Findings Chapter 2 - It’s like a project in itself really

* [Introduction to This Writing](#introduction-to-this-writing)
* [Chapter introduction](#chapter-introduction)
* [Study of Vignette A](#study-of-vignette-a)
  + [Descriptive Commentary on Vignette A](#descriptive-commentary-on-vignette-a)
* [Cultural, Interpersonal & Personal Planes Analysis of use of GDPs](#cultural-interpersonal-personal-planes-analysis-of-use-of-gdps)
* [The use of GDPs explored from a cultural focus](#the-use-of-gdps-explored-from-a-cultural-focus)
  + [Developing a language for informal feedback for peers to influence modification of games](#developing-a-language-for-informal-feedback-for-peers-to-influence-modification-of-games)
  + [The propagation of use of GDPs and associated practices stemming from playtesting](#the-propagation-of-use-of-gdps-and-associated-practices-stemming-from-playtesting)
  + [GDP used to allow exploration of home and professional funds of knowledge and practices](#gdp-used-to-allow-exploration-of-home-and-professional-funds-of-knowledge-and-practices)
  + [GDPs and the role of documentation and supporting resources](#gdps-and-the-role-of-documentation-and-supporting-resources)
* [The use of GDPs examined from an interpersonal focus](#the-use-of-gdps-examined-from-an-interpersonal-focus)
  + [GDPs used to facilitate prioritisation](#gdps-used-to-facilitate-prioritisation)
  + [GDPs aiding the process of division of labour](#gdps-aiding-the-process-of-division-of-labour)
  + [GDPs used to scaffold ideation processes](#gdps-used-to-scaffold-ideation-processes)
  + [GDPs facilitating designing for others](#gdps-facilitating-designing-for-others)
  + [GDP driving adoption of emerging technical processes](#gdp-driving-adoption-of-emerging-technical-processes)
* [The use of GDPs explored from a personal focus](#the-use-of-gdps-explored-from-a-personal-focus)
  + [Use of GDPs to support debugging and the product revision process](#use-of-gdps-to-support-debugging-and-the-product-revision-process)
  + [GDPs as a way to surface and discuss embedded computational, design and systems concepts](#gdps-as-a-way-to-surface-and-discuss-embedded-computational-design-and-systems-concepts)
* [Discussion on the use of game design patterns in the 3M learning design](#discussion-on-the-use-of-game-design-patterns-in-the-3m-learning-design)
  + [Discussion on the use of design patterns and pattern catalogues](#discussion-on-the-use-of-design-patterns-and-pattern-catalogues)
  + [Game design patterns as a intermediate-level construct to facilitate developing coding fluency](#game-design-patterns-as-a-intermediate-level-construct-to-facilitate-developing-coding-fluency)
  + [Emerging concepts of learner agency in the design](#emerging-concepts-of-learner-agency-in-the-design)

## Introduction to This Writing

While this piece of writing contains results, it also contains references to concepts and methodologies which will be included in other chapters in the final thesis. Other emerging chapters are online at - https://mickfuzz.github.io/phd4/thesis\_chapters.html

To help the reader understand the data in this chapter the following notes on indicative content of other chapters are included.

* Introduction: Research landscape to develop coding and design practices in line with social understandings of learning. In particular, the intentional development of a communities of designers and coders with an educational aims has been explored paying particular attention strategies that educators can use to aid these processes.
* The study uses mixed methods in methodology. Informed by analysis of journal diary, notes, video data and interviews (Methodology)
* The research design evolved as the game making program was underway based on feedback and reactions of the participants (supported precedent in design-based research) (Methodology)
* A long pilot course and development stage resulted in extensive support materials and activities which I have structured into a learning design called 3M (Methodology + see recent chapter on 3M model)
* Game making is based on remixing code from an existing incomplete game template (half-baked games & scratch community processes) (Literature review and Design)
* A narrow data samples has been used for this study. This prioritises a greater depth of analysis on interactions over generalisation. This is justified in a section on data validity in methodology chapter.
* Results chapters use Rogoff’s 3 planes of analysis (Cultural, Interpersonal and Personal) which aligns well with AT (Methodology)
* The concepts of resolving tensions within activity from Activity Theory are used to explore the evolution of the design of activities and resources. (Design results)
* The working definition of game design patterns (GDPs) is explored in the literature review and enhanced in the design chapter is augmented with analysis of specific examples of their use by participants in practice.

## Chapter introduction

The purpose of this chapter is to explore how the data gathered informs thinking on two research questions of this study.

* How can game design patterns support the development of coding practices by novices?
* **New question** - How can learners build agency in an evolving community of game makers?
* How can the experience of fostering community focused digital game making practices inform wider practice in this area? (begun here but mostly in next / discussion chapter)

In the last chapter we explored inclusive pedagogical methods that helped participants become part of an emerging community of game makers. This chapter analyses a sub-activity system to the creation of the whole game, namely the implementation of individual game design patterns (GDPs). In the following sections, I draw on such detailed observations of participants in practice to outline how GDPs are used by participants. This chapter begins with a Vignette of parent child interaction in game making process. The vignette is used to introduce an overview of interactions of that pair generated from analysis of video data. The chapter then focuses on the use of game design patterns by participants in these interactions and then in reference to cultural, interpersonal and personal planes of activity. Finally a discussion section explores implications and observations from these findings in relation to existing research and concepts.

## Study of Vignette A

In this section, to give an overview of the specifics of the learning design, and participant interactions I use vignette to give a snapshot of the interactions of one family. This comprises a table of participant dialogue, screenshots and description of their interactions including gestures followed by a summary interpretation of interactions.

*Context:* For the previous five minutes the parent and child dyad have faced a blockage of a glitch in the software which prevents their wider objective of adding game design pattern of adding keys and doors to their game. In short, in this pattern the users must collect a key and then navigate to a door to progress to the next level. The parent Sh has been trying many different things and changing things in the code while the child Th has been expressing boredom. In response, Sh has expressed frustration and irritation. Finally Sh suggests that Th asks the facilitator Mick for help to resolve the coding blockage.

|  |  |
| --- | --- |
| **Dialogue** | **Activity** |
| S: You can ask him for help again if you want. | S is using laptop. T is nearby |
| T: Why |  |
| S: Cos I can’t do it. |  |
| T: Ok |  |
| Exchange follows with Mick and S - Problems solving using debugging methods and exchange of information about sequencing of variables. Ends after several minutes with game exhibiting desired behaviour of moving to next level after collecting a key.  The exchange contains good troubleshooting of sequencing  And uses the debugger as a tool.  A fuller transcript is available.  One error is due to different versions having different state names. So documentation is out of date. | |
| M: Great | (On Screen) Platform game bug is resolved shown by game progressing to the next level. |
| S: Thank you.  Did you see T? | (Off Screen) M cedes computer to S. |
| T: Did it do it? |  |
| S: Yes |  |
| S: I’ll reload it. Here you go | S presses refresh key to reload the game |
| T: Laughs | T moves laptop to point towards her and takes over using the keyboard |
|  | |
| T: (Makes excited cat noise) | T is using arrow keys to control game and check desired behaviour of player progressing to level 2 is working. T progresses to level 2 where there is not a door or key. |
|  | |
| T: We need a door in this one.  I want to make more levels now.  So you can put lots of doors and lots of keys  (in a rhythmic voice)  Key Door Key Door Key Door (in a sing song voice)  Key Door Level. Key Door Level, Key Door Level | S moves computer back to take over the use of they keyboard. She changes tab and navigates to the section of the code which deals with the design of levels and placement of different game components. |
|  | |
| S. Well. Where are you going to put your door in level two?  Do you want to change the layout at all? | S gestures to screen |
| T: What’s door? | S doesn’t move laptop but gets closer to take over keyboard |
| S: d |  |
| T: Ok |  |
| T: d . Door right there. | Inserts a “d” in an array high above a platform |
|  | |
| S: And where’s the key going?  Must be quite hard through get to that door  Key is K |  |
| T : (Calmly) Ok just let me do something | T starts to delete multiple characters on a line |
| S: (Alarmed) What are you doing? |  |
| T: Back. You’ll see!  T: Ah yes |  |
|  | |
| S: What are you.. Ah you’re making a platform. I see. | T starts to add x’s to replace the spaces she has deleted |
|  | |
| T: See! |  |
| S: So where is the key going? |  |
| T: Key, so it’s K? |  |
| S: Yes |  |
| T: Where’s K gone |  |
| S: There | S indicates where the K key is on the keyboard by pointing |
| T: Back. K …. for Key |  |
| S: Are you putting one in your third level as well? |  |
| T: Yes and then I’m going to make more levels | T Smiles at S and then looks away for a short while |
| T: You do this bit |  |
| T: And then delete those H’s. And in there. |  |
| S: No. No. |  |
| T: Then, let me do it. | (off screen) T takes over the keyboard |
| S: It’s no fun having a game with any hazards to avoid. |  |
| T: Is for me!  How do you go that way back?  So sorry for deleting… They just… | (on screen) T deletes hazards in Level 3 of game |
| Back Back Back  O O O.  (laughs)  (replaces another) Back O (giggles) | (on screen) T adds three coins to Level 3 by inserting o in the matrix |
| S: Goodness me what what the point of designing our car with fumes if we’re not going to use it? |  |
| T: I don’t know (sighs)  T: You put the key and door in this one.  T: Go on then. Key - Door - Person. |  |
| S: Person? |  |
| T: Key Door Person. | (off screen) T indicates that her mother is the person she is referring to  She add a platform and a key and door to the design of the third level |
| T: (laughs) | Sh starts play through of their game again. She progresses past the first level, completes second level. The game then goes back to the first level unexpectedly. |
| S: Is this the first level again |  |
| T: I think so |  |
| S: OK so our the door for the second level goes back to first level and we want it to go to the third level presumably |  |
| T: Yes (laughs) Yes. Why could that be? That’s silly |  |
| T: I’m just going to go see something | S dives into the code to try to troubleshoot the problem. T watches for a short while becomes disengaged. She then leaves mother to explore the wider environment of the room and objects within it. |

