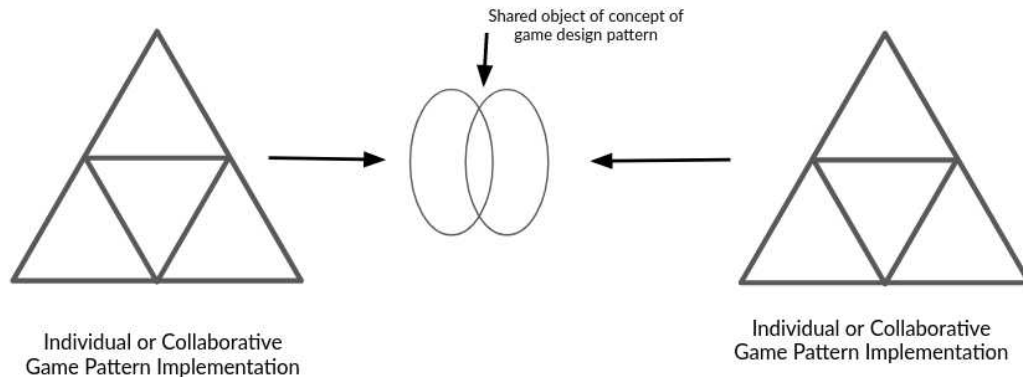ial which provided more descriptive, step-by-step guidance. Other participants relied only on direct verbal help from myself or other participants (like Toby) to help them progress. A later section of this chapter details the different supporting documentation guiding participation in game making practices. Chapter five then explores activity occurring at this scope in more detail with a greater focus on emerging concepts as tools.

Smaller objectives and actions - Implementing discrete code and design structures

While the previous section focused on a mid-level objective of adding or altering game features (also referred to as game design patterns here), implementing more complicated patterns involved several stages and varied tools. Using the terminology of Leontiev (2009), in the example above Toby undertakes certain chains of processes in a fluid way that shows that actions had become operations. An example of such an operation is his quick navigation between different areas of the game code, the game preview window and other sources of documentation. In contrast, some tasks are new to Toby and are performed more hesitantly. At times in extract above, Toby is careful and hesitant, checking and rechecking the process of copying and pasting new code into his game from a specially prepared code example.

Identifying shifts in participant activity in terms of scope facilitates analysis of the complex and interwoven cultural, social and personal actions in a communities activity (Rogoff, 1995).

Above, Toby shifts between the wider activity of making a game and sharing it with peers and narrower actions implementing concrete code structures. This aspect is developed in chapter five. In addition to these shifts in scope, analysis of community processes also involves transformation over time. The next section offers a description of the evolution of tool use in different phases of the study with an aim to situate analysis of emerging tensions in the use of tools, resources and processes in the following section.

Conflicts and design tensions experienced by participants and facilitator/s

This section examines design contradictions between system elements and tensions experience by participants which developed into conflicts which significant in stopping participant progress. The term conflict here is used to describe a more serious break down of activity due to an accumulation of contradictions and tensions.

The processes here aligns with AT approach of formative interventions in that the surfacing of tensions due to changes in the learning design over time are examined. I chart the detail of the evolution of resources and other responses used helps illuminate broader, systemic issues concerns in the overall activity (Barab et al., 2002).

As such, this chapter explores the following areas of tensions and barriers.
NOTE - (be clearer on terms still)

- contradictions in the tool set of the early design.

- contradictions which stimulated the development of authoring tools and supporting documentation.

- issues of identity and interventions designed to mitigate cultural barriers of participation.

Tensions due to lack of access and familiarity to game programming tools

Addressing contradictions arising in use of coding tool use and game asset creation - often experienced as more individual level

Conflicts arose due to a contradictions between the desire of participants to use the tools to create games and their lack of experience of them. FINESSE THIS WITH MORE DETAIL - accumulation

The process of helping participants build familiarity and competency in tool use was further complicated by my own motivations for working with as authentic as practical game authoring tools.

Activity theory has a broad definition of the concept of tool which in this context would include

the use of language, concepts, supporting resources and production processes. This section focuses on software tools used to create games in this program and technical process needed within those applications.

Thus crux of the conflict is that coding games to create a webpage using a text environment is a complex process. While some of the group had some experience, many did not. In early stages, this gulf created a tension between the need to use the tools and text coding processes which often resulted in frustration and paralysis in the activity of game coding. While, tensions stemming from gaps in knowledge and practice are commonly addressed in formal learning environments through forms of instruction, the informal nature of this learning setting, and early experiences where I found instruction based teaching uncomfortable, encouraged me to prioritise approaches which avoided written or to whole class instruction.

The resulted in a design challenge involved designing a bespoke game authoring environment by adapting existing tools and embedding where possible affordances within that toolset to facilitate participants to address gaps in their knowledge without needing explicit instruction. The following sections explore the evolution of the tool selection and the tool use of participants in response to this conflict in the game making activity system.

The motivation was to align to authentic web technology due to my previous experience. The reason for working with code this authentic bears exploration. Characteristics of broader dimensions of authenticity of tools and processes in learning environments are explored in the literature review. A brief summary being: similarity to workplace settings; facilitating activity which serves a need of participants; an ability to showcased resulting work to an audience (Shaffer and Resnick, 1999; Warr and West, 2020). While my choice of open source digital tools allowed for a high level of aligning with professional practices, it was also potentially complex in terms of use of tools. In part my motivation for choice of code language (Javascript, Phaser) was driven by desire to align with common workplace practices and to be potentially extensible for future student-led web projects.

By P2, learners first experience of the experience of the tools was in the process of playing an incomplete game in a webpage and controlling the character using the computers arrow/ cursor keys which for many was a familiar process. Due to an intentional fault, players needed to click a remix button, and alter the underlying code to progress in the game. The following section outlines responsive design process surrounding two main elements: the starting game code template and the code authoring toolset.

