

**THE REPUBLIC OF TURKEY
BAHÇEŞEHİR UNIVERSITY**

**GRADUATE SCHOOL OF SOCIAL SCIENCES
GAME DESIGN MASTER PROGRAM**

**A COMPARATIVE ANALYSIS ON DIFFERENCES
BETWEEN LONG TERM AND GAME JAM GAME
DEVELOPMENT CYCLES**

Master's Thesis

HADI ÇAĞDAŞ ERK

Supervisor: ASSOC. PROF. BARBAROS BOSTAN

İSTANBUL, 2018

**THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY**

**INSTITUTE OF SOCIAL SCIENCES
GAME DESIGN MASTERS PROGRAM**

Name of the thesis: A Comparative Analysis on Differences Between Long
Term and Game Jam Game Development Cycles

Name/Last Name of the Student: Hadi Çağdaş Erk

Date of the Defense of Thesis:

The thesis has been approved by the Graduate School of Social Sciences.

Asst. Prof. Burak Küntay
Graduate School Director
Signature

I certify that this thesis meets all the requirements as a thesis for the degree of
Master of Arts.

Asst. Prof. Güven Çatak
Program Coordinator
Signature

This is to certify that we have read this thesis and we find it fully adequate in
scope, quality and content, as a thesis for the degree of Master of Arts.

Examining Committee Members

Signature

Thesis Supervisor

-

Assoc. Prof., Barbaros Bostan

Member

-

Asst. Prof. Güven Çatak

Member

-

Assoc. Prof. Çetin Tüker

ACKNOWLEDGEMENTS

First, I would like to thank my parents for endless support and perseverance with everything I have done in which I could not do without them. Secondly, I thank our department head Guven Çatak for giving me this opportunity to be able to study game design master programme in Turkey by establishing one which was unprecedented at the time. Thirdly, I thank my supervisor Barbaros Bostan for patience and guidance he has shown for the path I have taken for this thesis.

I would like to specially thank Beste Bahçeçi for unyielding support through my thesis process, İlker Sevgen for pushing me to do things I needed to do even when I did not want to, Emre Sağlık for always having the right words to prove me wrong, Ece Egemen for positive encouragement and endless proposals of help, Emre İnal for keeping & remembering all the information for everything and sharing them with me, and Güven Çatak for being the only source of light in a pitch-black dungeon when everything seemed dire.

I would like to thank these wonderful people for creating games with me in game jams, for without them, this thesis would not be, Emre Sağlık, Ece Egemen, Emre İnal, Arda Çevik, Çağdaş Yıldırım, Yasemin Baran, Berkay Özercan, Kerem Oktay Bilgici, Onur Kek.

ABSTRACT

A COMPARATIVE ANALYSIS ON DIFFERENCES BETWEEN LONG TERM AND GAME JAM GAME DEVELOPMENT CYCLES

Hadi Çağdaş Erk

Game Design Graduate Program

Thesis Supervisor: Assoc. Prof Barbaros Bostan

May 2018, 50 pages

In the game development, there are four stages commonly used, which are analysis, design, code and testing. Game development life cycle takes example from software process models, enclosing the processes, methods and tools that assist the game development cycle. Game jam development cycles not only share similarities, but also differences with a long term game development life cycle. In this study, three games that developed during several game jams were explored in terms of game development stages to explain game jam development cycle. Similarities and differences were discussed using these stages. It is concluded that, especially the time restrictions is the main cause of differentiation of the game jam development cycle, however it is also possible to argue that game jam development cycle fits the game development life cycle as a compressed development process.