### Descriptive Commentary on Vignette A

The parent has taken on the role of solving harder code problems following instructions carefully. The child is one of the younger participants when blockages occur she explores the room or to dwell on the periphery of other participants interactions. The child wants to add additional levels to the game. It is likely that this stems from observations of other games being made and conversations overheard. As this pattern has been added to other games and has become a popular topic of discourse between participants.

For both parent and child there is an explicit awareness of game making patterns, the cultural conventions of a platformer game and concept of the game as a dynamic system. This awareness drives their work on the design and coding mechanics of the game. The target game design pattern Keys and Doors is repeated in a sing song voice by the child along with chivvying .

Understanding of the game as a dynamic system is seen clearly in the parent’s alarm at the child’s deletion of all elements of hazard. This interaction shows the parent guiding the child to imagine the user experience through the use of terms from GDPs. The parent uses GDPs as a prompt that they are designing for others. The child has also thought ahead to this time of playtesting. Both child and parent make reference to the imagined player experiences. In the vignette above Sh mentions “it’s no fun if their are no hazards”. The response of Th “It is for me.” indicates a desire to play against game conventions with a desire to confuse or disorient players.

Game making patterns are outlined by both the child and parent. They talk of adding more levels, of the mechanic of keys and doors in this vignette. After a blockage caused by debugging coding is removed, the parent immediately re-engages the child. The parent says “Did you see Th?” and then prompts the child to play-test the game. The child is keen to both replicate the GDP of keys and doors in the second level and to add a new GDP of adding additional levels to the game.

Cracking open a this seam allows for new possibilities. The process of working on a new GDP seems to motivate and sustain the child’s activity. The child shows high engagement at this stage outlining her plan for adding pattern to make progress in the process of designing their game. The transformation from the period of time when her parent was problem solving code is dramatic. For this child this alternation between higher and lower engagement with the coding processes happening on screen was fairly consistent. The implementation of a new GDP often involves adding new code to the game or a significant change in code which may need debugging. For this pair, larger coding activities were beyond the ability of the child. The transition from adding one GDP involves playtesting. In this example, the process of playtesting also involves the child speaking and identification of modifications to and new GDPs to implement.

The adult is building proficiency in coding practice. This is shown in a variety of ways including; finding and comprehending supporting documentation, fluidly navigating between undertaking more advanced coding to implement new GDPs, playtesting and preparing the coding environment for more basic coding of her child.

This chapter now expands to represent the diversity of the coding and community practices of participants through focusing on their use of game design patterns.

## Cultural, Interpersonal & Personal Planes Analysis of use of GDPs

In the preceding section, a short interchange between participants was analysed using three foci of activity cultural, interpersonal and personal foci. In this section I present the results of analysis of other video recordings as separate observations on the way GPDs are used by participants and facilitators to further their game making objectives. Acknowledging that these observation do not fall nearly into each of the three planes, I begin with those oriented more to cultural focus before progressing to observations oriented chiefly to interpersonal activity and then ending with some observation on GDPs impact on personal knowledge construction.

## The use of GDPs explored from a cultural focus

In the previous chapter the role of cultural activities to engage and sustain engagement in this game making programme were examined. This chapter allows us to focus in on the way design patterns used are by parents, children and facilitators in in this emerging game making community.

### Developing a language for informal feedback for peers to influence modification of games

**The concepts of game challenge and game feel evolved through informal feedback during playtesting and served to influence peers to modify their games to increase the enjoyment of peer players.**

The discussion of game challenge, specifically comment about how ‘hard’ participant games were as the most common interaction during playtesting. The concept of difficulty for most of the participant’s games was dependent on the interaction between the feel of the game controls and elements of game challenge associated with placement of hazards and moving enemies. The term *game feel* has varied interpretation but is generally framed as the responsiveness and feeling of control over the main character during the core movement of the game. In this case, it effects the ability of players to move between platforms and avoid enemies. In this design the jump mechanic is determined by the use of variables controlling gravity, jump velocity and movement velocity.

The regular playtesting of games allowed participants to give each other feedback regularly and game feel was one of the aspects that young people in particular to gave frequent feedback on.

<!-- PERHAPS INSERT HOW - SCREEN SHOT? -->

Parent Mi had been focused mostly on completing asset design. The only changes she had made to the deliberately frustrating initial player movement (discussed in design chapter) was to change player jump velocity. Player jump (y) velocity was set very high but left right (x) velocity was slow. This created a very frustrating game feel.

As a facilitator, I shared feedback about the frustratingly slow movement time and an indirect feedback on the high velocity jump value. I use quite indirect language when giving feedback and while I reference Mi’s frustration rather than giving direct feedback.

Ch1: That looks nice (referring to the graphical look of the game)

Mi invites Ch to play as she can't progress due to the difficult game controls.

Ch1: It jumps super high but so slow

Pause

Mi: He has to go slow be cause he's an astronaut, you see.

Ch1: It's hard.

Ch1 leaves.

Mi: (to peer parent with proud tone.) It's hard. Wow.

Fi comes to play the game.

Fi: How much jump speed to you have?

Fi: Your jump speed is massive.

We can see that Mi justifies the game feel of a very high fast jump with a narrative response about the character being spaceman. However, the limited amount of time anyone plays her game and her own frustration in playing it is telling. The game feel is frustrating in the wrong way here. Mi seems to initially misconstrue the feedback she is getting here equating her ability to make the game hard as a positive thing. However, towards the end she notes the frustrating nature of the game.

Other children come and play the game but only for less than a minute before leaving. While their feedback is non verbal the very short length of time that some of them spend is noticeable. After the last one leaves Mi comments “It’s so frustrating.” In these interactions we can see a consistent message coming from peers in the playtesting process. They praise the look of the game but offer constructive feedback to help improve the GDP of jumping.

While the players do not tell Mi directly to change the game, these comments appear to direct direction of the design to comply with an emerging community norm of jump feel stemming from the personal experiences of the participants and from tangible feeling of lack of control over the player’s character in the game. The same message is delivered in a variety of ways, above we can see feedback from Mick trying to bridge a technical and conversational approach, direct feedback of the personal challenge level and an interpretation of the cause from Ch and then a more specifically technical explanation involving the naming of the variable *jump speed*.

### The propagation of use of GDPs and associated practices stemming from playtesting

**The implementation of particular GDPs by participant pairs or individuals often spread through peer activity incorporating or emerging from playtesting.**

For example, the work of the child to add 21 levels to their own game served as a way to publicise this possibility. The process was also spread by that child’s willingness to help others to add that feature to their game. This shifted dependence on myself as a facilitator, or on the instruction-based support documents. This excerpt shows this more experience child coder Te, agreeing to show another child Ch1 how to add new levels.

Ch1: Why’s that enemy in every level

Te: He’s not.

Ch1: Can you show me how you add more levels on to yours?

Te: Yeah sure.

Pause

Te: I’m just going to have one go of beating this (refering to his own game which he is playtesing). It’s 21 levels in it. So .. Yeeeeah.

Pause

Ch1: It’s like parcours in Minecraft but times. It’s like playing the game Wipeout. Have you ever played Wipeout?

Te: Er not really.

Ch1: Or seen it?

Ch1: That’s like my second level.

Te: Ah so hard (Te fails at a high level on his game and starts to move off)

Te: (To someone else calling for attention) No I’m helping (Ch)..

\_(Te then follows Ch to his workstation to help him implement more levels.)\_

When Te moves to Ch’s game he playtests it and then looks at the code. He notes that Ch has added a variable for a fourth level but then goes on to demonstrate to to add an array representing the next level, and a conditional statement to select level 4 when level 3 is completed. At Te uses the keyboard completes this work, Ch reads aloud the code which is being typed in by Te.