Design choices of the starter game template

FOCUS ON THE TENSION FIRST AND SHIFT FROM P1 -> P2

The initial use of a working structural template was an intuitive response based on my own experience of teaching technology. The choice to pre-select a particular genre was initially a pragmatic response to tensions experienced in P1. When offering feedback to address her family's feeling of isolation from the coding process, the parent of the family described in part

two of this section had more hands on play and use of the tools of production before being called on to make creative decision, likening this to an arts studio approach.

I realised that to allow for this playful experimentation, a purely structural template was insufficient and that a working game template was preferred. To create the template I drew on the structure of online starter templates from several sources to create a minimal template for a two-dimensional platformer game.

Structural design of the game template

A web code project using the chosen game framework phaser consists of several interlinked files of Javascript, HTML and CSS and image files. (see glossary & appendix 4.x)

While access to HTML and CSS files of the base project was available in the left menu as show in by default participants would see only the JavaScript file names game.js (See Figure 4.x below)

Variable editing for player movement

To accelerate and support the experimentation of users, I identified changes to the code that were easily recognisable game experience features and where small changes could provoke a high impact on the game experiences. These include changing gravity, altering the player jump height and walking speed.
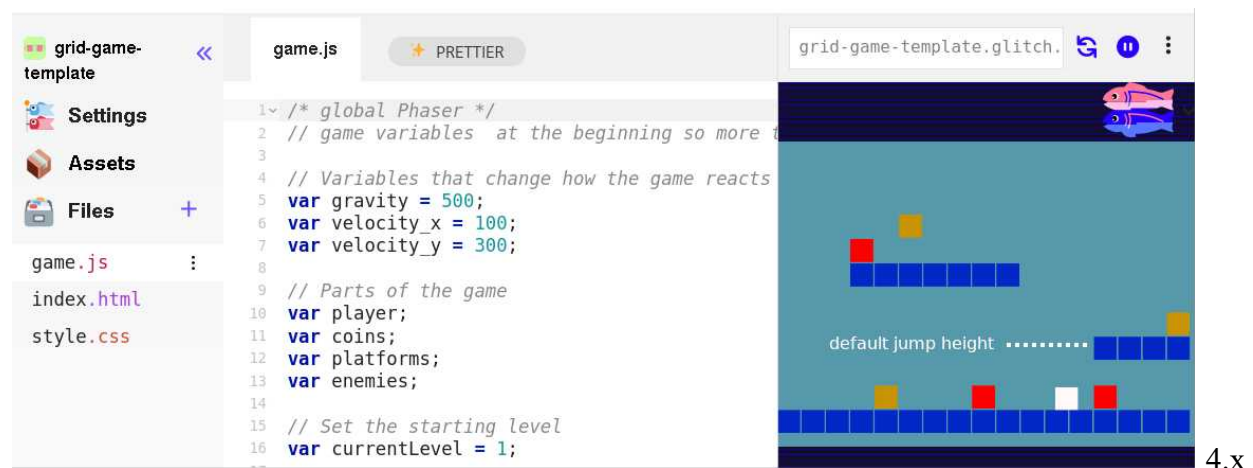
4.x glitch coding environment with code structure of left menu, a central code window with code, comments and game preview on the right.

The starting template began with the game in a broken state thus inviting players to modify the game to fix it. The player's maximum jump trajectory was not sufficient to progress via a jump to the first platform. To progress, participants needed to change alter at least one of the key variables were highlighted at the very start of the game code (see Figure above). In their research Kynigos and colleagues (2018) explore this concept as a half-baked games where incompleteness or bugs in behaviour are a provocation to participants to correct or to further modify them. This

10

process also aligns with the motivations and techniques of the UMC framework explored in the literature review , in particular the guideline to "create choices that show visible and immediate changes" (Lytle, Catete, et al., 2019:6). In this design, the first participant choice and the need to transition from the *use* to *modify* stages is forced at an early stage by the half-baked design. While this design decision compromises the user choice initially, it allows a carefully scaffolding of early coding experiences and promotes a shared experience for all participants in a way which facilitates and encourages peer learning. After this shared first change, participants next choices varied greatly. While some participants engaged with extensive experimentation to find a player movement feel that seemed just right, others, mostly adults or younger participants, were much less concerned with this aspect of game play, despite sometime frustrating resulting player movement. Data explored in the next chapter supports foundational claims of contructionist computing and UMC advocates that greater user choice over the design process contributes to participant motivational and a feeling ownership of their projects (Peppler et al., 2009; Lytle, Cateté, et al., 2019)

Level design and prototyping

The use of a graphical grid structure to edit level design helped balance concerns of accessibility with the use of authentic code language.

Platform games often conform to certain patterns in terms of the elements involved in a level and their affordances. Common elements include the player that you control to movement of, platforms to be jumped on, enemies to be avoided and rewards to be collected. In the templated used in P1 the process of adding game elements was involved relatively complex process which involved changing parameters of functions to alter to adjust their location. An example of the code needed is included one of my tutorial chapters / appendix 4.x [1].

Complexities included: adding function parameters for screen coordinates which made it tricky for participants to intuitively place game elements where they wanted them; each element needs to be added separately; difficulties concerning adding graphical elements of different sizes; in addition for each element added different code elements needed to be added to three different functions in the code.

These complexities created barriers and frustrations into the game making activity at a stage which in P1 accumulated into a risk of participants giving up at worst or significantly reduced a participants experience of computational fluidity (as explored in the literature review) at best.

The complexities outlined above are addressed in GUI oriented tools like Unity through the use of a concept called tilemaps (Erhard-Olsson, 2018). This design choice allowed alignment with design principles for tools for novice coders, in particular the importance of visual approach to facilitate the programming multi-media projects for novices (Guzdial, 2004; Resnick et al., 2009). While such an approach was possible graphical based tools like Scratch, this was in conflict with my motivation to use authentic, text-based tools.