ÖZET

UZUN SÜRELİ VE GAME JAM OYUN GELİŞTİRME SÜREÇLERİNİN FARKLARI ÜZERİNE KARŞILAŞTIRMALI ANALİZ

Hadi Çağdaş Erk

Oyun Tasarımı Yüksek Lisans Programı

Tez Danışmanı: Doc. Dr. Barbaros Bostan

Mayıs 2018, 50 sayfa

Oyun geliştirmede, analiz, tasarım, kodlama ve test etme olmak üzere yaygın olarak kullanılan dört aşama vardır. Oyun geliştirme döngüsü, kendisine yardımcı olan süreçleri, yöntemleri ve araçları kapsayan yazılım süreci modellerinden örnek alır. Game Jam geliştirme döngüleri sadece benzerlikleri paylaşmaz, aynı zamanda uzun vadeli oyun geliştirme yaşam döngüleri ile farklılıkları da paylaşır. Bu çalışmada, Game Jam oyun geliştirme döngüsünü açıklamak için oyun geliştirme aşamaları açısından çeşitli Game Jam'lerde geliştirilen üç oyun incelenmiştir. Bu aşamalar kullanılarak benzerlikler ve farklılıklar tartışılmıştır. Özellikle zaman kısıtlamalarının Game Jam oyun geliştirme döngüsünün farklılaşmasının ana sebebi olduğu sonucuna varılmış, ancak Game Jam oyun geliştirme döngüsünün, oyun geliştirme döngüsüne hızlandırılmış bir geliştirme süreci olarak uyuşmakta olduğunu iddia etmek de mümkündür.

TABLE OF CONTENTS

1. GAME DEVELOPMENT CYCLES.....	1
1.1 PHASES AND TYPES OF GAME DEVELOPMENT.....	2
1.2 GAME JAMS	3
2. LITERATURE REVIEW	7
2.1 GAME DEVELOPMENT.....	7
2.2 GAME DEVELOPMENT IN GAME JAMS	9
2.3 GAME JAMS	10
3. DATA & METHOD.....	13
3.1 CASE STUDY 1: FRIENDSHIP.....	13
3.1.1 Goals	14
3.1.2 Audience.....	14
3.1.3 Space	14
3.1.4 Materials	15
3.1.5 Opening Ceremonies	15
3.1.6 Brainstorming.....	15
3.1.7 Protoyping.....	16
3.1.8 Building the Game.....	16
3.1.9 Testing and Refining the Game.....	18
3.1.10 Closing Ceremonies	18
3.1.11 Follow-Up.....	19

3.1.12	Audience.....	19
3.2	CASE STUDY 2: SHLACK!.....	20
3.2.1	Goals	21
3.2.2	Audience.....	21
3.2.3	Space	21
3.2.4	Materials	21
3.2.5	Opening Ceremonies	22
3.2.6	Brainstorming.....	22
3.2.7	Prototyping	22
3.2.8	Building the Game.....	22
3.2.9	Testing and Refining the Game.....	27
3.2.10	Closing Ceremonies	27
3.2.11	Follow-Up.....	27
3.2.12	Audience.....	28
3.3	CASE STUDY 3: BESTOW.....	29
3.3.1	Goals	30
3.3.2	Audience.....	30
3.3.3	Space	31
3.3.4	Materials	31
3.3.5	Opening Ceremonies	31
3.3.6	Brainstorming.....	31

3.3.7	Prototyping	31
3.3.8	Building the Game	32
3.3.9	Testing and Refining the Game.....	34
3.3.10	Closing Ceremonies	34
3.3.11	Follow-Up.....	34
3.3.12	Audience.....	35
4.	RESULTS	37
5.	DISCUSSION	40
6.	CONCLUSION	44
	REFERENCES	46
	<i>BOOKS</i>	46
	<i>PERIODICALS</i>	47
	APPENDICES.....	50
	APPENDIX 1: FRIENDSHIP CONTROLLER CODE	51
	APPENDIX 2: FRIENDSHIP ENEMY CODE	56
	APPENDIX 3: FRIENDSHIP CONTENT.....	58
	APPENDIX 4:SHLACK! CONTENT.....	59
	APPENDIX 5: BESTOW CONTENT.....	62

FIGURES

Figure 1.Wave Picture	17
Figure 2.Purple Enemy Ship	17
Figure 3.Player Ship	17
Figure 4.Enemy Ship and Obstacles	18
Figure 5.Character Parts.....	23
Figure 6.Hit Animation.....	24
Figure 7.Jump-Hit Animation	24
Figure 8.Protection PowerUP	25
Figure 9.Wall Hold/Slide	25
Figure 10.Level Design.....	26
Figure 11.Background Characters	27
Figure 12.Bestow with Placeholders	32
Figure 13.Player Actions.....	33
Figure 14.Environment	33
Figure 15.Timer, Combo Counter, Enemy Counter	34
Figure 16.GGJ Success Stories.....	38
Figure 17.Comparison Table.....	43