Exchanges like this allow the propagation of GDPs. The process of playing a game of another and sharing your appreciation of it invites participants to add new patterns to their own game. In this example, the process is very direct with the one asking another to help them directly. It is very likely that Ch has noticed Te helping others add levels to their games and thus this may help him to feel empowered to do the same. The propagation here is emerges from and is completed entirely through peer activity. A different and more common pattern of propagation was that participants notice and comment on a game element or pattern during during playtesting, and then to use supporting resources or facilitator help to implement it. A less frequent pattern involved participants’ diligent and deliberate use of supporting resources to identify and implement features without peer influence. Typical examples of propagating patterns include placing hazards in tricky places like a lava pit, the use of moving enemies and changes to jump dynamics.

In addition to the propagation of main game design patterns, sub patterns and related design concepts emerged organically from the community. The concept of safe zone in the game of Ch and Pa arrived as a direct result of after adding a moving enemies GDP, the extensive use of that pattern dominates the game challenge to such an extent that it is essential for players to quickly identify and use ‘safe zones’.

### GDP used to allow exploration of home and professional funds of knowledge and practices

**GDPs allow participants to share and explore their home funds of knowledge and practices in the emerging learning community.**

In the previous chapter, the ability for participants to bring home funds of knowledge (as per Lit review) into the new learning community was explored via the use of graphical assets and game narratives. In the previous example we can see a similar process occur as Te self test his game exposes the dominant game experience of timed jumping.

Ch: It’s like parcours in Minecraft but times. It’s like playing the game Wipeout. Have you ever played Wipeout?

Te: Er not really.

Ch: Or seen it?

Parcour in Minecraft and Wipe out are both game experiences whose main mechanic is about judging jumps to landing accurately. Ch makes links to his existing experience of games making comparisons between Te’s game, commercial games and his own. In doing so Ch is able to show his knowledge and analysis of gameplay patterns to this community. While his motivation is not clear, one interpretation is that Ch could be making this contribution not only to openly share experience but also as a offering in return for his request for help which he has just made.

In the following example home funds of knowledge are utilised not only in terms of home experiences of games but also in terms of design and problem solving practices. Da and Te are working closely as a pair. Da invites thinking outside of the constraints of the suggested design early in this first session. The following interaction shows a rich interchange where the parent is trying to draw on the game playing experiences to promote innovation in the design of the existing template.

Da: Have you thought about pushing it a bit further and have a different style of game?

Te: What do you mean?

Da: Well the previous style of game was a platform (makes shape with hands) game wasn’t it? You went along and there was gravity pushing down (points down). There are other types of games aren’t there?

Te: Pause. I don’t know what to do thought.

Da: Well quite but what other games are there? again

Te adopts with this suggestion readily once he understands Da’s suggestion. He then approaches Mick with a suggestion.

Te: You could have a game where every 15 seconds 10 seconds you could add and enemy to such and such a random number between such and such (holds up hands to indicate parameters). You could block it somewhere.

Da: So instead of.. instead of the world… the world being sideways. We could have the world being looked down on. (reindicates the change of perspective)

Te: Hmm. How should I do this then?

Da: That’s a good question. Shall we ask Mick to see if that would mess things up or not?

Te: Mick

Mick: Hi ya.

Te: Erm. Thinking about what game to do . I was thinking can we make like a Pac-man game kind of thing (indicates movement of character with hands)

Da: If we had an on the top game rather than a platform game

Mick: I think it could work. You could kind of adapt that game by, kind of, removing gravity.

Da: and see what happens?

Mick: and see what happens.

–

Da expresses his desire to for the pair to try something new by implementing a pattern not in the menu of GDPs provided. Their new choice is a change of perspective which involves a new game pattern of a new movement game mechanic. The specific proposal is to remove a jumping game mechanic and using a 2D top down movement mechanic used in maze and adventure games (e.g. Pac-man and Zelda games). This decision can be triangulated with interview data from Da on the motivation behind his involvement in volunteering at Coder Dojos.

Honestly, it's just it's just my hobby and I love it is the main reason. In fact, it's probably the only reason. If I can, if I can persuade / cajole / trick my kids into being involved at the same time, then that's even better. Personally, I think that's about it. I've always been interested in computers. I love, I love, I love programming. I'm no good at pencils and pens drawing or anything like that. But writing software is the closest I get to a creative outlet. So I just love doing that.

Given this additional perspective, I interpret Da’s influence to divert as a way of embracing a creative challenge and bringing his child along for the ride. However, Da is also aware of potential challenges of straying too far from the template. He does not want to “mess things up”. This tension has a parallel to a professional practice of “forking” code-bases in open-source code communities. The practice of forking can involve taking a code base in a new direction and the benefits of adaption may be out-weighed by disadvantages including the friction involved in splitting an existing community and duplication of effort. The parent checking with a Mick a guiding community member about the advantages and disadvantages of a major fork in the code structure mirrors this professional tension.

My own positive response to their suggestion was driven partly from knowledge of Da’s cultural background a both a professional coding and a volunteer supporting children’s coding programmes. While simultaneously checking with other groups that they use the starting template as a base, to avoid overload as previously discussed in design decisions, I encourage this pair to see what happens as a potential learning opportunity. I am conscious that the change of movement may open up different possibilities for new game patterns that this pair may be able to solve. This outweighs the possibility that the pair will get bogged down in complex code problems or structures which may be beyond the capacity of the young person. After all even if they encounter father must solve, the apprentice does not need to understand everything in order to benefit from observing the master at work.

### GDPs and the role of documentation and supporting resources

**Participants draw on their cultural experience and access to supporting resources and processes at home and from work contexts to guide interactions with others.**

A broader description of the design and use of supporting documentation is explored in previous chapters. Here I explore the specific use of GDP related resources. I set up the working pattern based on my professional experiences and my own academic and cultural interests. However, my choice of a walled garden approach in terms of limited design patterns and bespoke documentation based on patchable code snippets created a possible tension with the professional practices of some parents with coding experience.

Code examples were initially the starting resource. The idea being it was quick to see the behaviour in context. The, use of code examples by Te in 2019-05-08 shows the effectiveness of the use of code examples by participants. The following example outlines the process of Te finding a GDP to implement and adding it to his game.

NOTE – THE FOLLOWING TABLE NEEDS MORE DEVELOPMENT - TIMINGS ARE KEY

|  |  |
| --- | --- |
| **Time and Dialogue** | **Activity** |
| 0:00:00 | Te begins the main part of this session by playtesting his game. Piskel and documentation tabs are already loaded onto his screen as part of the setting up process by facilitator.  Te tries different tabs on the browser until he finds the graphical memu of GDPs.  *See screenshot below.* |
|  | |
| 0:00:55 | T clicks on the Moving Enemies image which opens up a new tap with a code example.  Te examines and scrolls down in the example sourcecode.  Te swaps tab to his game source code. He removes a variable called tween1.  Te returns to example sourcecode and scrolls until he finds and highlights a large section of the relevant piece of code that will add a moving enemy to his game.  *See screenshot below.* |
|  | |
|  | He checks the code and then checks the game output to test that the code creates the desired behaviour.  Once this is verified he progresses to copy and paste the code from the create function of the sample code to the create function of his own source code. |

The timings of the process of patching the code show that the learner is hesitant in the process. He checks the code and then checks the game output to test that the code creates the desired behaviour. Once this is verified he progresses to copy and paste the code from the create function of the sample code to the create function of his own source code.

In the next iteration of game making Te works with his father Da.

NOTE - REPLACE WITH TABLE

Te: Right what shall we add next? So it follows?

Da: yeah that would be interesting wouldn’t it.

Te: you know like the ghost in pac-man

Te: I think it’ll be in here.

Te navigates to the online pattern menu.(Looking at menu of Game Mechanics. He can’t see what he is looking for.

Te: We could add some player health.

Da indicates at the moment this works in the game. They appear to be at a blockage

Te: Shall we try to get Mick for this?

Da: Ah we could google it.

They google "phaser pacman"

In this interchange Te looks to try replicate a previous process of work where he scans the provided menu of patterns to find one that matches his goal. Rather than persist with the imagined pattern Te suggests progressing by changing goals to one which is listed in the pattern menu. The father resists this diversion by suggesting that they use a search engine to step outside the sandbox of provided tutorials and code examples to find resources from communities on the internet.

The pair’s process of using supporting documents here as they work on adding a pattern of *following enemies* is very different from previous example of self-directed code patching from code examples. The professional practices of accessing professional documentation involves cultural elements here of a family learning culture between these two playing out in these processes. Te and Da’s design process is more guided and focused than many other participants. As this interchange exploring professional documentation resources continues, the father starts as a facilitator taking a lead from the direction of the child. However, as the dialogue progresses he begins to be more directive, initially by asking leading questions and testing existing knowledge. Finally in order to complete the research task after reaching the beyond of the child’s knowledge, the father issues a series of more direct instruction, directing the research and proposing a coding solution for their new game design pattern. (INCLUDE TRANSCRIPT AS APPENDIX?)

