Enhancing Game Jam Experiences:

Finding more productive and focused group work interactions through establishing a framework

Torsten Hansson Interaction Design Year 1 Master project 2014 K3, Malmö University Supervisor: Simon Niedenthal

Abstract

The thesis will focus on the methods of establishing group work objectives and in turn create focused groups that spend more time being productive and enjoying their efforts than having to go through trivial yet troublesome organization and structure sorting evaluation periods.

Key Words

Game Design, Game Jams, Rapid Prototyping, Social Interaction

Abstract **Key Words Figures** [1.0] Introduction [2.0] Background [3.0] Theoretical Foundation [3.1] Problem area [4.0] Related and Relevant Examples [4.1] Achievements and Gamification [4.2] Software Development Methodologies [4.3] Design Games [5.0] Methodology [5.1] Workshops [5.2] Interviews [5.3] Prototypes [6.0] The Iterative Process [6.1] First Workshop Phase: The Idea of an Interface [6.2] Evaluation of the First Workshop Phase [6.3] Second Workshop Phase: Prototyping an Interface [6.4] Evaluation of the Second Workshop Phase [6.5] Third Workshop Phase: Validating the Framework [6.6] Evaluation of the Third Workshop Phase [7.0] Conclusion and Further Research [8.0] Acknowledgements [9.0] References [10.0] Appendix [10.1] Technical Details [10.2] How to use Mutter from the PDF Handbook [10.3] Game Jam Checklist from the PDF handbook [10.4] Physical voting methods from the PDF handbook

[10.5] Game Jam Practical Troubleshooting tips from the PDF handbook

[10.6] Notes on a Voting Interface

Figures

Figure 3.0.1: Development process patterns which are most commonly found.

Figure 4.1.1: An example of an XBOX achievement that would pop up on the bottom of the screen once achieved.

Figure 4.2.2: A simplified visualization of how Scrum works. Gregory Heller. 2011. <u>http://gregoryheller.com/</u>

Figure 4.3.3: Grow a Game. Mary Flanagan, Zara Downs, Jay Bachhuber. <u>http://www.tiltfactor.org/growagame</u>

Figure 6.2.1: A quick sketch of a scenario with status updates in the first workshop.

Figure 6.2.2: A sketch of an interface involving timers and the suggestion list with votes.

Figure 6.2.3: A table with all the suggestions put forth, along with their secondary features.

Figure 6.4.1: A screenshot of the program's layout, along with two timers being set.

Figure 6.4.2: An iteration of the program where progress bars were introduced, together with a separate presence list.

Figure 6.6.1: An iteration where color emphasized the difference of the timers when the layout changed.

Figure 6.6.2: A table of implemented changes made after the game jam event, focusing on interface clarity.

Figure 6.6.3: The difference in color indicating the character limitation in name creation.

Figure 6.6.4: Screenshots of the interface after all the implementations done post-game jam.

Figure 6.6.5: A table of implementations made after the final feedback session, focused on timer information and input.

Figure 7.1: A table of implementations that may be considered for future development.

Torsten Hansson Enhancing Game Jam Experiences

[1.0] Introduction

Game jams are chaotic and have no true structure or form. Generally the task of organizing its structure or form is roughly provided through volunteers after a set of guidelines. This is a problem as not everyone participating in the jam is able to feel comfortable and can concentrate on jamming within their own groups without a clear outline of what to do or how to act. This is not abnormal and arguably it is even encouraged, as the participants are to dictate their own methods and objectives to create their own game within a limited amount of time. Yet with such a limited amount of time the need to have focused and productive groups is imperative. Establishing group work objectives and going through organization and evaluation periods take up a lot of time, which might be better spent on tasks direct and specific to the creation of the game. The desire here is to explore how the experience of a game jam can be enhanced through a framework to grant users a more enjoyable and fulfilling event experience.

One assertion that should be dispelled right now is that this is an attempt to focus on micromanagement in group work. Furthermore, this is not equivalent to known software development methodologies such as Scrum or Kanban and many other existing frameworks. The difference here is the understanding of a game jam project, which is a collaborative effort, and the restrictive 48 hours to do such a project where not so much emphasis should be placed into the development methods or process more than getting accessible communication and information flow and achieving well-planned objectives during that time. We do however bring software development methodologies as a relevant example as frameworks that one can compare to.

This paper starts by defining what makes a game jam event, and later attempts to further understand what makes a game in group work context and settings. The first issue to be addressed is stopping ideation while keeping communications open when ideas have fully saturated the session. The second issue is timekeeping and logistics for the jam, and how to approach the next iterative sessions. The interesting part of these issues is that there are a hoard of resources and free-to-use materials and sources for developing games, yet while there are so many different methods to create games there are no workflow-based tools used to help with the process of creating a game for participants involved at a game jam. The focus group for this project is a mixture of chosen peers who are designers and also previous participants of game jams who are enthusiastic about creating games in game jam settings or creating any content in a short amount of time.

Torsten Hansson Enhancing Game Jam Experiences

[2.0] Background

The first game jam which started 2002 was "designed to encourage experimentation and innovation in the game industry" (Hecker, 2002). Many game jams today follow the idea of creating a game in a limited time on a decided theme or set of constraints, mostly over the span of 48 hours as people can then dedicate a full weekend without the event interfering with their daily lives. There is still an aspect of experimentation and innovation, although now this aspect is more on a personal basis in terms of learning experiences or generating content for possible future use, as the events have become more popular and technology is now much less of a constraint. As Salen (2006) has described about the 0th Indie Game Jam, it was centered around a single technology at first which was the ability to display 100,000 sprites on the screen at once. This later became less about technology in response to the industry, which kept introducing new hardware where "game design creativity goes out the window while all the developers are busily trying to outdo one another in technological splendiferousness". Salen (2006) also concludes her paper with two quotes stating different perspectives of technology still a part of the relationship with game design in respect to innovation, where on one hand technology is needed in small experimental work and accessible gameplay and on the other the ease of participating and understanding play should be the core of innovation. From these two forces a much more defined "indie scene" in game development has emerged which has become more independent and free from the industry standard, leading up to the Global Game Jam event.

The Global Game Jam is an annual event held near the beginning of the year and near the end of the month of January. It is a software- and hardware-agnostic event, meaning that you are not restricted in whatever tools or materials you may use for the duration of the time it takes to create a game.

The Independent "Indie" movement has been a trend which has been adopted into the game industry lately, as the popularity of these game jam events has risen along with other events supporting more independent publishers and also the trends of more grassroot participation such as crowdsourcing, crowdfunding and the ever present open source communities that continue to grow.

The game jam experience can be likened to other events of a similar nature. Events such as hackathons are most similar where there is collaboration of small teams in short, intense periods of a few days in creating a prototype. Yet other events such as LAN (local area network) Parties where people come together to play the same computer or video games on an isolated network have a similar kind of energy. There also may be days in a classroom setting which can be comparable to an experience of a game jam where students work together on a project in a limited amount of time.