1. GAME DEVELOPMENT CYCLES

The development of complex software systems (e.g. Game development) in a heterogeneous environment needs a systematic integration of a variety of disciplines such as electrical, mechanical and software engineering (Musil, Schweda, Winkler, Biffel, 2010). Software development life cycle also known as software process models is a strategy that encloses the process, methods and tools which is used to develop the software. Typical software development life cycle contains four phases such as; analysis, design, code and testing (Ramadan, Widyani 2013). Analysis means to measure user requirements and create software creation requirements. In design part, those requirements translated into more detailed models, in coding period, code generation is done with respect to the models, in testing phase all elements are tested together to see if they work properly (Ramadan et al. 2013). Game development and design are one of the most inquiring software engineering processes. Since this process has multidimensional character, it becomes trickier. Game development process contains not only software engineering in addition to engineering it contains design, art, psychology and innovation (Pirker, Voll 2015). Because of that coordination and cooperation between different backgrounded people are necessary. Game industry has estimated exceed 30 billion dollars per year worldwide, creating games are the fastest growing segment of the entertainment market and it makes this area an excellent field for career advancement (Novak, 2011). This rapid growth results in launching game development programs by many universities. Games are often assumed successful if they create discussion and attract attention, quality of the game design or actual impact on players often remains unexplored (Mitgutsch, Alvarado, 2012). Games can evoke the players learning ability or can impact their lives (Mitgutsch et al., 2012). Games can be considered as constructive tools to foster learning, social change or advance understanding of social issues for players and also for game developers (Mitgutsch et al., 2012).

1.1 PHASES AND TYPES OF GAME DEVELOPMENT

There are several phases in the development process: concept, pre-production, prototype, production, alpha, beta, gold, and post-production. Each phase needs certain members of game development team and differs on specific objectives (Novak, 2011). Concept phase starts when an idea for a game is envisioned and ends when planning of the project is started. In this phase a designer, programmer, artist and a producer are needed from the game team (Novak, 2011). In this phase game subject is decided and the idea transferred into written form (Novak, 2011). In pre-production phase a proposal is prepared. In prototype phase the goal is to create a tangible game prototype, game engineers are often used for prototyping (Novak, 2011). Production phase is started when prototype has been approved, this phase is the longest phase of them all. In this part the game is actually being produced. In alpha phase, the game is ready to be played from start to finish. However, there can be a few gaps and art might not be finished but engineering wise game is completed, in this period game is polished and finalized (Novak, 2011). In beta period, Bugs are focused on, the aim is stabilizing the project and eliminate many bugs as possible before publishing the product to market (Novak, 2011). In gold phase, the game is sent to manufacturer and tested thrill and found acceptable. Finally, in post-production period or post release period, game developers can release and replace new versions of the game to improve original game or patches to fix software bugs (Novak, 2011).

Technology enables the game development to become more complex and producing more realistic games but it also highly increases cost of the game (Godoy, Barbosa, 2010). Producers wants to ensure a game development with minimum delays and mistakes. Agile methodologies can address these problems since they have simplicity, adaptiveness and being incremental. Agile methods are more iterative and more embrace change, they are more communicative and open to feedback (Ruonala, Yliopisto, 2016). It uses iterations to build more value into product (Ruonala et al, 2016). Game industry has a grown interest on agile methods for their nature of promoting collaboration, self-organizing cross-functional teams and frequent delivery of business value (Ruonala et al, 2016). In agile model of development testing out ideas early on is emphasized. It is chosen to remain in preproduction if the fun of the game is found (Ruonala et al, 2016). Some well-known examples of agile methods are Scrum and XP (eXtreme Programming) (Godoy et al.,

2010). Game design and development is such a multidisciplinary area, there is a discussion that planning and managing such complex tasks should not be performed in an agile method, but they should be performed with more classical detailed approaches (Aslan, Balci, 2015).

Rapid prototyping is an important technique in game design. The quickest way to build a game or test an idea is sketching them out, to test new ideas sketching or in this content rapid prototyping is a useful tool. Since developing games are becoming more complex and more expensive, this technique can save time and money, after an idea tested using rapid prototyping, if the game becomes successful in other words if the idea is proven to be successful it means the game can be continued to be worked on to make it ready to the market. Prototyping is one of the most powerful tools that developers have to explore possible designs. It allows to experiment with new tools and ideas cheaply and rapidly. There is a time constraint in rapid prototyping, in addition to time there can be number restrictions that how many people can work on a single prototype and also there can be a restriction about theme. One of the examples which rapid prototyping can be seen is game jams.