This approach appears to be influenced by Da’s experience as a software engineer and volunteer at Coder Dojo (Glossary). Interview extracts (included as appendices) show a direction to support the novices direction as a facilitator where possible.

I try never to touch the keyboard of who's there. If they are stuck on something I always tell them what to do. Even if it's then taken me five minutes to explain what a semicolon is. And point. It's that key. Because it was just, I could do it so effortlessly. I think I'm sure I put people off very quickly by "Dave did something really quickly. I don't know what it was.".

This extract from interview data indicates a priority to support the learner to develop independently but to still be very present in the support process.

The interaction of Te and Da could also be studied from an interpersonal foci, one of guided participation. Da is modelling these practices, speaking them aloud and asking questions to test his son’s understanding. In the next section, the guided participation of other family pairs and more temporary pairings are explored.

## The use of GDPs examined from an interpersonal focus

This section examines the use of GDPs to facilitate guided participation via interpersonal interactions in game making. Guided participation in this context involved, guidance on organising design activities, various forms of problem solving and help to shift design perspectives. While the primary source of material is from pair interactions between children and parents, at times peer interaction between non-pairs developed into guidance. For example, in the in the example above where Te provides assistance for Ch1 in adding levels to his game.

### GDPs used to facilitate prioritisation

The following exchange between participants Fi and Ma shows GDPs being used to try to organise future activity.

TURN IN TO TABLE WITH DESCRIPTION

Ma: I’ve brought the music, and also we could just concentrate on one thing and just change that. You know, keep working through.

Fi: Yeah. I think I want to get an enemy in - oh no - my person animated.

Ma: So you want to get your person animated that’s the main thing.

Ma: Shall we concentrate on that and changing the platforms into something different?

Fi: Yeah.

Ma: Yeah?

Fi: I also want to make a theme tune.

Ma: Yeah. It’s, that’s what I mean, you can’t just skip around like that.

Fi: Hmmm.

Ma: Just cos it gets really overwhelming.

Ma: Yeah..? So…?

Long pause.

Ma: Well I’ll have a look at the code and see if I can make sense of that.

At this point the parent engages with a print out of supporting documentation on added an animation to the main character. This example shows the use of the approximate names of a number of game design patterns by the child adult. These are *get the person animated*, *get an enemy in*, *changing the platforms into something different*, *make a theme tune*. At this stage of their process, some of these patterns have been discussed and sketched out some started but only partially completed. For example, the child has designed different frames of animation but this has not been exported to the right format or implemented in code form. This interchange shows a tension between a more chaotic style of working jumping from one goal to another and a parental motivation to prioritise one work to be done. This tension is outlined when the parent gives an update on progress.

Mick: Hello.

Ma: Hi Mick.

Ma: So, we’ve made quite a lot of progress this week. I think the issue we’re having is that Fi’s super excited so we’re kind of jumping from one thing to another and that’s kinda overwhelming me a bit.

The child’s initial listing of features is a brainstorming technique. Such techniques are used to aid a creative process. However the parent seems to lack a process to map these out and then to work together to prioritise them. Instead he appears to be keen to quickly pick one. His suggested process is to then work through the documentation on that pattern. Fi appears reluctant to adopt this working process which creates a tension which is explore in the next part of this chapter.

**Other examples of use of GDPs for prioritisation** can be seen in the following participants. TO COME

### GDPs aiding the process of division of labour

In the last section, Fi and Ma used GDPs to attempt to prioritise work. However, following this Ma is engaged puzzling over documentation on how to add animation to a character for some time. This results in Fi being blocked from progressing. In the next exchange, they progress to more successfully divide their labour informally.

Ma: Quite complicated. But we can do it. But it would mean a lot of mucking around

Fi: Ah Er

Ma: Which is difficult to do while we’re here. But it’s doable.

Ma: It’s like a project in itself really.

Fi: Project in itself?

Ma: Yeah! (laughing). I just want to know like. We can get him in. So if I ask about the sizing.

Fi: Hmmn

Ma: I think you can edit the size here.

Fi: Why don’t you go here for a computer and you can do that?

Ma: Why. What. While you’re doing what?

Fi: Um making a sound track or something. I could do something like that.

Ma: Ok. Yeah. I’ll see if there’s any more computers in the cupboard.

The father describes previous lack of prioritisation as ‘skipping around like that’ or ‘jumping all over the place’ as ‘overwhelming’ the child seems happy to takes a more piecemeal approach. The child suggests a possible resolution to the current blockage. He suggests splitting and using one laptop each. He names one of the other pattern. The listing of GDPs in the previous exchange seems to empower the young person to direct an informal division on labour. The child continues to work on parallel patterns or component actions of pattern using tools and processes which he is more familiar with. This child also begins researching other toolsets, in this session, research to identify an online tool to create an short audio soundtrack. This serves the child as it allows them to avoid waiting for their father and moving different parts of the overall project forward.

The implementation of some GDP involved the use of different tools and activities. As learners build the familiarity with the component actions needed to implement design patterns, some start to specialise as they divide labour between pairs or peers, For example in this case the child appears to make a tactical decision allowing the father to decipher technical instructions and implement them in the code of their game as they creating assets in non-code / GUI environments.

Different pairs approached division of labour in various ways. In a later example, we see how although Mi and her daugher Ne are working on differnet projects on a fair distance from each other, Mi relies on Ne’s coding help. Sh and Th relied on the parent to do the majority of code implementation and shared one computer. The opening vignette shows the child use the name of a GDP as a way to communicate about the shared work of making a game.

Th: Go on then. Key - Door - Person.

Sh: Person?

Th: Key Door Person.

T gestures with her hands to indicate that her mother is the person she is referring to.

The utterance by the child “Key Door Person” work on the game design pattern called Keys and Doors to the adult. The child appears to consider the level of complexity needed to add a new pattern into the code to be beyond her ability and thus directly delegate the task to her mother. Feedback from the parent indicated that this division of labour was partly due to reading ability.

"Th got on better during the coding once the student who was hovering initially left us alone. Because every time Th hesitated, she jumped in to do it for her. Whereas I know her better so can judge how to facilitate more minimally, and I resist the urge to fix things immediately when she struggles. Plus she can't read yet, so she was recognising the relevant bits of code by matching the individual letters, which takes longer."

The parent outlines her strategies used to address lack of reading ability as a barrier to participation. The design choice of a grid of letters representing different elements of the platform game appears appropriate in the case of a novice learning to code and read at the same time. After the child has delegated a coding task to her mother she undertakes other activities. At times her activities directly contribute to the main goal of game making. At times the parent asked the child to seek help from facilitator. On another occasion when the child appeared bored of waiting for parent to solve a code problem, she approached the facilitator to ask for help on behalf of the adult without prompting. At other times she engages more peripheral activities such as watching older children playtest each others games, or observing community activity from under the table.

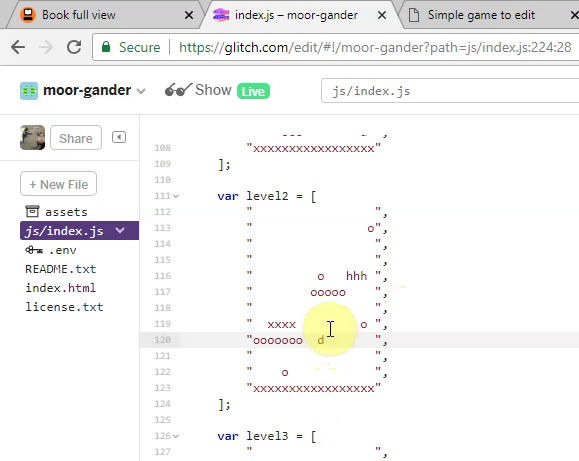
While the child’s activity away from the screen and the main objective of coding and creating assets for their game could in a conventional educational model be seen an non-productive, in a community model of learning this can be interpreted differently. For example as legitimate peripheral activity (Lave, 1991), or as an observation stage of LOPI model (Rogoff, 2014). The possibility for children to not engage in community activities is seen by Rogoff (2003; 2016) as an important characteristics in participation based models of learning.

These divisions of labour seem beneficial for many participants however complex tensions emerge between varied goals as focal activities shift. Study of there interactions show the parents and children adopting varied strategies to navigate solutions to these tensions. The complexities of this design process are explored in more depth in the following discussion chapter.