In P2, I revised the starting template to created game elements was based on a visual design in a

---

1   https://en.flossmanuals.net/phaser-game-making-in-glitch/_full/#create-a-game-space

grid format (see figure 4.x below) which replicated some of the use of use of tilemap technique within the text code format. Changes to the text based grid in the code area on the left would be immediately seen in the right hand project preview area.

Figure 4.x - Grid based editing of level design with a simple key for hazards, coins, and platforms.

This solution abstracted away complexity of asset placement to prioritise ease of use to help sustain participant engagement. Technically, this approach involves the construction of a data array for each level of 17 blocks which can be one of the following: x (platform); h (hazard); o (coin); or could be left blank (see Figure 4.x above). The end result was that while participants altered a text-based array, the grid structure had a strong visual correlation with the resulting game layout. This adaptation had a positive impact on engagement with level design in initial stages. Many participants spending significant time and effort undertaking many iterations of changes to the level design. In the vignette in appendix 4.x referenced above, Toby designed many levels making the game very challenging but still technically possible. Participants varied in their approach to level design, some drew on their experiences to mirrors platform game conventions, while others enjoyed working against these conventions, a theme which is explored in more detail in chapter five. DEVELOP THIS SOMEHOW.

Thus, here a technical approach helped resolve tension described above. Black boxing the process of adding assets via x,y coordinates and reducing repetitive process.

Design adaptations to facilitate novice use of a code playground

Code playgrounds, as described in chapter two, are an online environment used to test, share or invite help from online users. My choice to use a text code environment risked maximising barriers to participation by not profiting from design decisions in specialist coding software to help novice coders (previously explored in LR). These elements include steps to reducing syntax errors, shielding complexity, facilitating community commenting, sharing, remixing and other forms of collaboration. Learning computer coding presents challenges in part due to unfamiliarity with and potential complexity of code authoring tools and environments (Guzdial, 2004). To shed light on this conflict and to situate later findings, this section surfaces

observations from my journal notes and video data on how participants responded the tools introduced and my design adaptations.

DROPPED THE REMIXING / SHOWCASE - COMMENT / SUMMARISE?

THE IDEA BEING HOW CAN THE IDEAS OF CONSTRUCTIONIST TOOL KITS INTO THIS TOOL SET - PERHAPS PICK UP IN OBSERVATIONS SECTION. CHECK THE MOVED SECTION IN CHAPTER 6 NOW.

Remixing and community inspiration

The literature review outlined that one of the key principles of constructionist approach is promoting community inspiration and interaction. In the Scratch community this is achieved through and extensive online community which encourages remixing, collaboration and peer review via comments (Roque et al., 2016; Monroy-Hernandez and Hill, n.d.).

The shared community element on the other hand was quite different for glitch.com web playground tool due to its very mixed user base and range of different web based projects. In this learning design I took a decision to not promote this online community as a source of inspiration and code bases. Instead one code base was promoted. Sharing of ideas and inspiration did happen both on the playtesting of each others games and via incidental exposure to the online work of others. This happened as they shared log account details and thus when logging in to access their own work they would also see and have access to the code and graphical creations of others. I took advantage of this shared access in shared social and playful missions which are described later in this chapter.

Addressing code syntax errors

COMPRESS AND LINK TO ABOVE PARAGRAPH

Many software projects aimed at for novices use visual coding (block coding) approaches to reduce complexities of code use (Resnick et al., 2009; Bau et al., 2017). This process reduced possibilities for syntax errors (see glossary) and often provides a limited set of blocks to reduce complexity. While this approach not explored here for reasons previously outlined, there are some features of the glitch code playground which help. For example, the environment can detect the file type as javascript from the file extension and then uses a static analysis tool (linter) scan the code for signs of inconsistencies of code and syntax errors and highlight them (Tómasdóttir et al., 2017).

In glitch such code syntax errors are highlighted via a red dot provided a quick visual indication of where the error occurred. In addition if the user hovered over the dot would give an error message. In analysis of pair interactions, I frequently [how much from video data] observed the non-coding peer notice and point out the red dot, thus preventing further errors.

ADD IN HOW FREQUENTLY THE PARTICIPANTS DID USE THIS.

DESCRIPTION OF MY INTERVENTION. At times when addressing code snags experienced

by participants I would scan the code looking for the red dots and drawing attention to them.

If this was not helpful I would also use the console of the in-built developer tools of the browser (Collins and Barton, 2011).

Distributed vs. self-contained approaches to asset creation

A key element of game creation is the creation and management of graphical and audio assets. Many coding tools for novices provide a library of prebuilt assets and tools within the environment to alter or creation graphical and audio multimedia assets to facilitate and accelerate the creation of games and coding projects. While there are practical limits to the audio and graphical authoring capabilities of tools like Scratch (Payne and Ruthmann, 2019), such self-contained approached reduce possibility for compounding errors and complexity caused by the compatibility of file formats and migration and management of external asset files.

There has been extensive research supporting the motivational value of the ability for young people to bring their interests into multi-media creations via choice of assets and narratives (Peppler and Kafai, 2007; Resnick, 2014; Kajamaa et al., 2018). In line with these findings, I observed a palpable a sense of achievement when participants succeeded in seeing and hearing their creations in their game after making the final changes in code. For some, the sense of a achievement appeared magnified by difficulty caused by the unfamiliar environment and processes. EVIDENCE?

THE BENEFITS OF ASSET CREATION AS A WAY TO INCORPORATE FUNDS OF KNOWLEDGE - WHERE IS THIS EXPLORED.

In P1 I observed participant showing a high motivation to incorporate assets. Responding to the enthusiasm of participants in asset creation, I had introduced six additional software tools to participants including tools for pixel art, sound editing, sound creation. A table of the different tools is included as appendix 4.x

A summative timeline is included as a diagram.