1.2 GAME JAMS

Game jam concept means that small groups of game enthusiasts come together to work on game design, starting with an idea and concept to a final project (Pirker et al. 2015). This process has a small window of time typically 24-96 hours (Scacchi, 2015) and a restricted location, theme and tools. These restrictions make the game development task more challenging. These gatherings have a purpose to bring different backgrounds together. All participants have different skill sets, mind sets, expertise and different levels of experience with game jams. In addition to that these participants usually never met before and they should be able to create a game together, should take responsibilities such as game design, art creation, testing, programming and level design (Pirker et al. 2015). These jams can have for-profit or non-profit sponsors. Usually these gatherings offer no tangible rewards, but they make participants focus on going “for the win” (Scacchi, 2015).

Game jams have key rules that outline general event dynamics:

- 1- The aim is to create small, experimental games rapidly to inject new ideas into game industry (Musil et al., 2010).
- 2- A general theme is decided before event that all created games have to share (Musil et al., 2010).
- 3- Everybody can participate these events that can contribute to the game's production (Musil et al., 2010).
- 4- Game finishing time limit is usually between 24-96 hours. (Scacchi, 2015)
- 5- Team formation before the event is discouraged (Fowler, Khosmood, Arya, Lai, 2013). The team size should remain between 2 to 5 participants (Musil et al., 2010).
- 6- The events encourage the development of games for any device and teams can choose their desired development platforms (Fowler et al, 2013).
- 7- There is a public presentation at the end of these events and the best games are awarded by an expert jury and audience (Musil et al., 2010).

Growth of the game jams since 2002 has been exponential (Annakaisa, 2015b). These events can take place anywhere, such as castles, planes, busses, trams and even on a 52-hour train journey. They all have different timeframes, some are more time dependent such as 0h Game Jam and some can be less time dependent such as VR Slow Jam (Annakaisa, 2015b).

Indie Game Jams are described as open game jam events (Goddard, Byrne, Floyd'Mueller, 2014). Game makers, independent game developers, students, hobbyists and industry professionals can join these events. Indie game jams are not facilitated for commercial outcomes. Participants encouraged to be creative, playfull and experiment on new concepts (Goddard et al., 2014). These events are hosted annually, most commonly on university campus. Some of these events are: Global Game Jam, Ludum Dare, Toronto Game jam, Nordic Game Jam, TIGJam, Train Jam, BUGJam. In addition to Indie Game Jams there are industry game jams and academic game jams.

Global Game Jam is an annual game design competition taking place in the last week of January that has a purpose to making a game in 48 hours (Ho, Tomitsch, Bednarz, 2014)(Sampugnaro, Mica, Fallica, Bonaiuto, Mingrino, 2014). Participants register themselves until Friday afternoon, a welcome keynote video is produced by organizers is

shown, that professional video game developers give tips and motivation talks (Sampugnaro et al., 2014). At the end of keynote video, participants are given a theme to incorporate into their games and 48-hour marathon is started at 5 P.M. In addition to the theme, achievements are given to the participants also referred as diversifiers for example “Both hands tied behind my back” this achievement means that this game should be played without using players hands (Macklin, Martin, Dikkers, 2012). In these hours, game developers explore their limits and skill sets, strengths. They have to create a strong communication between peer group members to allow to work with multidisciplinary skills and creativity. They work on programming, graphic production, interactive media design, writing narrative texts, creating sound effects or soundtracks. Limited time results in that finished games have short gameplays and simple game designs (Ho et al., 2014). Level of detail usually seen in games is not possible because of time constraints (Zook, Riedl, 2013). Recommended schedule includes a short planning and team creation period, followed by development time until 2 days later 3 P:M: on final day, remaining few hours are set aside for teams to present their games to each other (Macklin et al., 2012). Global game jam has a physical requirement unlike other jams, it has held in dozens of locations each year (Fowler, Khosmood, Arya, 2013). Global Game Jam format does not provide winners, prizes competition. The main aim of this event is to bring participants together to encourage creativity and collaboration. Global Game Jam was founded in 2008, it is a project of the International Game Developer’s Association between 2009-2012 (History., n.d.). After 2013 Global Game Jam is managed by Global Game Jam Inc. which is an international non-profit with an IRS charitable designation based in San Luis Obispo (History., n.d.). Nowadays the Global Game Jam is world’s greatest game jam event. The Global Game jams started with 1600 participants in 23 countries and produced 370 games (History., n.d.). In 2010 these numbers raised to 4300 participants and 900 finished games, in 2011, 6500 participants in 44 countries and 1500 games, in 2012, over 10000 participants in 46 countries and 2100 finished games, in 2013, 12000 participants in 63 countries, in 2014, 23000 participants in 72 countries 4292 games, in 2015, 28800 participants in 78 countries and in 2016 93 countries (History., n.d.). These gatherings have the potential to provide an effective and focused experience and participants gain important skills in prototyping and collaboration (Fowler et al, 2013). Since Global Gam Jam is brought into different societies, sometimes this results in different understandings