### GDPs used to scaffold ideation processes

The design chapter explored the tension between reduced choice of genre of game. In my journal notes and observations of the games created, I note that the provision of a graphical menu of GDPs significantly decreased in time spend in ideation phase by providing scaffolding and a restriction of choice. Analysis of participant use of the menu as detailed in the documentation section above Te supports this analysis. Other techniques that leveraged the characteristics of game design patterns to support the ideation process emerged in community design activities.

The use of paper prototypes was one technique used by several parents to support their children to form and develop their design ideas. In our starting vignette the parent notices the child’s difficulty in using cursor and delete/backspace keys to edit a matrix allowing level design. The parent provides a book with grid paper to allow the child to replicate the matrix. The parent is then able to transcribe the design to the code example while engaging the child by checking she has interpreted the design correctly.

 *Graphical representation of code*

The way the code is structured has been chosen to allow a graphical analogue between the lines of code in the form of a comma separated array and the appearance of the resulting game output on the screen. The parent uses the graphical representation of design in the code template as a jumping off point to make a connection to home practice of sketching things out in paper. The process of turn the sketched into reality on the screen and sharing with others appears to be transformative in terms of the engagement level of the child.

There are other examples of the use of paper prototyping of GDPs being used by parents to provide scaffolding for their children. When invited to share about their design process in a post session interview, Fi and Ma also discuss the use of paper to clarify initial GDP ideas in interview data. When asked if they were able to Ma prompts

Mi: Tell me a bit how you came up with those aims in your game, in terms of coming up with a plan.

Ma: What for the things that we need to do to it to finish it?

Mi: Yeah. Oh, just even from the beginning point. How did you plan together as well?

Ma: Well we started off on paper didn’t we. That's the first thing we did. I think it was a benefit actually. We, we did a lot of sketching didn’t we and a lot of brainstorming ideas and seeing and trying to test out whether it would work.

Later in the interaction, the parent outlines a different use of prototyping, that of sketching directly into software. For this pair, the child appears to prefer sketching directly into software. The father also appreciates potential problems of translating ideas from paper into a digital format.

Ma: I was very excited by seeing Fi playing with this because it's interesting that everything doesn't have to be a paper and pen.

It's nice to just for the kids to feel that they can sketch on Piskel straight off the bat without taking a tutorial or being told by an adult. It's really intuitive and you just go straight into it.

Mi: That's an interesting thing because in some ways it started off with people working on paper because I thought that would be really accessible. Yeah. And maybe it was through observations of people just going “Do you know what I’m just happy sketching on Piskel”.

Ma: (Animatedly) The kids, all the kids I saw not just Fi.

Mi: Digital sketching.

Ma: Yeah.

Mi: It seemed to have value in that you were just doing it in the same format that you would use for the game.

Ma: Yeah. I think it's really important. I think the pencil and pen thing just didn't work did it. We sketched... It's got its place. But it's, the kids weren't that interested in using the graph paper to block out Piskel. It didn't translate. It was just easier to block it out straight in software.

A similar sentiment is expressed by another pair as use the they ideation helped by GDPs is the following example which shows Te and Da are creating a new tilemap for a maze game. Te is able to map existing knowledge of tools and home knowledge of the kid of game he is imagining to rapidly make revisions.

Te: Oo. Shall we try to make it (unintelligible). Cos in pac man you can go off the edge.

Da: and you wrap round the other way?

Da: Yeah, yeah. We can do that. Save that for version 1.1

Te continues making changes to the code design.

Da: What’s the theme? What are you drawing?

Te: What? I’m trying to make like a pac-man type thing.

Da: Alright. What if you sketched it on paper first? Or have you got it in your head?

Te: I’m just kinda going for it it. (laughs)

Da: Ok go for it, see what you get up to.

Te: I’ll leave a hole there.

As aligned with the learning design principle of rapid feedback, changes in the code which Te is altering to impact on the new design pattern of a top down game, are immediately apparent in the preview window. As such, Te does not feel the need to prototype on paper.

**Other examples of GPDs helping the ideation process**

NOTE - TO DEVELOP AFTER MORE ANALYSIS OF VIDEO DATA. The use of GDPs to scaffold the ideation process was a common pattern used by X of the Y sessions analysed.

In summary, some GDPs allow the spacial exploration of design in a way that suits being mapped onto paper, or onto graphical software which allows for a similar sketching experience.

### GDPs facilitating designing for others

As explored previously, playtesing as a regular practice can shift learners to a perspective of designing for others. This section examines how some GDPs provoked participants to imagine the experience of end users of their game. Game design patterns focused on gameplay rather than code structure focus on recognisable behaviour. Thus in the same way that visually organised code can aid ideation, designing code framework to help participants alter on a noticeable change in game play can foreground key GDPs.

In the starting vignette of the chapter Th and Sh come into conflict over the imagined experience of future players. The parent is keen to keep a sense of game balance to ensure a sense of challenge for the imagined player. Sh shares “*Must be quite hard to get through that door.\_*” *when Sh places the exit door high above a platform. She then continues -* “It’s no fun having a game without any hazards to avoid.”\_ The child seems determined to remove all hazards. “*It is for me!*” the child counters. She may be aware of the implications for game balance but takes pleasure in this destruction of the key challenge of the game as an act of disruptive play (as explored in the previous chapter). This interaction shows the use of terms from GDPs to both explain and negotiate a conflict over the imagined user experience.

Later Th interacts with one of the student helpers and outlines her motivations in design.

H1: Have you enjoyed making the game?

Th: Yes

H1- Has it been a lot of fun

Th: Yes and I like making it frustrating. that other people find it frustrating!

...

Th: You’ve nearly got to mine. Mine’s very hard to get to.

H1. Is it?

Th: You’ll like it when you get to it.

H 1: How many levels do you have?

Th: Four. Mine’s the last one. And it’s very fun. Do you want to guess about it?

H1: Erghm. Is there lots of bikes?

Th: Yes, guess how many there are?

H1: Is it the whole screen?

Th: YES! Laughs

...

H1: I will get it to your level

Th: You seem to not give up. that’s good

Th turns away to get a hug from her mother.

H1: I got to your level

Th: Good! (laughs)

Th: It’s a secret, special one. (...) If people tried hard they would get to my level.

Th comments that she wants players to bes frustrated when playing the game and that this is a contrast to final level which has only rewards and no hazards. This being a secret, special experience which plays against the norms of platform game design, thus provoking player surprise and fun. She notes the persistence of the student helper who pushes past her frustration to complete the game. Sh remark “If people tried hard they would get to my level” shows her awareness of that not all players will persist in the same way.

A common proposal of research on professional and participatory design processes is that that ideation is more productive when informed by a realistic sense of the end user experience. (explored in Lit Review) The examples show different forms of designing for others. One is to imagine a user experience and make a playable game in the abstract which matches the mother’s approach. Th’s playful imagining of the experience of a more immediate audience of fellow game makers and supporting students appears to provide a tangible motivation with rapid rewards.

GDPs aid the overall process of design for others by providing discreet and clear goals which are nested in the wider goal of making an engaging game. This process involves shifts in perspectives from participants as they engaging with objectives on different scopes of activity. For example, participants may get caught up in a particular design goal on a micro-level, engagement may drive an implementation of a quirky characteristic. When this game is self-tested or playtested, that characteristic may not withstand the shift in perspective to the wider goal of making an engaging game.

**Other examples and interpretation on designing for others / shifts of perspective**

In interpreting data there were other examples of pair partners and peers either commenting on or suggesting to others that they should imagining others user experience to suggest game design alterations.

### GDP driving adoption of emerging technical processes

The emergence of technical processes happens particularly on a pair level. It is at times motivated or sustained by a drive to implement or complete a GDP. The processes are not static but are modified by the community as they are adopted and passed on. Sometimes very explicitly as a parent explains, facilitates or guides a process as with Te and Da in the last chapter with documentation. At times technical processes are passed on informally in more casual help interactions as illustrated in the following example.

NOTE - COVERT TO TABLE

Mi continues to do solo design using the Piskel graphical too. She encounters a design problem. When erasing a part of the design she gets rid of background colour. Mi asks for help from partner but receives misleading advice which does not help her progress.

Mi: Oh no it’s not done that has it?

Mi calls the name of her child across room with theatrical gesture and loud whisper voice

Mi: “Ne!”

Mi then makes face, wiggles head and shrugs at parent peer. The other parent laughs.

Ne arrives to help.

Mi: I’m trying to delete them but they turn light grey.

Ne: So you want to get rid of them?

Mi: What are you doing? You have to tell me what you are doing so I can do it myself.

Mi: laughs

Ne: laughs.

Mi: I’ll just have to keep shouting at you if you don’t tell me.

Ne uses the mouse to select the grey background colour with the colour picker tool, then the pen tool to fill in gaps in the design. She then swaps the active colour back from grey to black by clicking the option to swap foreground and background colours.

Mi - How did you do that so quickly? I’ve got to like, carefully... (makes hand gestures to show a sense of hesitant keyboard use)

Parent peer laughs

Ne bounces up in place and smiles broadly.