Similar patterns of use emerged. Participants would identify the need for an asset in their game. They would then use the separate software to create that asset, and the be supported to save assets to their computer's hard drive in a compatible format. They would then need to upload assets to the code playground environment, discover the text link of the asset, and then insert that link into the main JavaScript game file at the relevant line of code. Undertaking the full process involved learning a complex chain of these individual actions. Some participants became remarkably adapt at this, thus transforming this chain of actions into a fluid operation. Thus, the distributed nature of the toolset used helped build authentic digital literacy skills.

Despite this benefits, I was concerned that in P1 the diversity of asset creation tools approaches was over complicated and risked distracting time and attention from other code-based actions involved in the activity of game making. Thus, in P2 I reduced the number of suggested assets creating tools.

14

This tension can be described as a play paradox. WHERE IS THIS FIRST INTRODUCED I was looking to avoid too much time spent in asset creation at the expense of other processes which would develop coding concepts and practices. It also increased the possibility of peer learning as less tools were being used.

ALSO NEEDED TO CHANGE DUE TO THE CHANGE IN GROUP WORK DESCRIBED LATER.

Use of Piskel

START WITH SHIFT FROM P1 DIVERSITY TO P2 FOCUSED & OVERCOMING PRACTICAL BARRIERS.

A pixel art approach to graphics allowed a balance between the potential for positive engagement with game making and the potential drain of time to this one activity, thus helping resolve one of the tensions emerging in P1. Piskel had proved to be intuitive for many younger participants with three main areas: a set of editing tools; a canvas for creation; and a set of tools to export, save and import work (see Figure 4.x).

Figure 4.x - Interface of Piskelapp tool

Participants spent a widely different length of time creating these graphics for a variety of reasons. Some took a long time to master the process of using the editing tool while others created images rapidly but would keep redesigning and recreating their game elements. The process of game art creation opportunities seeding narrative and artistic creativity is explored in more detail in chapter five.

–>

15

Observations on the game template and core tools

The previous sections have outlined changes to the core tool use including the use of a starter game template which I enacted to address the struggles participants had in becoming accustomed to game making. Despite the relative simplicity of the highlighted affordances outlined above, complex and divergent patterns of design behaviour began to emerge. Small code changes resulted in potentially large changes in game behaviour, appearance and difficulty aligns with a long standing concept of HCI research that feedback is motivating for system users (Malone, 1982; Bernhaupt et al., 2015). In addition, the use of revised in starter template in P2 allowed participants to maintain their games in a mainly working, shareable state allowing regular feedback from peers. The use of a starting template was inline with core motivations of UMC technique, namely, to build familiarity with the code structure, and confidence in use of tools in an accelerated way. As explored above, the shift in template design from P1 to P2 involved a much more careful and structured experience for participants in use and modify stages.

Despite these modifications, participants still needed help to access and use the relevant affordances in code template. In early trials with a limited number of families in late P1, I would point out the relevant part of the code and ask open questions e.g. '*I wonder what would happen if you changed that gravity variable?*' However, for larger groups this was not optimal, risking leaving some unsupported. My response was to turn to supporting documentation. In addition, participants soon began to ask how to make modifications to the game which needed additional code structures, a development signifying a pivot to the *create* phase on the UMC model. The next section explores similar tensions related to the role of supporting resources and documentation.

Participant conflict associated with project navigation and use of documentation

- Contradictions associated with project navigation and the use of supporting documentation - interpreted here through the lens of guided participation

PERHAPS INCLUDE TIMELINE OF DOCUMENTATION IN HERE

This section details a conflict where participants, frequently wanted to add new features or changes to their game but were not clear how to implement them. In P1 this caused delays and participant frustration, as although I could work with people directly to help them I had limited time with each participant or pair. The inability for participants progress due to lack of game and coding practices was compounded in early stages by little or no supporting documentation or supporting resources. This section outlines the evolutions and design choices informing the different forms of supporting documentation used to address this conflict.

The process of creating documentation which created enough support for users to be able complete coding tasks but also facilitated participants to choose their on path presented significant challenges.

MENTION HERE THE DISTINCTION BETWEEN PRIMARY AND SECONDARY USE OF

Quick Start Cards

To address the limit of one-to-one facilitator support to seed the starting processes outlined in the section above, I used printable resources which highlighted the key lines of code and how they can be altered to impact on game behaviour(see figure 4.x below). These *quick start* cards provoked a strongly scaffolded initial interaction with the code in a way strongly aligned with the use and modify stages of the UMC framework (Franklin et al., 2020).
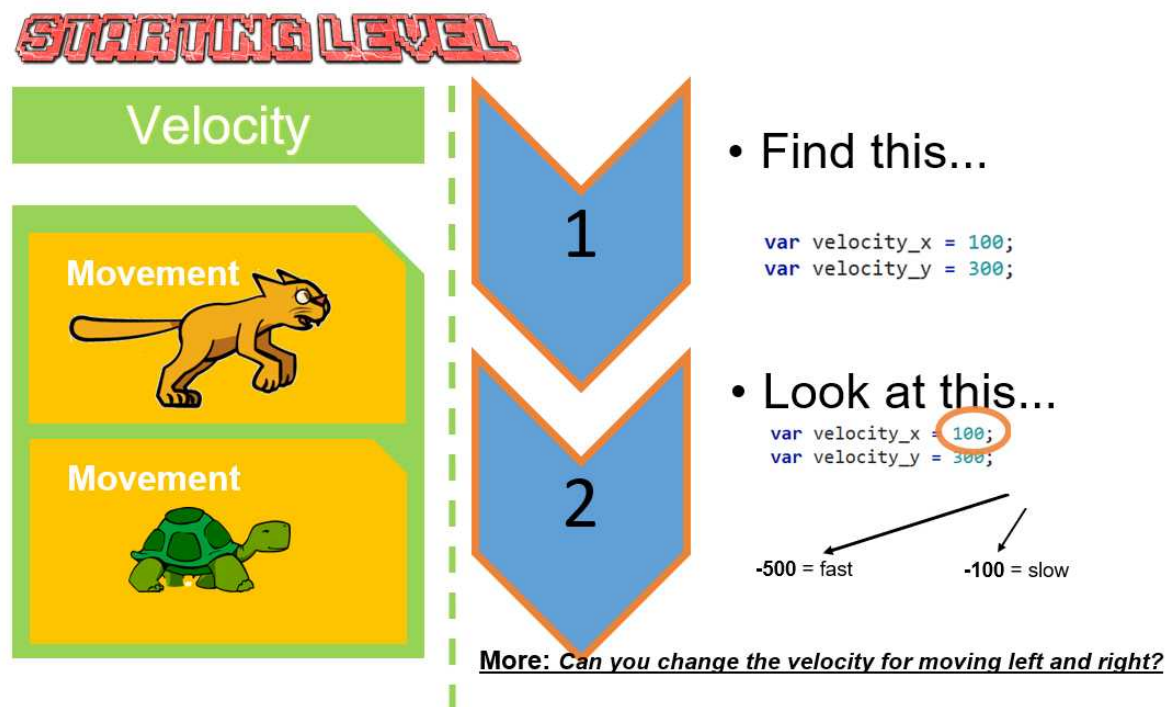
4.x