The environment of a game jam is much like a festival or a conference, where people convene together for the sake of the event. The motivations to coming to a game jam range from learning, to networking with people, and to generally be productive or have an enjoyable time. Most game jam events start late in the afternoon on a Friday as its a convenient time for people who have work. On the first day of the game jam, there usually is a short presentation followed by a keynote inspiring the participants and giving advice on what to expect of the event. After the keynote, the theme is revealed and participants form groups themselves with people they know and others who are friendly and present. The next few hours are then dedicated to starting the project in this newly made team. Most teams sort out logistics and hold a short round of ideation during this time. Few teams may even already have a quick prototype of their game ready by the end of the day. The second day is dedicated to prototyping, playtesting and collaborating on the project. Unless specified by the event, teams sort out their own food and breaks. The collaboration usually continues throughout the night, where teams spend energy in completing the game being made to its fullest. The way teams handle sleep varies from team to team, where some participants may even go home and come back if they live closeby. The following day leaves room for polishing the final prototype of the game and adding finishing details. There is a final deadline in the afternoon which precedes presentations of the games made. If the turnout of participants exceeds the amount of available time to present work, there may be voting that happens beforehand to help select who presents. A final presentation is given afterwards to all participants which helps signal the end of the event and thanks them for their participation.

[3.0] Theoretical Foundation

A game jam is further definable through its different elements. To Musil et al. (2010) there are Interaction Design paradigms present which takes the pragmatic approach, seeing design as a "reflective, know-how bricoleur, self-organizing system whose strengths lie in compound seeing and experience. The end product is the result of an ongoing dialog, a reflective conversation that presents a solution to a unique, situation-based problem whose constraints are mainly defined by the designer". The paper also mentions that the game jam "fits no known software engineering process" and yet has a basic system engineering process where there are requirements, functions, design iteration and synthesis along with evaluation. In software development there are three fundamental patterns which include the prototyping, spiral and waterfall models shown in the figure below.



Figure 3.0.1: Development process patterns which are most commonly found. Wikimedia Commons.

Regardless of which pattern is used, the requirements during a game jam are "balanced towards the remaining development time and the team's tool proficiency" (Musil et al., 2010). In lieu with the Interaction Design paradigms described briefly above, agile-like scenario and usability testing is used for design synthesis and evaluation. A theory relevant to having a framework is Donald Schön's concept of design as a conversation with materials. Schön argued that materials have the agency in the process of designing. Since the design process is unpredictable, the designer is forced to reflect while making informed decisions about what they work with and what steps to take next. This even applies when conversing with digital materials (Dearden, 2006).

Dearden (2006) explored the material properties of digital systems and different practices with digital materials to correlate with how designers work with real life materials. The concept of a material utterance is introduced, an abstract concept of a designing activity which is made digital and is intended for someone else to read. After setting definitions of some very high-level concepts, it is described what tools and practices are considered for digitally designing. One tool called DENIM is a website design environment where you create a visual mind map with the ability to couple titles with paper mockups of how the website would look. What was noted as important in using such a tool was that users were able to write notes or comments in different fashions so that they "could be properly taken into account".

Combining both of these two perspectives lies a reflective process of doing tasks and following up on them with decisions, while being aware of the current status of every group member performing at a game jam. An on-going dialog between participants would allow most things to be taken into account while still following a structure defined by the participants themselves.

Another important aspect in game jam situations is that everyone contributes to the event. This draws a lot of similarities to Fullerton et al.'s (2008) notion that "all contribute to design", where every participant is able to contribute what they are good at or that every suggestion is taken with consideration. Or other methods as every team would have their own way of dealing with the creation of a game. The result should be that "everyone who works on the game should have a sense of authorship in the final product and be able to say with pride about some aspect of the experience". Fullerton (2008) goes on to say that as this sense is vital there should be good communication established with each team member, and allow for everyone to be heard. Several tips were added in bullet form which include setting up meetings to discuss the status of the project, a suggestion/ideas list, one to one creative talks, open brainstorm sessions, asking peers for help and sharing authorship of the ideas.

As Musil et al. (2010) describe, the format of a game jam set a particular approach with certain elements that can be identified as design-based or development-based strategy which inspire innovation. The different elements are listed as follows: Participatory Design, Lightweight Construction, Product Value-Focused, Rapid Experience Prototyping, Aesthetics and Technology, Concurrent Development and Multidisciplinarity. These elements in turn encourage particular group work interactions. As observed in Kasurinen et al. (2013), these include making changes to the design if there are too many issues, attaining external feedback, or focusing on the technicalities of the game to make it work. Kasurinen et al. (2013) is relevant as the context of finding how well prepared students are to game design development is relatable to the broad spectrum of experience found in different group setups. According to the paper, the students of the Smith course, the "largest problems in the actual development work were related to the communication within the development team". This is reflected in the following study of the game course revolving around the Global Game Jam event described in the paper, where "professional teams communicate more on their ideas and concepts" while the student teams "focused on technical aspects". Observations from the latter study also concluded that more than half of the professional teams would have done their planning differently. Planning in this case is the project management as well as the game design. The student teams would have used different tools more than planning, although this would be in part to the nature of their dogged persistence when focused on the technical aspects of the game development whereas the professional teams were more open to external input and easily accepted design changes.

[3.1] Problem area

This paper will contribute to the area of knowledge of what is called Game Studies. The methods and approach to reaching this understanding would revolve around experiences in previous game jams that we have personally participated in, the workshops and interviews that help prototype the framework for group work and with the playcentric approach to creating such games that Fullerton (2008) has established which bridges iterative design with game design. The qualitative research where the data gathered to verify our theories can be verified as empirical data from the observations we have made with the user workshops that have been conducted. Academic sources such as conference papers, website news along with any blogged materials will help establish this data.

Collaborating in an intense short project in a game jam requires constant group work knowledge. Group work knowledge in this case is a term that describes the situation at any given time in a group, along with an understanding of what's happening. Knowing what everyone on a team is doing removes any conflicts or possible misunderstandings of what the current tasks may be. As the duration of the jam is a short time, there is also a need for the work to be done as efficiently and as smoothly as possible. This means you want a focused team who wants to spend time making things work. Productivity is then a high priority as well in order to get a playable prototype out. These factors therefore lead to the following research question: how would we create an easily attainable understanding of group work knowledge to encourage more productive and focused collaboration?

[4.0] Related and Relevant Examples



Figure 4.1.1: An example of an XBOX achievement that would pop up on the bottom of the screen once achieved.

[4.1] Achievements and Gamification

These are systems and frameworks made to help motivate players/users beyond the game or beyond their real life and daily tasks. Achievements are non-repeatable extrinsic rewards to the players who go out of their way to complete tasks within the confines of the game they are playing. They were initially popularized in games on the XBOX, where the fourth wall is broken and the player is informed they have achieved something through a task that was set beforehand. Gamification are rewards to users who complete tasks given by the parameters of that reward. Arguably both of these systems disrupt immersion of playing a game or getting something done at some level. Given how flexible and unpredictable a game jam is, it also might not fit the need for dynamically changing goals and tasks as they are redefined. This example is put forth to show that there should be considerations in how the design solution is implemented and perhaps also the question of how pervasive this should be for group work to function well.