Mi- Thanks

Mi: So am I like back with the black now?

Ne: Yeah but if you want to delete it just press X (which switches between foreground and background colours) and then do it.

Mi: Oh X. Alright Bubs. Thanks.

In this interaction the parent is focused on completing the action of creating a graphical asset of a hazard as part of the activity of adding the GDP of including a hazard into the game. Ne appears reluctant to help at first and when she does she is mostly non-verbal and makes changes quickly in a way that her mother cannot initially follow or replicate. The process of explaining this to her parent would be more time consuming. There may also be a power dynamic happening as well with the child enjoying showing proficiency without sharing the process perhaps as a performative demonstration agency or growing status within this community.

We can explore this behaviour from the child’s perspective using terms from Activity Theory. The process of exporting and importing has become an ‘operation’ for Ne through repeated practice. We can see that the operation of changing pen colours on the graphical tool is one which the child has been able to translate into a effortless process whereas the parent is still consciously building her competency. Ne has operationalised the process and it becomes part of the toolset of practices that she can draw on. Mi also benefits, as she is able to draw on the expertise of her child to undertake that process. She is also keen to develop her own competency as indicated by her asking child to explain the process.

#### Other examples of GDP driving adoption of technical processes

References to such emerging practices driven by GDPs were present in many exchanges during directed playtesting and pair interactions. For example, after Fi gave feedback on the jump speed Mi’s game, he continued by sharing a process to redundant space at the edges of design sprite characters.

NOTE - COVERT TO TABLE

Fi: for people with background like yours You can use the cramping tool.

Fi leads Mi to his workstation and involves his father in the process.

Ma: Show what? What are we doing?

Fi: On this one it’s like this.

Mi: Oh that’s good how did you do that?

Fi: The cramping tool. (laughs nervously)

Ma: The what?

Fi: What… Is it cramping? (gestures with hands as scrunching / clutching motion)

Ma: For doing what? What did we do? I don’t know what we’ve done.

Fi: People have used the whole block.

Ma: Oh yeah. We’ve just cropped it. So it’s got no border around it. So you don’t set things off when you get really close to them.

Mi: Aaaah. I see yes. Cos the corners actually could have. (makes a small square gesture with fingers)

Ma: In Piskel. You can crop it to the sprite - cause it take that area too. (gestures – draws a large square with hands – then gestures to the edges). You approach an enemy if you’re close to it, it’ll trigger it.

Mi: ‘cause, sometimes you think how am I just sitting on this ledge here?

Ma: And you’re floating?

Mi: Yeah. Yeah that’s what’s happening. So..

Ma: So you can put you’re sprite back in again and you can crop it down.

There are many other examples of processes being adopted by participants through the implementation of GDPs including the fluidity of navigation between playing and coding window shown by all younger participants and many adults and the development of keyboard and mouse coordination to facilitate navigation within the code environment and external support resources to facilitate the code patching process.

## The use of GDPs explored from a personal focus

In line with conventional schooling approaches, computing education in formal settings has large focus on the acquisition and testing of personal knowledge and skills. However, following Rogoff’s interpretation of this personal plane as participatory appropriation (Rogoff, 1995), knowledge or processes which individuals adopt, reuse and transform fits within this plane. Thus beyond solo activity demonstrating personal knowledge, expressions of personal knowledge or practices when they are shared back into the community activity are also valid here.

This chapters’ examples of interaction of the participants shows the development of the effectiveness and confidence in participants personal communication surrounding articulation of characteristics games and vitally the processes involved in their creation. This section explore participants experience of GDPs from a personal focus and in particular an examination of and practices of debugging and product revision.

### Use of GDPs to support debugging and the product revision process

*NOTE - This section is particularly underdeveloped at this stage*

Many participants spent significant periods of time improving, testing and fixing coding errors in their games. Analysis of the coding of video data showed that revision and debugging was often a solo effort. In a way that mirrors the spread of other creative technical processes, certain revision and debugging practices that were transmitted through interaction with the facilitator that were adopted and used by the community. Some practices were straight-forward, for example the swift navigation between the source code window and a preview window of the live game. Others were more specialist like the use of the developer console of the internet browser to debug JavaScript errors or the process of hovering over red dots in the code playground to explore error messages.

In the area of product revision the repeated, solo, incremental changes of the details of implementation of game design patterns indicate a personal appropriation of concepts like game feel and challenge. The experience of debugging appears to be a particular practice evoking certain feelings. Feelings of frustration alternate with elation at solving a tricky bug. As I built proficiency as a facilitator I began to identify different kinds of errors emerging as explored in the previous chapter. The use of code patching often provoked glitch bugs which where actually behaviour did not match intended behaviour. In analysis of interactions with participants when trying to solve coding blockages, I note different strategies in responding to such errors. For some participants I quickly solve them with short explanation to allow them to continue. For other participants who I judge to be receptive I may celebrate the glitch and explore with them the opportunities they provided to understand the related code in a way that allowed the exploration of more abstract concepts using a concrete example afforded by the mechanics of the game design pattern. The following section provides an example of this kind of interaction and examines the surfacing of computational thinking concepts in particular.

### GDPs as a way to surface and discuss embedded computational, design and systems concepts

As explored in the last section, the implementation of GDPs and resulting errors can surface computing concepts present in the concrete application of code that have emerged organically at different stages of the creative process. Taking Wing’s more abstract definition of CT, many examples arise in recorded interactions without being explicitly taught.

*Decomposition* is shown in several of the examples revisions to the agreed overall goal which break a larger problems into more manageable steps. For example Da the parent suggests to his child “Save that for version 1.1”. *Generalisation / pattern recognition* is present in the work of nearly all participants as The 3M approach lends itself well to exploring pattern recognition as patterns are readily available to participants in starting code and the extra patches that are added. *Sequencing / algorithms* are frequently explored in the resolution of errors with participants. In one interaction with Sh, exploring how a bracket placed in the wrong place can effectively break the game yielded a productive discussion on the importance of correct code sequencing (PERHAPS INCLUDE AS APPENDIX?). *Abstraction*, identified by Wing as the most vital CT concept, merits a deeper examination and is covered in a later section of this chapter.

In the previous chapter, the use of a map of learning dimension in the studies design was examined in relation to contextual tensions relating to the motivations surrounding of curriculum concepts. Beyond this broad mapping of systems and computing concepts to aid facilitators to highlight I also sketched out metacognitive activities to explore these concepts on completion of each GDP. As explored in the design chapter, later revisions of the design of supporting materials for each GDP included links to online descriptions of design, systems and computational concepts. Thus, beginning with experience and progressing to analysis in a sway that mirrors reflective professional practice. However analysis of my journal entries show an ongoing reluctance to shift learners away from the practical implementation of repeated game design patterns to focus on more abstract, de-contextualised conceptions of the knowledge.

My concern hinged on the potential disorientation of the learner that imposed shifts of focus may provoke. As learners shift between different stages of creation the object of their activity shifts from the larger goal of making an engaging game to a narrower goal of implementing a game design pattern to narrower still of completing one of several actions to complete the implementation of a GDP. In the language of activity theory the change of objective denotes is a shift in scope of the activity system.

In the literature review and methodology chapter we examined different interpretations of agency used by researchers using activity theory (Hopwood, 2022). In interpreting the results of this chapter it is of value to explore Sannino’s concept of transformative agency by double stimulation (TADS). This concept of agency is of particular value for this study as it acknowledges both the transformative role of the learner to the learning environment in a way which reflects the mutual development of this design.

My intuitive reluctance to impose shifts in the scope of activity systems, can be interpreted through the lens of TADS. For me to prompt a shift to a activities to reinforce recognitions and connection of learner generated code to computing and systems concepts, would impose a objective (first stimulus) and expose a new set of secondary stimuli for learners to draw on. THIS SECTION NEEDS DEVELOPMENT IN RELATION TO EXISTING RESEARCH.

It is of value to examine the learning context and the motivations of the learners both children and adults. Unlike formal schooling setting there are requirements of teaching to a curriculum and potential exam content. Thus as there was no external imposition, and no organic desire to explore more abstract concepts, I could trust my instincts as a facilitator to not detract attention from participants following an organic and flexible pattern of implementation, self-testing, improvement and playtesting.

Indeed following foundational literature of this study on CoP and CoL, personal appropriation of practices and concepts is demonstrated in community activity through evolving peer practices. As explored previously pairs or individuals alternate between community playtesting and pair/individual design work. They share comments and the are guided in future design decisions by their interaction with the games of others.

One to one instruction from facilitators was limited which encouraged the community to teach each other. The community of learners began to pitch in in their own ways and develop their own practices. Here learner agency is transformative not only of personal dimensions of learning but also the cultural setting, practices and tools available to learners. The process of playtesting other games allowed participants to share their on emerging interpretations of game making concepts like game feel and challenge without being directly taught. Further, the community sense of what is appropriate of fun evolves as mutually and therefore cannot be taught explicitly.