of the event (Yamane, 2013). Global Game Jam board is highly encouraging publishing the games produced during the event under Creative Commons rights that leaves property into the developer's hands, so they can work on the games after event and make them possible to release and upgrade (Sampugnaro et al., 2014). Distribution channels are mostly digital delivery platforms such as Steam for PC games, App Store, Google Play for IOS and Android platforms. Follow up survey results shows that participants of Global Game Jams continued to work on the games after event is concluded. In addition to that participants establish new companies following the Global Game Jams with the groups they met in the event ("Findings from the GGJ", 2014). In this angle Global Game Jams encourage entrepreneurial behavior.

In the early times for game jams the concept of game jam described as a mix of design and development strategies (Yamane, 2013). However, it can break down to eight different stages (Yamane, 2013):

- 1- New product development
- 2- Participatory design
- 3- Lightweight construction
- 4- Product value focused
- 5- Rapid experience prototyping
- 6- Aesthetics and technology
- 7- Concurrent development
- 8- Multidisciplinarity

Game jams continue to proliferate with addition to new events and locations. This growth has created new contexts such as game jams for research. In this perspective research and literature on game jams are rapidly increasing. Game jams are important subject to study for several reasons; their effect on the learning process of game developments, shaping the future development cultures, being a window to compressed development processes, they also have a giant role in the social aspect of game industry (Annakaisa, 2015b). In this thesis game jam approach is focused because, this field is active for over a decade and there are economically successful games that has oriented out from game jams.

2. LITERATURE REVIEW

2.1 GAME DEVELOPMENT

In this paper application of a methodology for game genre and player experience innovation called “play-centric design” described (Fullerton, Chen, Santiago, Nelson, Diamante, Meyers, Song, Deweese, 2006). For experimental project called Cloud which is made by students from the USC School of Cinema-Television’s Interactive Media Division, this method is used as a primary design methodology (Fullerton 2006). The application of this method is detailed as a potential model for innovating in the realm of overall player experience other than the traditional research focus on technology or subject matter innovation (Fullerton 2006).

The team identifies participation in game jams as ludic craft which is considered as a constructive form of play (Goddard et al., 2014). The properties of game are investigated on the playful vs. gameful continuum (Goddard et al., 2014). The team has derived a set of guidelines to facilitate ludic craft reflecting to their experiences as facilitators and participants of jams in indie, industry and academic (Goddard et al., 2014).

The team explains that to able to respond to the expectations of the industry and business school, marketing faculty started to use of group projects extensively in their curricula, however social loafing seems to be a common problem with this method (Aggarwal, L. O'Brien, 2008). In this paper writers identifies easy-to-implement project set-up factors to see the impact of reducing the social loafing and finds that while increased scope of the project and the team size increases the social loafing, having multiple peer evaluations during the progress of the projects reduces the social loafing (Aggarwal et al., 2008).

It is reviewed that agile methods are used in different degrees in game development (Ruonala et al, 2016). Writer states that using agile methods leads to better quality of games with prototyping, playtesting and feedback loop while also communication and ability to take responsibility of the team improved (Ruonala et al, 2016). It is also added that some difficulties observed from management issues, quality of code, multidisciplinary teams and lack of training in agile methods (Ruonala et al, 2016).

In this paper writer aims to investigate some alternative to few agile methodologies which address the problems found in the game development and suggest an approach to guide people who are new to this area (Godoy et al., 2010). Writer proposes a methodology named Game-Scrum and states that it has been applied in the development of a game for teaching software engineering (Godoy et al., 2010).