SCRUM Visualized



Figure 4.2.1: A simplified visualization of how Scrum works. Gregory Heller. 2011. <u>http://gregoryheller.com/</u>

[4.2] Software Development Methodologies

There are frameworks which do work for development of an application or program. These are relevant examples because the iterative process in Scrum or the ideology in Kanban would play a dominant role in the designing of a game if they were to be applied. To describe either in detail is beyond the scope of this paper, but to give a concise description: Scrum has incrementally unified goals for a team, where a short amount of time is dedicated to one iteration with regular meetings throughout. The Kanban method allows a task to be taken from a visual list of tasks in a context where continuous incremental changes are worked on at the same time. Scrum has the structure and flow that is desirable in working on a project, yet for rapidly developing a game it falls short in making decisions and keeping communication open. Kanban on the other hand has very much an idea of what is being sought for with keeping communication at a collaborative level and with pulling tasks from the work queue there is a definitive workflow which is visualized. The issue of using Kanban for a game jam is that its focus is to limit work-in-progress states, which are the core states found when in a 48 hour long game jam. We also do not want to severely limit the choice of method for development either when creating a game, as every team may choose a different methodology to work from.



Figure 4.3.1: Grow a Game. Mary Flanagan, Zara Downs, Jay Bachhuber. http://www.tiltfactor.org/growagame

[4.3] Design Games

Finally as Musil et al. have observed, "due to its rules, a game jam can also be understood as a type of a design game". A design game is an exercise or method in the form of a game where the objective is to find possible design features and ideas based on own experiences. Design games are also made in a way that the players form a team as they receive input and feedback from each other about what they present in the game. Examples of design games would include Flanagan et al.'s Grow a Game, a card game where where different categories of cards are randomly drawn and players will have to describe a game fulfilling each criteria. This is an example where participants of a game jam can use helping materials to create innovative games quickly to both explore possibilities and also as a way of analysis in knowing what ideas and mechanics could work for them. This is very much in tune with Schön's idea of a dialog between material and designer and supports a healthy on-going dialog during the course of the jam.

Torsten Hansson Enhancing Game Jam Experiences

[5.0] Methodology

To test what kind of product should fit the requirement and to clearly identify the problems from a design standpoint, we will first ideate and reflect on possible scenarios. It has been suggested scenarios fulfill four roles, identified by Bødker (Sharp et al. 2007). These are: as a basis for the overall design, technical implementation, a means of cooperation within design teams or as a means of cooperation across disciplines. The first two identifications would be the main prerogative in this case, as the focus of creating this framework relies on it's overall design and implementation to make a believable or working product when creating the prototypes.

[5.1] Workshops

We performed a total of three workshops with users and chosen peers of designers. All were enthusiastic about either creating games or want to improve the group work collaboration process, mostly because of the limiting amount of time to execute this project. Not all users will be designers at a game jam, yet we have chosen to work with designers for their design experiences when working in groups and for their critical design sense when making this framework with them.

The first workshop aimed to determine how users think about working in such situations where there is limited amount of time, and if they have any observations to share about the way of managing the information between group members, or about any potential gaps when performing in a game jam. A lot of this is to get a general bearing on how this framework should be executed. What is important for this first workshop is that a dialog starts openly and that ideas that are generated are able to present positive and negative issues of working in such a situation.

The second workshop will focus on rough prototype demonstrations and sketches developed from the ideas that emerged from the first workshop, and feedback from these testings and discussions will be used to further iterate the designs of and thinking surrounding the prototypes.

The third workshop would focus on the confirmation from the users on the findings so far, if the chosen approach to the solution makes sense to them, and to get overall feedback on the latest iteration of the prototype before iterating upon it further. For the approach and the materials needed for the workshops, it is important that the users can write down or draw out their ideas in sketches or explain their thought processes and the reasons why they imagine it to improve their experiences with working in groups.

When conducting these workshops, inspiration is drawn here from the interaction framework as defined by Cooper et al. (2007). These are a series of detailed requirements that systematically answer how the product may look, and what the flow, organization and behaviors of it may be. Cooper et al. (2007) outlines the following six steps:

- 1. Define form factor, posture, and input methods
- 2. Define functional and data elements
- 3. Determine functional groups and hierarchy
- 4. Sketch the interaction framework
- 5. Construct key path scenarios
- 6. Check designs with validation scenarios

As this is more geared towards goal-directed design and not user-centered, the steps would be appropriated so that after the flow and organization is defined, the user will step in and help sketch the interaction framework, and comment on what is functional and together with the designer help construct a key path scenario that is ideal for the user.

[5.2] Interviews

Interviews are conducted during the workshops, where as a designer the questions are asked and raised when the potential user does not fully explain their reasoning behind their decisions. This would make this an unstructured form of an interview where the question is more spontaneous and takes more in-depth conversations about what is being thought about. An unstructured interview also allows the user to properly justify their answers. In this light, it may be better to consider the interviews conducted as part of workshop activities, and were not done separately.

[5.3] Prototypes

To further test our design ideas, low-fidelity prototypes will be created to allow users to interact and give them a way to experience what the product's uses are and how the experience could be like. Its also a way for designers to reflect upon their design choices (Sharp, Roger, Preece, 2007). Sharp et al. (2007) also describe that Liddle (1996) recommends prototyping to come before writing any code in software design. This is how the use of sketching and low-fidelity prototyping differs to its high-fidelity counterparts when creating a digital artefact.

[6.0] The Iterative Process

[6.1] First Workshop Phase: The Idea of an Interface

This workshop was mostly aimed at starting a dialog and at discovering what a user would do in this situation and environment of a game jam. This workshop was divided into four phases. During the first phase the thesis was introduced and a little bit about what is being researched was stated. To give a quick pitch to the participant on what this is all about and what they would help design for. The second phase involved asking the users to help design an interface where it is possible to understand group work knowledge. Their design should accommodate two ideals: helping the group be more focused, and helping the group be more productive. Because of the time constraints of a game jam, the ability for fast and direct communication at all times was deemed of utmost importance. This led to the only other requirement we had set for this design: it had to include a way to let group members communicate directly, such as a chat window. It was also explained that the communication is not aimed to be used casually, and could for example be a stream of objectives accomplished by group members in chat format.

[6.2] Evaluation of the First Workshop Phase

This phase was with two interaction designers, both of whom have attended at least three game jams. During the process of the workshop questions were asked to clarify any loosely described features or ideas.

The first idea was a set of status updates which were current to each member. It would help indicate what they are working on right now or if they want a group/individual meeting. There was discussion on what happens if there was a problem and that team members may call upon others in the group to quickly assist with it personally or if a screenshot were to be shown and feedback could be generated through the status.

Figure 6.2.1: A quick sketch of a scenario with status updates in the first workshop.

The idea of a screenshot feed, where screenshots show the status of each member regularly was also suggested, where feedback may be requested but otherwise group member's screenshots were regularly updated at short intervals.

Another idea was to use a suggestion/idea wishlist or feed which implemented some kind of voting system. The idea behind this was that group work was not feel ensnared by feature creep, and that "small things can get done whenever someone can look at it without waiting for a meeting". Feature creep is when the ideation for an iteration gets out of hand and developers get bogged down with trying to implement new features and new ideas. Here a screenshot would also help if necessary, and group members may vote for an idea whenever they have the time. We compared this to Google's +1 or Facebook's Like kind of voting, where only voting may only be positive, instead of a rating system which make a suggestion less decisive.



Figure 6.2.2: A sketch of an interface involving timers and the suggestion list with votes.