## Discussion on the use of game design patterns in the 3M learning design

The majority of this chapter has focused on interpretation of observations of participant interactions. A deeper exploration of concepts is present in the following chapter which synthesises the observations of this study in relation to broader research. The remainder of this chapter begins this process with a closer look at the use of game design patterns as a intermediate design construct to facilitate coding education and at learner agency in relation to the user of GDPs in this data.

### Discussion on the use of design patterns and pattern catalogues

The process of creating a learning design where students were able to choose from a curated set of game design patterns is covered in the design chapter of this thesis. In this section the response of users is explored in relation to other research on this areas.

Firstly the concept of a restricted activity is well trodden ground outlined in many guidelines including Bruner’s reducing degrees of freedom (Wood et al., 1976). In a design education intervention working with 11-12 year olds Eriksson and colleagues (Eriksson et al., 2019) used a collection of curated patterns to prompt learners to analyse and then propose changes to an existing collaborative game called Stringforce. It is useful to compare the supports and approach of Eriksson’s Stringforce study with that of the 3M intervention of this study.

The overall goal of the analysis of GDPs also differ, in the 3M study the principle goal is analysis of the use of GDPs by learners to aid their goals in creating their own games. In Eriksson’s study the principle goals to is to address the perceived “challenge how to make results from research work related to this within Child-Computer Interaction (CCI) field easily transferable to future CCI research.” (Baykal et al., 2019) The Stringforce study involved learner analysis of games, the ability to change level design via graphical editor and co-design of proposed conceptual changes to existing games. Unlike 3M it did not involved the learners then adding new patterns to games using code.

One similarity is the number of patterns presented to learners. In this iteration of the study 3M presented 20 patterns in the menu of options and the Stringforce study selected 14. Their selection criteria for patterns to include in co-design stages included the following concerns; concrete patterns were favoured over more abstract ones to aid the learner comprehension, patterns chosen matched the learners’ capabilities, patterns that were game mechanics were also prioritised as were pattern suggested by the learners.

In the 3M design the process of offering a choice of games patterns emerged to counter a previous open design process of which many learners found too challenging. The patterns emerged chiefly in response to requests from learners and partly from facilitator decision making which broadly matched the criteria of the other study. However, in Eriksson’s study the authours selected from an extensive, pre-existing pattern collections (Bjork and Holopainen, 2005).

While their results on the advantages and challenges of patterns use focus principally on the perspective of designers (as a way to frame analysis of the activity and shifting interpretations of pattern interpretations), the participant is also shared. I explore some of the opportunities and challenges they surface addressing learner perspective.

GDPs served both researchers and participants by *providing a common language* to clarify first learner expression and researcher’s analysis of gameplay experience. Gdps functioned as an *inspirational structured design tool* Eriksson’s study outlines the utility notes teacher observations that GDPs served to stimulate learner imagination and ideation stages. The use of a patterns and their collection as a form *intermediate-level knowledge* by both researchers and participants is under-explored in this study but builds on related work by two of the authors and is explored in the next part of this chapter.

In 3M the use of GDPs by participants has been outlined in this chapter. A summary is included in Table 6.x

SUMMARY TABLE HERE -

This table will be developed by adding a column “Summary of user actions (use perspective of agency and affordances in processes)”

https://docs.google.com/document/d/1gBAwPkOtvyFJyoJeoK6HmOTyZ6np5fcqn5Jf5\_j2ZLA/edit#

### Game design patterns as a intermediate-level construct to facilitate developing coding fluency

**Summary:** Game design patterns provide learners with a suitable vehicle to engage with coding practices partly due to their position between abstract computational concepts and concrete implementations of code structures.

#### Problematising conceptions of abstract / concrete concepts in computing education.

NOTE - PARTS OF THIS WILL LIVE IN LIT REVIEW.

The term abstraction has varied interpretation even within the field of computer science education (Hazzan, 2002). Papert and Turkle’s (1990) encourage a of diversity in approaches to teaching coding beyond a formal, abstract approach “that emphasizes control and through structure and planning”. Their celebration of concrete coding approaches, including the use of tangible physical and digital objects, and more piecemeal, bricolage approach has influenced the design of popular educational programming software and pedagogy through the constructionist school and maker movements (as explored in the literature review). In a challenge to this article Wilensky (1991) questions the nature of abstract in this context arguing that all objects and concepts are abstract until familiarity makes them more concrete to the user.

Tedre and Denning’s (2016) review of CT cautions against a too narrow definition of CT that highlights formal abstractions as represented by Wing’s take on CT (2008). This is not to argue that Wing’s approach to CT is without technical merit (Lodi and Martini, 2021), rather that its adoption by educational bodies like CAS in the UK and similar bodies internationally has risks. The inclusion of formal CT frameworks in curriculum and formal testing has provoked mechanistic teaching of decontextualised concepts via formal teaching methods to the detriment of hands-on exploration and creation of personally meaningful projects (Resnick and Rusk, 2020).

In teaching computing pedagogy the concept of levels of abstraction can be taught to students to help them undertand the level of abstraction that they are working at (Statter and Armoni, 2016; Waite et al., 2016, 2018a). To quickly review LOA, the levels are Problem, Design, Code, Running the Code. And the purpose is, “Levels of abstraction has been interpreted as a hierarchy to enable teachers and learners to describe which level they are working at, rather than as a methodology for programming projects.”(Waite et al., 2018b)

This conception of levels abstraction is also present in the analysis of the different scopes game making activity systems. Through this lens the most abstract activity system is the larger one who’s objective, to make an engaging game that tells an environmental story, aligns with the problem level of LOA. The level between abstract and concrete is that of choosing, implementing and testing game design patterns, which aligns with design. The most concrete in this interpretation is then the implementation of different lines of code or creation and migration of digital assets.

#### Implications for facilitators and designers

In professional coding environment and training programmes design patterns, particularly in object-oriented approaches and in the domains HCI are seen as a middle ground between abstract CT concepts and more concrete techniques.

In order to be useful, patterns must present an abstraction of good practice at a meaningful level of granularity. Formulations that are too abstract will be impractical in real design use; those that are too specific will be difficult to re-use in new scenarios.[@dearden\_pattern\_2006, p. 20]

Earlier in this chapter the work of Eriksson and colleagues using gameplay design patterns with young people (2019) drew inspiration from the value of design patterns as a form of “intermediate-level concept” as advocated by fellow researchers as a way of sharing results of research (Barendregt et al., 2018). Similarly, the implementation GDPs was the identification as a key unit of activity and analysis for this study, the justification being that it was at this level that richest use and development of tools and processes by the emerging community took place.

The observations of this chapter show the advantages of GDP as an intermediate design concept, hovering in the space between too concrete to be repeated and too abstract to be grasped by novice game makers. GDPs create a tangible link between concrete player experience and the affordances of a guided creative process. Learners use of GDP as relatable and flexible constructs that facilitate communication, sustaining engagement, planning and division of labour. The creation of designed objects using GDPS aid the personal appropriation and the propagation of technical and social processes game making practices. Previously abstract concepts or processes become concrete through familiarisation via direct use and indirect observations through community participation. In many of the outlined uses of GDPs in chapter we can see processes at play that help bridge shifts in design perspective. Both the GDPs and the sub-actions of the wider activity design become short-cuts which stand in for previously tricky to complete set of actions. Rather than promising the transfer abstract concepts to other domains, we see learners build competency in participation in replicable processes. These processes which aid future iterations of the GDP implementation design cycle. The process of operationalisation of these sets of actions contributes to the creation of an informal, complex networked resource of operations which complement the more visible curated catalogue of GDPs.

Game design patterns or their fragments are used as a form of design short cuts. Examples from the above include, *get an enemy in* , *animate player* or *get it in the game* (when referring to transferring an graphical asset from authoring tool to the coding environment). The advantages of such shortcuts are, as discussed, to help with the prioritisation and ideation processes, to facilitate peer propagation of ideas, and potentially to inform debugging and improvements to increase game playability.

NOTE - SUPPORT NEEDED FOR SOME OF THE BELOW - AND MORE DISCUSSION NEEDED ON CHALLENGE IN GENERAL

There are challenges of the use of these short cuts and at a more general the design choice to lead with a menu of intermediate-level constructs in the form of a menu of GDPs. There may be confusion over use of terms to new comers and these GPD related terms may hide more complex patterns within the name. For example, the shortcuts in Fi and Ma’s interaction hides a large amount of problems solving which seems “overwhelming”. GDPs can limit the ideation process through an accelerated approach. Also as the menus themselves are not all used by students, and while GDPs do propagate from student to student, which risk further constraints on the process of asking questions about user experience and exploring ideas before committing to implementation.