- Example of a Quick start card

The quick start cares were created at the end of P1 by trainee computing teachers in my faculty who volunteered to support a holiday outreach event for young people. I developed them by theming them into starting level, next level and boss level. Participants were encouraged to pick a card based on their interest and difficulty level.

Code snippet examples

The use of code snippets, while a promising authentic practice, presented initial challenges. To support a hands-on approach and responding the design choices of participants, I began by creating discreet code examples illustrating the requested elements. The use of code examples or snippets in code playgrounds is a common professional problem solving practice (Yang et al., 2017). These code examples allow users to see the behaviour in context with the code and output

side by side. While code examples existed on the Phaser website and support forums, in line with other support sites like stack exchange, their utility has limitations including lack of relevant, consistency, being removed from the domain context, and not being structured in a self explanatory way (Treude and Robillard, 2017). Thus, while I initially encouraged participants to search within these examples, and authentic documentation and help forums, the resulting confusion and difficulties experienced by participants, prompted me to create other more bespoke documentation.

Creating a bespoke set of code snippets helped address the challenges described above. In P1 I responded by creating one off documents with the relevant code which were both printed, emailed and shared via google drive. In line with the practice of accessing help via code snippets, the code examples could be to be copied and pasted into the game. This process lacked coherent process for participants to navigation to the resource they need. The process also lacked a consistency in signposting how the code listed fit within the existing structure. To help students see the code in the correct structure, I began to create code snippets within code playgrounds and distinct project. These projects shared the structure of starter game template and added only the code needed for the additional requested functionality. My rationale was that each pattern added builds familiarity with the code structure, a theory supported by observations of Toby's proficiency in the vignette above.

## Structuring instructional tutorial chapters

For each code example I created descriptive step-by-step documentation which in P2 were developed into chapters of an online book [2]. Each code snippet linked to a descriptive chapter. These chapters attempted to balance an informal, hands-on approach over pre-teaching concepts, with the request from some from parents in the feedback to P1 (see appendix 4.x) to provide background concepts and explanations of coding constructs. To address this tension, in addition to self-contained chapter focused on instrumental code changes needed to implement game features, I created opening chapters of the online manual which were more traditional in format and explained underlying concepts that the starting template had initially abstracted away from the participants.

Theming the a collection of game design patterns to aid navigation

Introducing documentation created additional tensions in the learning activity, specifically, I observed users failing or struggling to find the right resources online. This prompted me to design and introduce a themed hub for both snippets and tutorial chapters. My aim was to mitigate potential learner alienation from unfamiliar technical documentation through accessible design and to relating documentation to participants existing gameplay experience rather than underlying coding constructs. While time consuming, the process of aligning documentation, code snippets with more general concepts of game analysis, served to simplify the navigation of documentation. The guiding principle is that key affordances of the supporting secondary stimuli are designed to closely align with the objectives of leading activity at the predominant scope of

2   https://en.flossmanuals.net/phaser-game-making-in-glitch/_full/

activity.
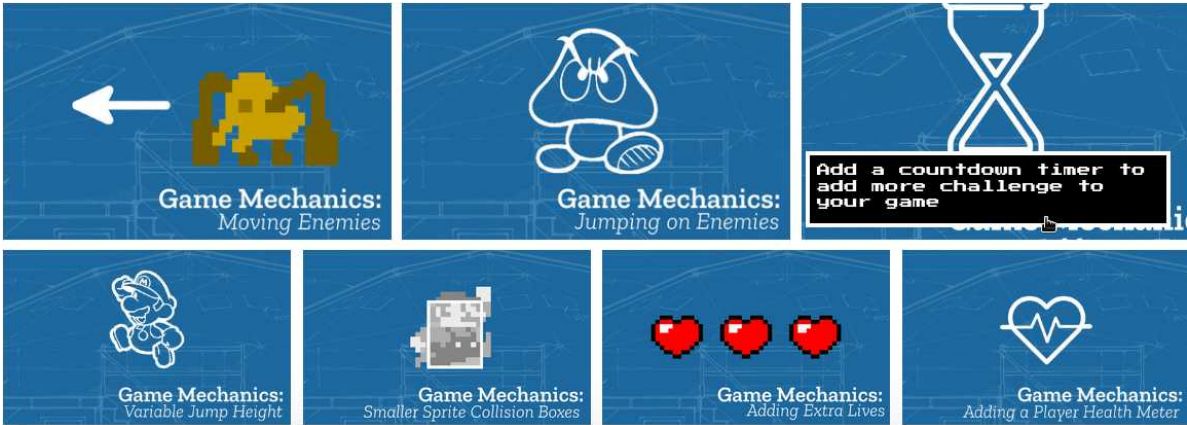
Pattern Collection

INTRO SENTENCE

The process of theming the patterns has the potential to develop the community knowledge of game making concepts (Holopainen et al., 2007). In grouping the game design patterns into categories for the documentation hub page, I drew on academic and professional interpretations of game elements (Tekinbaş and Zimmerman, 2003; Salen et al., 2006; Schell, 2008). Schnell's detailed analysis of tens of game elements presented as design lenses was too complex for this audience. Instead, I adapted a simplified introductory framework developed for use in youth-oriented Game Jams to help novice game makers hack/analysis and then adapt key elements of non-digital games (Cornish et al., 2018).