When asked to design an interface, thoughts about using timers arose. The use of them would help approximate when group members could disrupt others and when to expect the next update. When a task is finished, it was also suggested that a pinging noise would sound to alert the user that they should expect an update.

Talking further on improvements to the chat interface, the idea of a Noticeboard chat for everyone in a game jam was an idea where a group member could ask for help from other groups using the same program. This was also to include a tab on the chat that allowed organizers to send global messages and warnings to participants. Both of these ideas would only function if everyone in the game jam uses it, and it's been pointed out that it would "make it more together". This aspect of including others outside the group will encourage more collaboration as well as communication. Avatar icons were also suggested to help distinguish users from each other. There was also mention of the ability to draw ideas and share them in real time with others directly in the program. It was decided that drawing ideas was a good suggestion, yet drawing collaboratively in real time was not as important, which seemed to make intuitive sense. Alternatively, there could be a redline feature where users may draw over other drawings to help critique or expand further on the idea.

Suggestion	Secondary features
Status/screenshot Updates	pinging, timers, avatars
Suggestion/idea wishlist or feed	screenshot, voting
Noticeboard/announcement feed	none
Drawing	redlining, commenting

Figure 6.2.3: A table with all the suggestions put forth, along with their secondary features.

Looking back at the interaction framework model we were inspired from, a lot of these suggestions were to define the functional and data elements, with a little bit of understanding for what form factor, posture and input methods would be appropriate. Some suggestions were made from visualizing certain scenarios, where elements made more sense once they were drawn out. It was easier to work from the elements first as the phase called for finding ways to make a more focused and productive group. This together with the principle from Sullivan that "form ever follows function" seems clear and distinct in retrospect.

[6.3] Second Workshop Phase: Prototyping an Interface

This workshop focused on a rough prototype quickly developed from the ideas that emerged from the first workshop. The interface being made is to be used as a tool, so that participants would still interact with their team in real life when necessary. This workshop would keep this tool aspect as a point of reference when thinking through different scenarios with the users. Observations about how it should work and what can be improved will be a focus as it is important that communications and ideas between group members are kept relevant and useful.

[6.4] Evaluation of the Second Workshop Phase

This phase was with the same two interaction designers as the previous workshop, together with a concept artist intern who works at a game company. All of whom have previously attended the same three game jams together. This workshop was more like a string of different feedback sessions as each iteration of the prototype needed time in development to implement some of the suggestions before taking the workshop further.

For these sessions we took the use of timers from the first design of the interface and went through how it could function and help improve focus and production. The use of timers was chosen as one of many features that should be implemented. Timers were considered one of the first things to add to the interface from the first workshop phase, although discussions about other possible features took place during this phase as well. A working chat was first implemented, and a presence list to know who is online was needed so that users know whom they can interact with and chat to. Users could not idle in chat without the presence list available to show the possibility of someone reading another's chat input.

For this prototype, the timers were represented simply as the amount of time left next to the username the participant was using. The presence list was combined together with the timers at this point, so setting a timer would make a timer appear next to the username that set it on the presence list. The usernames in this prototype were generated through the program by adding a random integer between 0 and 1000 following the word "jammer".



Figure 6.4.1: A screenshot of the program's layout, along with two timers being set.

The feedback given from using this iteration was that the progress bars were still needed for the timers, and the presence list should be kept separate from the timers. The timers needed progress bars, as the interface at this moment was mostly text and it was hard to discern the chat from the timers. The only indication was through the layout, so the position helped where to expect such elements. For example the chat interface was placed at the bottom and disappeared closer to the top.

The reasoning as to why the presence list should be kept separate was that names that

have yet to set a timer made the timers list confusing and hard to understand. Keeping them separate would allow for timers with their appropriate tasks and usernames to be used exclusively for setting tasks and understanding the group work. The presence list would then function as to which clients are connected and viewing the situation, and who are also available in the chat.

	05m 03m 02h 10m	3m25s 1m8s 1h58m16s 8m35s
jammer139 (14:42:33) hello jammer762 (14:42:38) hi hi jammer382 (14:42:45) 03m jammer139 (14:42:52) 02h jammer383 (14:43:01) 05m jammer762 (14:43:11) 10m		
4 οη (jammer383, jammer382, jammer139, jammer762)		

Figure 6.4.2: An iteration of the program where progress bars were introduced, together with a separate presence list.

It was also suggested that the timers would be better associated to if the participant's name could be defined. Assigning random numbers to each user was not distinctive enough to be able to assign tasks. This issue arose when talking about connecting more than just two users to the chat. It was more troublesome to attempt remembering others by a randomly assigned three-digit number.

When discussing about the implementation of the global timer, the reactions were positive. "It would be good to have an overall timer, [and] together with the other timers it makes sense". The use of a global timer could be used to represent the deadline of the project, and in the relation to other timers, has the potential for being used to be able to plan ahead on what tasks should take priority before that time is up.

Enhancing Game Jam Experiences

	drawing 08m another 32m	loco loco	7m22s 11m48s	 08m 32m
3 online – evn, loco, dasifasdf				
evn (14:28:05) test timer 08m loco (14:28:25) another 32m evn (14:33:47) hello evn (14:33:59) i am going to do this fo evn (14:34:56) let's have a meeting in (loco (14:47:59) drawing 08m	r 05m 01h23mt			
let's hav	e a meeting in	evn	1h9m19s	01h23m

Figure 6.4.3: The iteration made after the workshop where a global timer was created, and usernames were defined.

[6.5] Third Workshop Phase: Validating the Framework

This workshop focuses on refining the prototype made from the second workshop. After the creation of the interface and understanding how it will be used in terms with timers, we should now know what should be improved and see that the prototype still makes sense in such a situation.

[6.6] Evaluation of the Third Workshop Phase

This phase starts with the same three people of the previous workshop phases, this time during the Global Game Jam 2014 event which happened over the weekend starting on Friday, the 24th of January.

Changes to the prototype during the event includes a prompt for participants to enter their name before using this tool, adding progress bars to the timers where one global timer was able to sync to other clients, associating names to these timers, and some repositioning of some of the layout that distinguishes the different elements. The presence list was also altered and separated from the timers both in layout and in code, so they now work independently.

Feedback from looking at this new iteration of the prototype during the course of the

game jam event mostly was in the form of changing the layout and look of the program to have it make more sense. One of the questions that came from the users was the function of the global timer. In the second workshop phase it made sense that a deadline timer for the game jam was visible as well, yet after being implemented was easily confused with the other timers even when it was separated from individual timers as shown in the previous figure. During these dialogues it came up that it may be used for much shorter deadlines and not just to signify the submission time for the game of a game jam. For example, it may be used to say when the next group meeting should take place. It was suggested that the global timer be made a red color and the other timers orange for the reason that the timers as elements should be the first thing the participant should focus on, the chat less so. With this the layout should shift all timers to the top, and the chat and it's elements remain at the bottom. The change in colors would help emphasize the difference between a normal timer and a global timer. At the end of the game jam I ended up with an iteration shown in the figure below. It was confirmed that this iteration improved greatly in its layout and allowed for all the information to be seen at a glance, which would help work in the participants favor.

evn let's meet downstairs	4d2h57m21s / 99h
evn drawing	1h36m56s / 99m
jack coding	58m6s / 01h
jill file sorting and organizing	
john carrying stuff	– 10m42s / 12m
saul whiteboard ideation	– 8m52s / 10m
paul coding also	3 9m4s / 40m
bob RESTING	29m13s / 30m
kevin making sounds and music	5h59m35s / 06h
01:13:22 evn: 99ht	
01:13:36 evn: let's meet downstairs 99ht	
01:14:11 evn: drawing 99m	
01:14:21 jack: coding 01h	
01:14:38 jill: file sorting and organizing 32m	
01:14:57 john: carrying stuff 12m	
01:15:07 saul: whiteboard ideation 10m	
01:15:19 paul: coding also 40m	
01:15:29 bob: RESTING 30m	
01:15:50 kevin: making sounds and music 06h	
8 online – kevin, john, evn, jill, bob, saul, paul, jack	

Figure 6.6.1: An iteration where color emphasized the difference of the timers when the layout changed.