Despite these challenges, research of data overwhelmingly indicates the positive affect of participants in implementation of GDPs. This is shown particularly in a growing sense of mastery towards technical processes becoming second nature and the resulting ability to share them with family and other peer groups. On a related note, the next section addresses the development and nature of agency in this learning design.

### Emerging concepts of learner agency in the design

This chapter has begun a process of exploration of the nature and evolution of agency in the practices of the participants through the varied use of GDPs. In this final section I continue to explore emerging thoughts on learner agency in my design through the lens of existing research. The theme of conceptions of learner agency in this practice will be expanded upon in the following discussion chapter.

**On affordances and anchors**

Affordances have been misused, assigned magical properties, including agency of themselves. However, affordances are originally conceived of as part of activity, not separated from it (B\a erentsen and Trettvik, 2002). Sannino augments the concept of transformative agency by double stimulation (TADS) with a metaphor of a sea vessel warping using kedging anchors.

We may think of the second stimulus as an anchor. Anchors are commonly understood as stabilising devices to prevent a vessel from moving. However, not all anchors have this function. Beside the heavy-weight anchors, there are also kedge anchors serving the purpose of ‘warping,’ that is, pulling the anchor once it has settled on the ground, for moving the vessel away from a problem area. [@sannino\_transformative\_2022, p. 4]

In this metaphor emphasises the active volition of participants to overcome tensions and blockages in learning. In our context learners would throw an anchor of intention out into the learning environment to then pull on to Not all throws will be successful. The anchor may slip or it may catch on something in the learning environment that allows the leaner to pull

Affordances in the learning design can be viewed in this frame as a catching point for these anchors (Hopwood, 2022). An effective learning environment provides a sea bed with many rocks (affordances) for warping anchors (volitional acts of participant agency to transform learning).

In the next chapter, I propose that The implications of these combination of concepts the practice of designing and facilitating effective and engaging creative and technical learning environments. I will extract observations from this research that extend beyond the process of coding into other domains.

**Tensions between facilitating agency and norming practices**

This chapter has explored the use of game design patterns by participants to aid the development of their game making practices. Participants are able to use the affordances of the existing learning design and add their own evolving practice to them as a way of expressing and building agency.

However as a seeming counterpoint to this growing agency is the norming effects of concepts that gain community currency in playtesting. The repeated attempts by participants to make the jumping mechanic of Mi less frustrating can be seen as a potential drag on the agency or autonomy of Mi as a designer. However this may be a false dichotomy. Such norming practices can be seen from a different perspective. The following chapter begins with a deeper exploration of learner agency in relation to existing research in this domain.

**The freedoms and restrictions of playgrounds**

While the process of direction may be less totally learner-driven approach than the first iteration of the learning design P1, working with a starting menu of game patterns with support still provides challenges to learners and complexity of working processes. Design blockages still occur, and participants have to work with facilitators in depth to overcome issues. There is the possibility to adapt existing patterns, to add new patterns from outside the curated collection or to shift to new game paradigm requiring a different set of patterns altogether.

In this research, similar metaphors have emerged in the pedagogical and technical process surrounding the concept of playgrounds and gardens. In the previous section the use of a curated set of design patterns can be referred to as a walled garden or sandbox. The process of checking the performance of games is called playtesing. The web-based environment which reduce the complexity of web development and provide community and immediate feedback are named code playgrounds. While some of this language is specific to the creation of games, other terms are also prevalent in non-game coding.

These metaphors invite a connection to play theories concept of as the magic circle. Play theorist outline the value of stepping into a more controlled area of voluntary experimentation where the fear of failure is reduced. Game rules are norms which seed participation in community processes. The playful context of the game’s magic circle can facilitate participants to adapt norms and rules to their own playing styles. Through this lens, the interaction of playtesting, code playgrounds and a sandbox of game patterns emerge as a key practices to facilitate and maintain learner agency. The discussion of the next chapter explores the intersection of these elements in more detail.

B\a erentsen, K. B. and Trettvik, J. (2002) ‘An Activity Theory Approach to Affordance.’ *In* *Proceedings of the Second Nordic Conference on Human-computer Interaction*. New York, NY, USA: ACM (NordiCHI ’02), pp. 51–60.

Barendregt, W., Bekker, T., Börjesson, P., Eriksson, E., Vasalou, A. and Torgersson, O. (2018) ‘Intermediate-level knowledge in child-computer interaction.’ *In* *Proceedings of the 17th ACM Conference on Interaction Design and Children*. Trondheim Norway: ACM, pp. 699–704.

Baykal, G. E., Eriksson, E., Björk, S. and Torgersson, O. (2019) ‘Using Gameplay Design Patterns to Support Children’s Collaborative Interactions for Learning.’ *In* *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. Glasgow Scotland Uk: ACM, pp. 1–6.

Bjork, S. and Holopainen, J. (2005) *Patterns in game design*. 1st ed, Hingham, Mass: Charles River Media (Charles River Media game development series).

Eriksson, E., Baykal, G. E., Björk, S. and Torgersson, O. (2019) ‘Using Gameplay Design Patterns with Children in the Redesign of a Collaborative Co-located Game.’ *In* *Proceedings of the 18th ACM International Conference on Interaction Design and Children*. Boise ID USA: ACM, pp. 15–25.

Hazzan, O. (2002) ‘Reducing abstraction level when learning computability theory concepts.’ *ACM SIGCSE Bulletin*, 34(3) pp. 156–160.

Hopwood, N. (2022) ‘Agency in cultural-historical activity theory: strengthening commitment to social transformation.’ *Mind, Culture, and Activity*. Routledge, 29(2) pp. 108–122.

Lave, J. (1991) ‘Situating learning in communities of practice.’ *Perspectives on socially shared cognition*, 2 pp. 63–82.

Lodi, M. and Martini, S. (2021) ‘Computational Thinking, Between Papert and Wing.’ *Science & Education*, 30(4) pp. 883–908.

Papert, S. and Turkle, S. (1990) ‘Epistemological Pluralism and the Revaluation of the Concrete.’ *Signs*, 16(1).

Resnick, M. and Rusk, N. (2020) ‘Coding at a crossroads.’ *Communications of the ACM*, 63(11) pp. 120–127.

Rogoff, B. (1995) ‘Observing sociocultural activity on three planes: Participatory appropriation, guided participation, and apprenticeship.’ *In* *Sociocultural studies of mind*. New York, NY, US: Cambridge University Press (Learning in doing: Social, cognitive, and computational aspects), pp. 139–164.

Rogoff, B. (2003) *The cultural nature of human development*. Oxford [UK] ;a New York: Oxford University Press.

Rogoff, B. (2014) ‘Learning by Observing and Pitching In to Family and Community Endeavors: An Orientation.’ *Human Development*, 57(2–3) pp. 69–81.

Rogoff, B., Callanan, M., Gutiérrez, K. D. and Erickson, F. (2016) ‘The Organization of Informal Learning.’ *Review of Research in Education*, 40(1) pp. 356–401.

Statter, D. and Armoni, M. (2016) ‘Teaching Abstract Thinking in Introduction to Computer Science for 7th Graders.’ *In* *Proceedings of the 11th Workshop in Primary and Secondary Computing Education on ZZZ - WiPSCE ’16*. M&#252;nster, Germany: ACM Press, pp. 80–83.

Tedre, M. and Denning, P. J. (2016) ‘The long quest for computational thinking.’ *In* *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*. Koli Finland: ACM, pp. 120–129.

Waite, J., Curzon, P., Marsh, W. and Sentance, S. (2016) ‘Abstraction and common classroom activities.’ *In* *Proceedings of the 11th Workshop in Primary and Secondary Computing Education on ZZZ - WiPSCE ’16*. M&#252;nster, Germany: ACM Press, pp. 112–113.

Waite, J. L., Curzon, P., Marsh, W., Sentance, S. and Hadwen-Bennett, A. (2018a) ‘Abstraction in action: K-5 teachers’ uses of levels of abstraction, particularly the design level, in teaching programming.’ *International Journal of Computer Science Education in Schools*, 2(1) pp. 14–40.

Waite, J. L., Curzon, P., Marsh, W., Sentance, S. and Hadwen-Bennett, A. (2018b) ‘Abstraction in action: K-5 teachers’ uses of levels of abstraction, particularly the design level, in teaching programming.’ *International Journal of Computer Science Education in Schools*, 2(1) p. 14.

Wilensky, U. (1991) *Abstract meditations on the concrete and concrete implications for mathematics education*. Epistemology and Learning Group, MIT Media Laboratory Cambridge, MA.

Wing, J. M. (2008) ‘Computational thinking and thinking about computing.’ *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881) pp. 3717–3725.

Wood, D., Bruner, J. S. and Ross, G. (1976) ‘The Role of Tutoring in Problem Solving\*.’ *Journal of Child Psychology and Psychiatry*, 17(2) pp. 89–100.