- **SPACE:** Where the game takes place.
- **GOAL:** What is the objective of the game? What are you trying to do?
- **COMPONENTS:** What are all the objects or actors in the game?
- **MECHANICS:** What actions take place in the game. What are the verbs involved?
- **RULES:** What can or can't you do in the game? What defines boundaries? Does play happen in real time or do you take turns?

The framework youth game jams and in the Q2L school to help participants develop their implicit knowledge of game design concepts in to explicit share vocabulary before engaging in digital making via collaborative analysis of common games (Cornish et al., 2018; Institute of Play, n.d.). Similarly, in early stages of my design participants completed a similar activity after playing retro arcade games [included in appendix].

I related this simple categorisation the emerging list of requests for game features made by my participants.

Game Mechanics	Game Polish	Game Space	Challenge Systems
Jumping on Enemies	polish.	New levels	New levels

This categorisation, while simplified, is consistent with professional and technical frameworks

19

popular in game making communities including: the MDA framework (Olsson et al., 2014) (which focuses on analysis of games based on the user experience), Elemental Tetrad (Schell, 2008), and DDE (Walk et al., 2017). The theme of using technical frameworks in an accessible way to facilitate the creations of novice participants is continued in the chapters five and six.

Responding to tensions and barriers in cultural aspects of the game making activity

Conflicts due to identity clashes and dysfunctional group work

MAKE MORE OF THE P1 -> P2 SHIFT AT START

Narrative summary of conflict:

This experience aligns with research in this area explored in the literature review which explore pedagogies to address barriers concerning identity and technology-driven practices (Kafai and Burke, 2015). These contradictions involved accumulating tensions which resulted in families feeling anxiety and alienation from the group coding environment and associated peer working dynamics. My journal notes see an evolution of attempts to try to build into the program, activities which help build the participants sense of their own identities of game makers or more generally digital designers. These are detailed later.

I illustrate this conflict using the experience of one family who withdrew from P1 three weeks before the end of the programme. While members of this family were able to engage in planing via sketching on paper and in creating pixel art, in later stages they were reliant on others to implement code changes. The freedom of choice and imagination allowed by designing on paper and via pixel art created compounding tensions. One crisis point involved a frustration of a child who was not able to implement a feature they wanted to add to the game next. A 3d bee design roaming a 3D landscape. I encouraged personal expression for the purposes of engagement and motivation. However, here it can lead to conflict and frustration as the desired changes of the child were not realistic to the technical scope of the project. The previous section explored the use of two dimensional (2D) platformer game as a technical template in P2. I also addressed participant expectations of an acceptable project scope for our timeframe as a non technical process.

When the family withdrew, they shared in feedback (see appendix 4.x) the moment that decided their withdrawal. that at one point they "looked around and just saw people doing hardcore coding and no longer felt that they belonged". In the end stages of the game production process, due to the dynamic of the larger group, they were reliant on others to implement code changes for their imagined game, unable to contribute fully at this point and found themselves isolated. Thus a contributing factor to this families alienation were tensions engendered by the large group size and compounded by frustrations stemming from unfamiliarity with tools and processes. In participant feedback, the parent of this family described in the previous section indicated that it took too long before in the planning stage and called for more hands on play and use of the tools of production before being called on to make creative decisions. The parent likened this to an arts

studio approach. This feedback contributed to choices outlined in other sections of this chapter. (WHICH ONES)

DROP - COMPRESS - CONCLUSION? The incident lead me to greater reflection on ongoing measures needed to prevent participant drop out for due to cultural tensions and negative affect to the working community. These include:

- An awareness of the danger that those more confident in coding create more involved code problems that need more facilitator time, potentially making others feel less valued.
- A concern for the fragility of learners positive affect towards the group game making process and thus the need for initial playful starting and closing activities to be continued beyond initial sessions.
- The value of a stronger buy-in by participants, ideally a greater commitment to the collective making process balanced with the need for low pressure (avoiding a negative sense of obligation).

REMOVED

COMPLETELY REMOVED SECTION ON PHYSICAL MAPS

On a drama processes

The introduction of a drama process was introduced in response to a the barriers to participation caused by not identifying with the culture of coding or gaming.

A game encouraging maker / player types

The drama process work in P3 to develop ways to encourage diverse and playful approaches to making had its roots in a game introduced in P2 where participants took part in an interactive the Bartle test. The Bartle test draws on work which …

The full process and rationale and initial reactions are detailed in appendix 4.x . In summary,

The process was very engaging, parents responded that the process gave them new insight into the digital identities of their children. As part of the process I then proposed that there different game maker types and asked participants to evaluate and discuss with peers what kind of game maker they were. From the following list.

- **Social makers:** form relationships with other game makers and players by finding out more about their work and telling stories in their game -

- **Planners:** like to study to get a full knowledge of the tools and what is possible before they build up their game step-by-step
- **Magpie makers:** like trying out lots of different things and happy to borrow code, images and sound from anywhere for quick results
- **Glitchers:** mess around with the code trying to see if they can break it interesting ways and cause a bit of havoc for other users

DATA REQUIRED FROM VIDEO!?

This self evaluation helped confirm my own observations of different motivations driving different participation behaviours. Fuller discussion follows in chapter six.

The success of activity spurred the creation of the creation of side-missions to be carried out in from of a dramatic process.