After the game jam event, the feedback concerning how clear it was to understand the timers and chat at a glance was developed further. The figure below is a table of these changes

in order of their implementation.

Implemented changes after the game jam event
Layout cues where the names were separated from the tasks
Character limits introduced to the input
Quick help screen
All timers sync, so it is not just the global timer that can sync
The trigger was changed from 't' for team/total to 'g' for global/group
A button interface for removing timers or adding 30 minutes / 4 hours
An idle count and global timer shows in the program title on the taskbar
The icon turns red and a 5 second alarm sounds when the global timer ends

PDF Handbook on how to use the program (together with overall jam help)

Figure 6.6.2: A table of implemented changes made after the game jam event, focusing on interface clarity.

A lot more focus was put on making things understandable in the next iteration. For example, when names were separated from tasks it was important that limits were imposed on how long a name should be due to the layout. Names could not exceed 110 pixels in width, so text showing what was allowed turns to a dark orange indicating how close the name was to the limit.

> Type your jammer name and hit Enter. (0.0px of 110px allowed)

> Type your jammer name and hit Enter. (109.0px of 110px allowed)

> > Toggle CTRL to get help at any time

Figure 6.6.3: The difference in color indicating the character limitation in name creation.

testUserName testUserName userOne userTwo userThree userFour	group meeting in room C123 drawing a 3d model rigging character drawing more concepts [break] + [snooze] +	Ih23m33s / 01h30m 24m13s / 30m 10m23s / 15m 40m51s / 45m 56m18s / 59m 7h57m34s / 07h59m
03:22:04 testU 03:22:45 testU 03:23:38 user 03:23:56 user 03:24:23 user 03:25:19 user 03:25:40 user 03:25:40 user 03:25:51 user 03:27:04 user 03:27:06 user 03:27:06 user	JserName group meeting in room C123 01 JserName drawing a 3d model 30m One hey One rigging character 15m Two drawing more concepts 45m Three checking files 03m Three [remove timer] 00m Three [break] 30m Three [break] +59m Four [remove timer] 00m Four [snooze] 04h Four [snooze] +07h59m	h30mg 00m 30m 04h
testUserName testUserName userOne userTwo userThree userFour	group meeting in room C123 drawing a 3d model rigging character Coordinate tasks together with this Change your timer by setting it aga	1h22m3s / 01h30m 22m44s / 30m 8m54s / 15m 79m22s / 45m 5 4m49s / 59m 7h56m4s / 07h59m
03:22:04 tes 03:22:45 tes 03:23:38 us 03:23:56 us 03:24:23 us 03:25:19 us 03:25:40 us 03:25:40 us 03:25:40 us 03:25:51 us 03:25:51 us 03:27:04 us 03:27:06 user	Timers are made by typing them at the end "Drawing some concept sketches 10m" "Coding 01h30m" Add lowercase g to set a global (or group) "Let's meet at the group room 03h20m "Deadline for submission 02hg" Scroll through chat with the mouse wheel of Enter+Enter will move to the bottom of The 335px limit is for timers. You have a 5 Four [snooze] 04h Eour [snooze] +07h59m	f of a text h30mg timer g" or Up/Down arrows f the chat 70px limit for chat
13	31.0px/335px coding work 03h	

Figure 6.6.4: Screenshots of the interface after all the implementations done post-game jam.

The limitation of 335 pixels was also indicated for timers, not for chatting itself. The way chat was limited was done naturally, through the area of the input where the cursor reached its limit.

It would be one less thing for the participant to think about when using this tool. The help screen was introduced as a way of getting used to the program as quickly as possible, giving believable examples of how it should be used and how to interact with the interface. The button interface was made to ease having timers and give options to quickly remove and add time to the current timer. The timer showing in the program title, the changing icon color and the alarm all help to give both discreet and obtrusive indications of the status of the global timer. The PDF handbook was a way to clarify what to expect of the program as well as instructions on how to use it. As the instructions alone took a page, we decided that adding more information regarding game jam situations and group preparation was appropriate and necessary together with the program's use.

A feedback session took place to see how well these changes fared, and if it made for more productive and focused habits. There was mention that when timers run out, it was "anti-climatic", especially for the global timer where it was hard to see what it was for, as it disappeared once the alarm sounded. A suggestion of allowing the timer to continue to show with the time elapsed since it was last finished was given. The issue could be that there is a lack of agency with the timer when it finishes, and perhaps there is a desire for a participant to actively confirm they are done. The suggestion would also help in seeing if a group member has not updated their timer status after it ends for a long enough time to give reason to check if everything is alright. At first there was no reaction to why the program's title should have a global timer and changing icon color. After mentioning that the program would be running in the background there was quick agreement that "in that context it does make sense, I hadn't thought that I wouldn't have it open the whole time". Overall "it's clear, it's not that cluttered, timers at the top, chat at bottom, it's very good. It makes sense to have the chat at the bottom and timers top, and the global timer on top of everything. I like that you are showing who is online". One suggestion was to change hitting Enter to type to allow typing to initiate text input. This was brought about since there are no other interactions with the alphanumeric keys which would conflict with typing. They were unsure of adding this feature and giving more feedback without first wanting to try it out at a game jam, as it seems quite naturally logical for both methods as a gamer and as a designer. These two suggestions felt crucial most because it was not intended that participants would have to scroll back through the chat to find out the last timers that were affected, and the mode of interaction needs to be more forgiving for anyone who would want to use this tool.

Implemented changes after the feedback session

Timers that run out idle, showing what the task was

Idle timers also show how much elapsed time has gone since ending

Alphanumeric keys allow for input to start, Enter to type still exists though

Escape key escapes input, which seems natural for input

Volume adjustment and interface for how loud the alarm can be

Figure 6.6.5: A table of implementations made after the final feedback session, focused on timer information and input.

The prototype addresses communication needs, and the main issue of staying on track. It does this with the use of the ability to declare timers via the chat. The timers also correlate focus and productivity, and give this to every individual participant to do. The program does its job and is unobtrusive, and cuts down on running around and seeing what others are doing. There is "incentive for you to do things, where you want to achieve more, timers do that, and at the same time you control them so there's not that much pressure". It was also mentioned that when you set a timer, you feel compelled to do the task you set for it. Seeing other timers also create the incentive of setting a timer for yourself to stop slacking and be part of the group work happening.

In project management it is the job of the project leader to keep things timely and to keep asking the group at regular intervals what their progress is. There are also milestones that are set, where the leader would see if they are reasonably followed during the course of the project. The work being done is therefore kept at an efficient pace, where productivity of the group is at its maximum performance. With the use of the prototype, the responsibility of management is divided to every member of the group. It is up to the group to meet their milestones. If what is said to be done is done on time, where commitments are met, then the role of project leader is fulfilled. Less project management is needed as there is reliability in the group, and there is trust that milestones are met without having to work on compromises between group member and group leader. In this manner there is also a respect in getting work done without interruptions from others, as the pace and times are set by each participant for when group members may interrupt. It was discussed and was considered good for the project leader or person in charge to have an overview of what everyone is doing where there was no need to constantly poke them.

On rethinking the workplace, Fried (2010) relates that removing interruptions and having more time will get more work done. Jason Fried is one of the founders of Basecamp, an

online project management tool and part of a suite of tools which help manage information and collaborative group processes. One way to remove interruptions is to "shift your collaboration between people to more passive things". Collaborative interactions happening at a more passive state would mean that there would be less interactions that are not as important interfering with ongoing work. Notifications may pop up, but they may be put aside while a person is busy and checked later when they take a break. "Even though we're right there, we'll use instant messaging, or email, and if someone doesn't respond, it means they're busy". This was insightful in understanding that physical presence in a working environment does not always mean that a response will be instant, and it is something the Mutter prototype helps accomodate.

When the handbook was talked about, there was an understanding that the text was to be used as a guide that contained methods and ideas on how to handle game jam situations for a group that need direction when using this "game jam kit". The prototype is not just one thing that resolves the problems of group work interactions in a group. The purpose of the handbook is to allow the group several options in understanding their situation, which in turn help enhance their interactions both physically while participating in the game jam and also together with the prototype in getting through the motions of the group's workflow and processes smoothly. In the discussion it was mentioned that the first page was most useful, though it is helpful for those not accustomed to going to game jams and those who need more help in setting up for making a game with others. The handbook "makes the uses of the app more apparent".

As there are many different processes and workflows a group can pursue, it was discussed that leaving workflow open for the group allowed for the most possibilities. The timers help keep the group on time, they "keeps you focused, even if you need to take a break you will finish the task first as you don't want to stop halfway". In the discussion of keeping workflow open, the ability to set the timers themselves and leaving task assignment to the group participant enables them to "do the things you want to do, and as a reminder that you are in the process of doing them".

[7.0] Conclusion and Further Research

Just as there are many different methods in attaining a design solution, where decisions are sometimes pursued rather serendipitously to explore the possibilities and where there is no true solution, there are many possible ways to establish group work objectives which in turn make for more focused and productive group work. There are also boundaries as to where one is more in need of being organized and where one is over-thinking the process and should simply just take action than deliberate.

Through the research on theory done, we have kept with the spirit of everyone contributing to design (Fullerton, 2008), and the notion that communication should be improved within a team (Kasurinen et al., 2013), and allow for ideas and concepts to be communicated on more. The prototype allowed for intercommunication to happen within a team without imposing too much for participants involved. The timers in this way also removed some of the troubles mentioned in the Kasurinen et al. paper about focusing and taking too much time on technical aspects, and instead inspired other game jam elements that Musil et al. (2010) have outlined to happen. As with Dearden (2006) through Schön, designing activities made digitally and intended for others to read are part of the reflective process, called material utterances, which we have interpreted as the basis to have a chat implementation for the prototype made, and the timer-task assignments which show the processes in action. Timers help keep track of progress and stop tangent discourses, while at the same time allow for knowing the availability or presence of others. The name of the prototype, "Mutter" is a hint towards how it is used. Participants who mutter towards themselves about what they are doing and how long it will take, allowing others to prioritize what else needs to be done, communicate ideas and find that balance between the development time and tool proficiency that Musil et al. (2010) write as a game jam requirement.

There is also more focus on contemporary research, apart from the mention of Donald Schön to bring up the relevance of working with digital materials. This was through a conscious decision of working with the aspects of game jam culture specifically. If pursuing a more solid theoretical foundation is necessary, the works of Buxton (2007) about Sketching User Experiences along with Krippendorff (2005) about The Semantic Turn look the most appropriate to expand on. Buxton for understanding what methods and limitations there are to sketching dynamic interactions for playtesters, and Krippendorff for the more science-like understanding of using the digital artifact, it's meaning to the playtesters and how to analyze and evaluate the artifact. The reasons for their exclusion in this paper is to keep focus on group work interactions that are productive and focused during a game jam situation.

The framework is barebone at its current state, with a program that helps coordinate tasks and communication and a PDF handbook which contains a one-page instruction on how to use Mutter, and five pages on practical and essential information. It should be noted that this framework may be used for the development of physical or card or board type games, although it may not be as beneficial. With physical games participants are more likely to use techniques such as bodystorming or exploring a physical place which do not rely on the use of a screen or computer. For card or board games there are traditional materials and the use of real life objects to help develop the game with. In these cases much simpler tools to measure time and assign tasks to keep focused would be preferred. If we were to reflect upon the framework for doing a remote jam, the first thing to address would be what would replace the physical presence of the jammers. Instead of physical presence, there may be another chat set up on IRC or Skype with voice/video communication. The participants would have to use Hamachi or some other Virtual Private Network (VPN) software which will bridge the connections together to use Mutter. The exchange of all the materials used for game creation will have to happen online and considerations about communications breaking should be addressed with alternative ways to contact the rest of the team as necessary. This is very different from normal game jam culture behavior, as being physically present is tantamount to the game jam experience. The following is a fictional use story of Mutter in a game jam situation:

As the only programmer on my team, I was in desperate need of alone time where I can get things done. Its very stressful to contribute such daunting code without another programmer to depend on. Everything needs to be up and running as soon as humanly possible. I have no time for people who need my immediate attention, I have a laundry list of tasks to get through. This jam has been faring a lot better since Mutter was introduced though. Now when people need my attention, they can calmly type it into Mutter and I will see it when I am taking a break from coding. I have less moments now where I am in the middle of coding and someone interrupts, and my train of thought has to start all over. As it is, programming involves a lot of periods where I have to digest what I am coding into the bigger scheme of things. Now with Mutter the group can even set an alarm through the global/group timer if they really need my immediate response. Usually it also means the group comes together and everyone will be there, which means less misunderstanding from person to person. We even chat on Mutter sometimes so I do not even need to leave my seat while coding if I need some quick feedback. I mean I know they are sitting right across from me, but if its just to look at one single feature, they do not need to stand up, walk over to my workstation and then have to walk all the way back to see what I could explain easily in text while uploading what I have to show. There is also no one over my shoulder anymore poking me about my progress! I update what I am up to into Mutter, and everyone is aware what I am up to. I feel less tension in what I am doing and more confident that I have my group's trust in getting it all done. I even think my frustration-free coding is making me a more positive group member.