AND - link more

Description drama process using extracts from vignettes and concepts of MoE

The drama process involved a fictional frame of aliens coming to judge the earth. An extract of how I introduce the drama frame, the main and side missions is included as appendix 4.x. Below is a short extract of the overall mission.

```
We have some guidelines:

  • Make a game about a big or small problem for your planet to solve. If you
can let us play it each week as you go along.
  • Give us an update each week by recording a group update.
  • Show you can work on your own but also work as part of a team.
  • We will also send you text messages with some mini-missions sometimes. Be
sure to tell us how you do.

Please now get started and come up with a new game about solving a problem on
Earth.
```

Extract from Appendix 4.x: Vignette of introduction of the drama process

The process of introducing a scenario for participant to respond to is common in project based learning in this vignette. Here it is extended using a dramatic element in line with Heathcote on Mantle of the Expert (MoE) and other process drama techniques (explored in Lit Review). The following section outlines key characteristics MoE (Aitken, 2013) in context.

Contracting into drama

As a facilitator, I indicate that we are entering a dramatic process and attempt to draw everyone into a commitment to the process. The collusion is noted by both parents and children in recorded exchanges. For example, after engaging with a code example provided by myself in the role of the aliens, one pair make the following comment.

> Parent: Look, this is what Mick's showed, what Mick's has done.

> Child: The Weean have done!

> Parent: The Weean, sorry yes.

Facilitator activities in role

My role is a link between the participant and the fictional commissioners of the games. The

transcript in appendix 4.1.a details an example of extended facilitator input as instruction. As a practitioner in this setting, I had tried to limit teacher instruction and to outline boundaries to prevent learner disengagement. However some input was needed to help address the previously explored tension choice with over-ambition, and signpost participants to needed documentation. The drama narrative helped resolve some of this personal tension by allowing me using the dramatic commission as a foil to help avoid the encounter feeling personally controlling [heathcote_drama_1985]. CLARIFY - EXTERNALISING LIMITS TO ACTIVITY

Learner activities in role

Beyond the wider fictional frame of the making activity, learners were also sometimes asked to undertake some activities in role, in particular reflection in role. This and other example are explored via vignette 4.1.b.

Vignette 4.1.b illustrates this process. In the extract participants are invited to take turns showing their game, recapping their progress and outlining next steps to the alien observers. I play the role as a liaison facilitating the process for the fictional, remote audience. I also draw attention to the secret missions that had been distributed to participants, which along with social missions are explored in the following section.

Use of side missions to encourage varied creative practices

One of the activities that participants carried out within the frame of the drama process was the undertaking of side-missions. These missions encouraged community-focused patterns of behaviour which I had observed in previous iterations of game making and via the game on playing and making types described above. These were activities not directly related to game making process. They were designed to encourage the exploration of the games of others, to encourage a playful approach. I had identified these behaviours as potentially helpful in maintaining the positive affect and identification with the on-going group process of game making. An extract of the table of both social and secret missions follows (a full table is available as Appendix 4.x). These mission were printed out on cards and one of each type was given to the participants in the first half of the first two sessions.

Your Alien Mission (social)	Your Secret Alien Mission:
Find out the names of 3 games that are being made.	Change the variables at the start of someone else's game to make it play in a funny way.
Make a list of characters in two other games being made.	Change of the images in someone else's project to a totally different image and see if they notice.
Find out the favourite computer games of 4 people.	Change the level design of the first level of someone else's project to make it impossible but try to change as little as possible to do that.

These activities were undertaken by most of the participants. In analysis of one key sessions involving missions from which the transcript was taken. Out of 10 participants, 8 visibly engaged

with the secret missions during the session.

The process was not without friction. While the some of the secret missions encouraged forms of disruptive play, griefing of playing against the game (Bakioglu, 2008; Bartle, n.d.). While this increases engagement for some learners, the process also suffered the danger that participants transgress levels of cheekiness and play which explores boundaries, to more disruptive ones. In this context this involved frustration and wasted time for other users. After initially engaging with the process of secret games, Toby and Dave, who were later working on a tricky coding process, expressed frustration at interference.

Thus, while promising, limits to this process should be evaluated to avoid overly disruptive behaviour which create barriers to progress. For example, time limits could be in place.

```
<!-- Mark: Right we've got a background in. Do you. Do you want to reply to
the Weean.
Ed: Yes. (Ed starts to type very slowly)
Mark: (after some time) While you do that I'm going to go do my mission.
Ed: What's your mission?
Mark: To find out about other people's favourite games.
Ed: Alright. -->
```

This chapter makes limited claims in terms of the effectiveness of this process due to limits in terms of data gathering. However, it can make a claim that this activity has significant potential. HOW EXACTLY?

In summary, this activity helped addresses one of the key barriers to taking part in a coding community that of alienation from the culture of coding. The use of missions introduced by facilitator are taken up voluntarily by participants and aimed to address conflicts caused by identity clashes by recognising and encouraging diverse making approaches and styles of community participation.

The positive reaction of young people to the drama process was also picked up in interview data.

```
Mark: Tell us about the Weean?
Ed: er.
Mark: What did you like about the Weean?
Ed: Just very silly and I don't know. They answered your questions.
Mark: Was it, because there was a sense of play, a bit of fun and a bit of
anarchy.
Ed: Yeah.
Mark: What else can you tell us about the Weean?
Ed: er.
Mark: There was the little sabotage things you had to do as well.
Ed: Yeah
Mark: So you liked those games where you had to kind of hack somebody else?
Ed: Yeah
...
Mark: The Weean thing was, I think, a massive hit for everybody because it
just brought sense of play and fun. And it took attention away from it being
overtly technical. Brought a personal element in and obviously that anarchy.
Kids love the absurd.
```

Observations on written interactions with the alien in the drama

As previously explored, an emerging area of tensions that I noted in journal notes and in feedback from participants was the area of supporting documentation and resources. In addition to the careful technical and pedagogical creation and organisation of resources undertaken in P2, I sought a way for participants to be aware of and access relevant documentation without interrupting their creative flow or the fun of the experiences.