The next step is to continue user testing and finding ways to be able to accommodate a wide range of different group setups and processes. Maybe the idea that either features get introduced to the group where they may choose to opt-in together to use for this project, or a modular program that is more aware of the status of a project and where users individually may choose their own layout and how to see the information to adjust and communicate accordingly. There are also possible ways to visualize progress or the ability to create visual data during the course of a jam as argued by Turner et al. (2013), where the creation of visuals allow for the data to become anecdata and personal narratives that are generated after experiencing the game jam. Implementing different forms of voting should be the next focus for the program, or perhaps a separate program that can be launched from within Mutter that would aid in voting digitally. The PDF handbook takes up internal methods for voting which may be adapted for playtesting, and making it possible for external voting that is automated allows for quicker feedback sessions to happen. The data from these sessions may also be visualized or mapped later to quickly give an overview of what to prioritize or what could make interesting pursuits or directions. There may be ways to enhance communications between jammers and playtesters alike with such a voting function, that may extend over its intended use. Other implementations and ideas may be entertained, listed in the figure below.

Other considered features and next steps

Suggestive timers or reminders to keep stray and tangent ideation minimal

Total time left for the jam, or ways to schedule the days and time more

Search/filter functions to help navigate through the chat

A collaborative to-do list, or one that may be preset or suggestive of context

Freely set and name divisions in chat for more narrative/organized text

Sketch-based environment through collaborative mindmap and drawing tools

Figure 7.1: A table of implementations that may be considered for future development.

Just like the design games example, having more of a sketch-based environment will also be a possible direction to take as a next step. As Brandt et al. (2004) have noted there are two main properties to design games, "the use of game pieces as vehicles for expressing design moves; and the structuring of concept design activities through game and play". Both of these properties can be used in creating an environment where participants can use playful and sometimes restrictive rules in order to come up with and generate further ideas. These can then be formulated into "design moves" which carry through to the design process and then into the development of a game jam game.

Further research should be made in these next steps, as well as finding more about group dynamics and how participants behave. An unfortunate limitation here is that there is not enough time to test it in enough game jams for the purposes of this paper. There should be more research on what people use in game jams, whether it is a freely used version control like Github (or something localized like Mercurial) or other forms of collaborative platforms like Google Drive or Dropbox. During the Global Game Jam 2014 event, the group we participated in used Google Drive to help distribute files and ideas. It was observed that coordination is important in group work. The tools or methods described in this paper do not need to help create or make a structure or form but the generation of ideas and the freedom to follow the process that may be set as a structure or form will make clear the work that needs to be done individually or in the group. As the program is based on communications and group workflow rather than the exchange of digital artifacts, there is enough freedom for expressing ideas and concepts that encourage this coordination to emerge on its own. This may be taken further in the development of the prototype to include everyone in the jam in some form, perhaps even at a global scale in the case of the Global Game Jam, or the ability to coordinate across different teams and encourage more passive communication that help get what is needed

done.

From the workshop phases done during the time of this project, communication and timekeeping were main motivations for using this tool. It was not specific in mapping out workflow, yet through the use of timers and manipulation of these timers together with the synchronization of the information, the program allowed for its use in workflow-based coordination and handling. It was then left to the group to decide more specific conditions about workflow, for example how to handle ideation saturation and own logistics and group structure. As these vary depending on group dynamics and individual skills, it was considered better to leave it open for the group to find the best workflow practices themselves. This sentiment was echoed throughout the design, and was something the user tests also confirmed when asking the users about group processes and workflow.

Finally, it is not just the tools and resources that make a game at a game jam. If participants "go through a number of cycles of formal playtesting during the prototype stage and throughout production, the play-centric process allows them to learn the value of player feedback and practice the art of integrating feedback without compromising overall design goals" (Fullerton et al., 2006). The neat thing with this quote from this paper is that the student research project they embarked on focused on two very specific things: the play-centric methodology and a clear design goal of finding emotional experience in games. Finding good ways to communicate ideas and concepts like this allows for a good direction to be set, and a solid group understanding that fosters the creation of that game.

[8.0] Acknowledgements

I would like to express my appreciation to Simon Niedenthal for his continued patience and understanding during the course of the project. I would also like to thank Jörn Messeter and Jonas Löwgren for their drive for motivation and feedback to get me to make what is seen now.

Finally I would like to express gratitude for all the encouragement and endless supply of support given by my friends throughout my studies.

[9.0] References

Brandt, E., Messeter, J., 2004. Facilitating Collaboration Through Design Games, in: Proceedings of the Eighth Conference on Participatory Design: Artful Integration: Interweaving Media, Materials and Practices - Volume 1, PDC 04. ACM, New York, NY, USA, pp. 121–131.

Buxton, W., 2007. Sketching user experiences getting the design right and the right design. Elsevier/Morgan Kaufmann, Amsterdam; Boston.

Cooper, A., Reimann, R., Cronin, D., Cooper, A., 2007. About Face 3: The Essentials of Interaction Design. Wiley Pub., Indianapolis, IN. (p. 127)

Dearden, A., 2006. Designing as a conversation with digital materials. Design Studies 27, 399–421.

Fried, J., 2010. Why You Can't Work at Work | Jason Fried. Big Think. (Retrieved 2014-02-13) <u>http://bigthink.com/videos/why-you-cant-work-at-work</u>

Fullerton, T., Chen, J., Santiago, K., Nelson, E., Diamante, V., Meyers, A., Song, G., DeWeese, J., 2006. That Cloud Game: Dreaming (and Doing) Innovative Game Design, in: Proceedings of the 2006 ACM SIGGRAPH Symposium on Videogames, Sandbox '06. ACM, New York, NY, USA, pp. 51–59.

Fullerton, T., Hoffman, Steven S, Swain, C., Zimmerman, Eric, 2008. Game design workshop: a playcentric approach to creating innovative games. Morgan Kaufmann/Elsevier, Boston, USA. (p. 366)

Hecker, C., 2002. IGJ0. The 0th Annual Indie Game Jam. (Retrieved 2014-01-02) http://www.indiegamejam.com/igj0/

Kasurinen, J., Mirzaeifar, S., Nikula, U., 2013. Computer Science Students Making Games: A Study on Skill Gaps and Requirement, in: Proceedings of the 13th Koli Calling International Conference on Computing Education Research, Koli Calling '13. ACM, New York, NY, USA, pp. 33–41. Krippendorff, K., 2005. Semantic turn: new foundations for design. CRC, Boca Raton, Fla.; London.

Musil, J., Schweda, A., Winkler, D., Biffl, S., 2010. Synthesized Essence: What Game Jams Teach About Prototyping of New Software Products, in: Proceedings of the 32Nd ACM/IEEE International Conference on Software Engineering - Volume 2, ICSE '10. ACM, New York, NY, USA, pp. 183–186.

Salen, K., 2006. Beyond 100,000 Sprites: Game Design and Technology. Adobe Design Center Think Tank. San Jose, CA, USA. (Retrieved 2014-01-02) <u>http://solutionpartners.adobe.com/designcenter/thinktank/sprites/Beyond 100000 Sprites.p</u> <u>df</u>

Sharp, Helen & Rogers, Yvonne & Preece, Jenny. 2007. Interaction Design - beyond human-computer interaction. 2nd Edition. John Wiley & Sons inc. West Sussex. (pp. 530, 554)

Turner, T. aka j., Thomas, L., Owen, C., 2013. Living the Indie Life: Mapping Creative Teams in a 48 Hour Game Jam and Playing with Data, in: Proceedings of The 9th Australasian Conference on Interactive Entertainment: Matters of Life and Death, IE '13. ACM, New York, NY, USA, pp. 15:1–15:10.

[10.0] Appendix

[10.1] Technical Details

The prototype was made with Processing, a free and open source programming environment dependent and based off of Java. It is a platform which helps make prototypes and allowed for quick iteration and development. Java needs to be installed in order for the executables to work. The executables along with the source code is supplied in the links below. Processing is found and downloaded from http://www.processing.org/ Java is found and downloaded from http://java.com/getjava

The prototype and the source code for "mutter" is found and downloaded from http://evnh.com/mutter/

[10.2] How to use Mutter from the PDF Handbook

What is "px"?

The measurement of px stands for pixels. The reason for this is that the layout has fixed widths, meaning input is limited by the amount of pixels available.

Key mappings

Enter/Typing	Input text into chat
Ctrl	Toggle quick help screen
Scroll mouse wheel OR Up/Down Arrows	Scroll chat
Left/Right Arrows	Volume Control

Syncing

Mutter automatically sends out sync data every 8 seconds. If the user who set the timer is not online, data will not be synced. Clicking the global timer sets you as the global timer sync.

Setting a timer

Numbers must be in a two-digit format, for example 2 will be 02, 5 will be 05. The timer is valid only when the text fits the 335px limit and the format is followed. The limit shows on the left of the input when you type.

Timers are made by typing them at the end of a text

"Drawing some concept sketches 10m" "Coding 01h30m"

Add lowercase g to set a global (or group) timer

"Let's meet at the group room 03h20mg"

"Deadline for submission 02hg"

Manipulating timers

Change your timer by setting it again as needed. Or you may use the buttons found on the bottom right. You may only add to the timer a certain amount of times.

00m	Removes your current timer
30m	Sets a 30m break OR adds 30m to the current timer
04h	Sets a 04h snooze OR adds 04h to the current timer

The taskbar/program title

The idle count shows how many users have no timer active out of how many are online.

The mutter text will update to the global timer if there is a global timer active.

The mutter icon turns red after a global timer ends and an alarm sounds.

[10.3] Game Jam Checklist from the PDF handbook

These are suggestions and may help as a sanity check if not used as an actual checklist.

Computer/laptop access

- 1. Program installations (Game Creation tools, IDEs, Audio, Graphics)
- 2. Extensions, USB drives/storage, headphones, adapters, cables, fans
- 3. Tablets, microphones, extra equipment like controllers
- 4. Mobile devices, tablets or consoles as desired
- 5. Either your laptop's webcam or mobile phone for photos

Logistics

- 1. Place to work
 - a. Wireless internet access
 - b. Whiteboard access (or your own, or a large sketchbook)
 - c. Where to keep things safe
 - d. Access to printers and scanners (for posters or assets)
- 2. Place to rest
 - a. Sleeping materials (pillows, blanket, ear protection)
 - b. Toiletries (toothbrush, toothpaste, towel, deodorant)
 - c. Extra clothing
- 3. Food
 - a. Immediate snacks & drinks
 - b. Water, fruit
 - c. Know places to get food
- 4. Where/How to get help
 - a. Contact information, maps, support
 - b. Know the schedule/activities

Ideation/Prototyping kit (optional yet recommended)

- 1. Cardstock, cardboard
- 2. Papers, post-its
- 3. Drawing materials
- 4. Scissors, cutting materials
- 5. Tape, paper & universal glue
- 6. Dice
- 7. Playing cards
- 8. Tokens if desired, otherwise be creative

Game specifics

- 1. (Recommended) A minimum resolution of 1024x768
- 2. Player controls, mechanics and gameplay work
- 3. Game challenge and game flow are balanced

- 4. Menus for starting and ending the game, and a help screen if needed
- 5. If the game ends, you should be able to easily start/retry the game
- 6. Audio (especially music) add to the game and presentation experience
- 7. External user testing or feedback has been conducted

The deliverable (optimally a zipped file for ease of download)

- 1. A readme with a short description, credits/contacts, game help, links to dependencies
- 2. Executables and libraries
- 3. Source files
- 4. Screenshots (3 are optimal)
- 5. (Optional) Presentation, Gameplay Video, Gamelog, Design document, Documentation

[10.4] Physical voting methods from the PDF handbook

These are suggestions and may be used to inspire other methods.

Pulled from a hat - Participants write their votes or topics into a hat, which later is shaken up and drawn from.

- → May be used to decide on an idea randomly
- → May have multiple hats which represent the ideas, and can place one vote per participant into them
- → May be used to decide role or task assignments

Drawing straws - Varying lengths of paper, sticks, or straws are drawn from a person's hand

Rock, paper, scissors - A hand game that decides a winner through a number of rounds

Flipping a coin, rolling dice, guessing closest to a written number

- → May be used to decide on an unwanted/desired task
- → May be used to pick out a participant for a task

Ball throwing - Participants take turns passing the ball to each other once before getting a second turn, where the ball allows speech

→ Suitable for situations where many opinions and votes have to be heard with less criticism

Evaluation tables, pros/cons list

- → Suitable for situations where a whiteboard is available and a complicated decision needs to be voted for
- → Evaluation tables may be simplified so that certain features/suggestions may be evaluated as positive (+), neutral (o) or negative (-)

[10.5] Game Jam Practical Troubleshooting tips from the PDF handbook

A quick rundown of things to look out for during the course of the game jam.

Division of different tasks (not to be confused as roles, everyone may contribute) Coding/programming/version control Artwork/assets including character design Game Mechanics Level/map Design Music/sound and audio Quality Assurance/feedback and user testing Documentation/copywriter

When stuck or struggling

Check if there are other priorities, the detail may be minor and disregarded. Reassess if making it technically work is worth it, and how much time it will take. Attempt to divide the problem into smaller ones; among team members. Look for external feedback and accept critique to change the design. Look for creative alternative, easier solutions and playtest them. Take a short break and revisit the issue later. Napping, sleeping or eating may help.

Recommended habits

A shared folder to collaborate together on.

A form of version control (it may be Mercurial, or a script, or manually duplicating the project).

Put up a schedule where everyone can easily see it. Preferably on a whiteboard.

Shared working area, and a quiet (and dark) isolated area.

Save early and often, playtest early and often. Take breaks early and often. Rotate working hours so the work process and flow is more natural (example: when one coder sleeps another can take their place to avoid the need to merge code). For assets and graphics rotate working hours only to keep sanity and sleep deprivation at bay. Keep a positive vibe and remember the time dedicated to this is for fun.

Quick overview checklist

(Philosophical) What defines a game?
Do core mechanics work?
Does it relate to or challenge the theme?
Is this a game of quality?
Is the gameplay fun?
Does it have good game flow, challenge and feel?
Is the game compelling or intriguing?
Has there been playtesting in at least three different sessions?
How long does gameplay last?
Is it understandable, can players get it?
Does the game have music or audio?
Have you added juice?

[10.6] Notes on a Voting Interface

Structure of vote creation

Question/Situation Choice Choice Addvote (if applicable for clients) Novote (if applicable for vote)

People online should be seen by both server and clients. Know unique votes through signature or connections. Setting an end timer (will reveal the votes to the vote clients).

Statistics shown during the reveal

Percentages for each choice Vote numbers Novote numbers Total votes