In P1 I created a mailing list for participants to ask technical and code questions of the wider group. My aim was to create a peer community helping each other rather than a helping relationship only between myself and individuals. However, this email list was hardly used for this purpose. Two parents shared that they were reluctant to bother other participants with their own issues. To address this in P3, as part of the drama process, I encouraged participants to write to the aliens to ask help. This is described and illustrated in appendix 4.x.

As well as exploring documentation and accessing technical help within the drama frame, student also engaged in playful dialogue with the aliens unrelated to game making. The process of asking the aliens for technical help within a code project sparked a playful process of informal chatting with the aliens. This chat began to fulfil a function of building insider rapport, creating a fun atmosphere, celebrating the completion of games in the absence of a public showcase, and signposting the achievements of other participants.

The process started with supportive and celebratory messages posted from the alien. The impact was significant with the young people with 5 out of 7 engaging by writing messages and all mentioning the interactions verbally during the session.

Concluding remarks on process drama

While inspired by the work on Heathcote on Mantle of the Expert, not all element of the process align directly. For example, the drama does not try to position the participants as expert game makers in a professional context.

This section has explored the use of the possibilities of a drama process to support both the leaner expression of agency and as a pedagogical process for facilitators to militate probable conflicts and obstacles for learners in the creative process. The mission-based interventions began as naturally occurring expression of learner agency which became incorporated into the drama process.

The impact of time pressure to finish games also compounded tensions involving shifting away from game making to reflective activities. Even with the use of reflection in role, reflection activity was not prioritised in final sessions as time drew short.

This aligns with existing research addressing the barriers to effective reflection and meta-cognition on PBL processes. NOTE - EXPLORE THIS IN LR AND DISCUSSION IF KEEPING THIS.

Chapter Discussion

NOTE - MOVED A SECTION ON THE EMERGENT DESIGN TO METHODOLOGY

In summary, this chapter has explored the complexity of the interacting tools and documentation in relation to their authenticity and impact on participant experiences. While the use of authentic tools and processes, while challenging for novices, can be facilitated by careful alignment of key design principles. However, some barriers involving social and cultural factors suited interventions of other pedagogies. In later chapters I explore the interaction between authenticity and agency in more depth using the emerging pedagogies as an analytical lens.

Summary of barriers and tensions explored in this chapter

The following summary table lists barriers and tensions drawn from the literature review that were significant when reviewing my journal observations, feedback from participants and video data. The purpose of this summary is to help recap and ground the reader before exploring the experiences of participants in more detail in chapters five and six. It focuses on use of tools but also signposts some social and cultural elements.

Barrier / Tension	Description	Related Intervention
Alienation from culture of computer coding	As outlined in literature review.	Use of games, play testing and other social approaches. Explored in chapter six
Unfamiliarity with group project management	Tensions arising through group miscommunication in P1	Avoiding large groups from P2 onwards, and starting with a half-baked game template which individuals or pair used as a base.
Unfamiliarity with language and syntax	Example: Javascript use of punctuation especially in arrays and in the structuring of functions.	Use of UMC pedagogy and quick start examples to build familiarity
Unfamiliarity with coding concepts	Example: the use of if statements and global variables	A starting template with much of the needed code in place, facilitator explanation when concepts are met in-situ, supporting documentation to reinforce previous experimentation
Lack of direct knowledge of game making conventions and experience of game analysis	e.g. common game programming patterns	Games and activities to access and surface tacit knowledge, use of a structured GDP collection to reinforce concepts
Tension between pre-choosing game genre and allowing for flexible choice of	Example of the family wanting to make a 3D exploration game, but my in ability to support that	Introduction of the templated 2D platformer in a half-baked stage to create an engaging first impression, and ability to quickly personalise the

Barrier / Tension	Description	Related Intervention
participants		game to increase participants' feelings of ownership over it
Tension between teaching the programming of game states and the complexity it introduced	e.g. play state, settings page, welcome screen, game over screen	Introducing one state in the game code thus allowing extension, but not drawing attention to it
Tension between engagement value of providing a wide range distributed we production tools and complexity and disunity it induced	e.g. move from diverse tools used in P1 to one raphical editor and one audio tool in P2	Restriction prioritised to encourage group work, as engagement was already high
Tension between the use of professional coding tools which increase authenticity at the prices of greater learning curve in adoption	While suitable open source tools exist, they need careful curation by facilitators to reduce cognitive load	Design of template to surface key affordances to mitigate barriers to use and start usage patterns.
Tension between producing extensive and varied documentation to support the different design choices of users and the increased complexity of navigation of the learning experience this brings	The process of finding and following documentation was patchy	Resources were provided in a menu form linking to both code snippets and instructions

Link to next chapter

This chapter has dealt with the evolution of design in initial phases and responses to learner experience to resolve tensions. A key focus of this chapter has been barriers to computer coding and use of particular software. I have explored alignment with the results of extensive research from the constructionist school. While much the focus of much constructionist research is on the design of toolsets to facilitate to personal knowledge building and expression via open project work, more recent work from researchers in this school has started to embrace the value of situated, community driven production as a lens (Kafai et al., 2020). One of my central proposals regarding the methodology of this thesis is that while under-explored in this area, there is great

potential in collaborative design based approaches to uncover situated and emergent practices in a way that can help seed community activity by other facilitators.

The following chapters have a focus on this gap. The next chapter explores a potentially fruitful aspect to help guidance for educational practitioners. The use of a design pattern approach. In this study the patterns take the form of working with common game elements game design patterns. In the language of the theoretical framework, I explore the process of working with GPDs as a germ cell of the overall game making activity. Chapter six explores the delivery and start to explore the impact of the process drama used and emerging community processes on cultural dimensions of the game making activity.

Gilbert is the worsted name.