Writer explains that questions regarding the quality of serious games' formal conceptual design in relation to their purpose are mostly stay uncharted (Mitgutsch et al., 2012). Most of the time for serious games, designers' good intentions leave insufficiencies and incoherence untouched and they are mainly assessed in quality of the work only but not also for intention-based design (Mitgutsch et al., 2012). In this paper it is argued that analyzing a game's formal conceptual design, its elements and their relation to each other based on the game's purpose is a first step on assessing serious games (Mitgutsch et al., 2012). Writer outlines the background of the Serious Game Design Assessment Framework and by exemplifying its use, introduces a structure for examining purpose-based games (Mitgutsch et al., 2012). "Sweatshop" (2011) and "ICED" (2008) are used to apply SGDA Framework to demonstrate how to assess the formal conceptual design of serious games (Mitgutsch et al., 2012).

It is stated that game is a software with a purpose to provide entertainment (Ramadan et al. 2013). Writer indicated that adopting the software development cycle may not be enough since developers face different challenges in its life cycle (Ramadan et al. 2013). It is stated that game development uses an approach called game development life cycle to direct the game development (Ramadan et al. 2013). In this paper writer argues that none of the existing game development cycles addresses how to deliver a good quality game and presents a new game development cycle model and guidelines to successfully make a good quality game (Ramadan et al. 2013).

In this paper, a methodology called GAMED (diGital educAtional gaMe dEvelopment methoDology) is presented which consist of a body of methods, rules and is embedded within a digital educational game life cycle (Aslan et al, 2015). A framework is described for organizing of the phases, processes, work products, quality assurance activities and project management activities to develop, maintain and evolve a digital educational game from start to end (Aslan et al, 2015). It is stated that GAMED provides a modular

approach for overcoming the development complexity and guides the developers throughout the entire life cycle (Aslan et al, 2015).

In this chapter, writer concentrates on all phases of game development- including concept, pre-production, prototype, production, alpha, beta, gold and post-production. It is stated that management techniques used by game producers is also introduced and development of leadership skills cannot be overlooked (Novak, 2011). It also focuses on features and functions of game documentation (Novak, 2011).

2.2 GAME DEVELOPMENT IN GAME JAMS

In this work, team studied the compressed development process of conceptualizing a game and realizing the game in a working product at the 2013 Global Game Jam (Zook et al., 2013). A multi-step approach is proposed: (1) using survey instruments to first characterize the space of game design process and (2) following with more detailed studies of aspects of these processes. The results are described of the survey administered to 2013 GGJ participants about design inspirations and process of implementing ideas in a game over GGJ, how they improved their game and obstacles they faced along the way (Zook et al., 2013).

Team explains that game jams are a useful approach on short-term, end-user focused prototyping which take innovation momentum from concurrent development, time-boxing, cross-boundary collaboration and applying techniques for lightweight software development (Musil et al., 2010). They are a best-of combination of development strategies and flexible design which can be very useful in identifying key product values in considerably less time (Musil et al., 2010). Team suggest that transforming game jam concept into a X-jam could potentially enrich current software development processes (Musil et al., 2010). Based on their findings game jam concept may support new product development of systems with an engineering team in heterogeneous environments (Musil et al., 2010).

In this work it is stated that Global Gam Jam provides a unique opportunity to study time-constrained game development at a massive scale (Zook et al., 2013). The team administered a survey about their game development process: (a) participants use diverse

inspirations; (b) set goals for their personal benefit, the impact on game players and structure of the game system; (c) rarely employ traditional prototyping; and (d) evolve their games by scoping down many ideas, grounding a vague idea through implementation and iteratively expanding a simple core game (Zook et al., 2013).

2.3 GAME JAMS

It is stated that Global Game Jam can provide for research and teaching activities (Fowler, Khosmood, Arya, Lai 2013)[4,17]. GGJ-RC tends to welcome more projects and jammer interaction for the benefit of the research projects with a unified data-gathering mechanism (Fowler, Khosmood, Arya, Lai 2013) [4,17]. Team also discusses that some methods where predominantly extracurricular activity may augment classroom experience in different forms (Fowler, Khosmood, Arya, Lai 2013) [4,17]. The continued growth of GGJ helps game-based research and education by compiling the classroom theory and practical experience with constraints of GGJ (Fowler, Khosmood, Arya, Lai 2013) [4,17].

People of all kind of disciplines with different backgrounds and skills are included by game jams (Pirker et al. 2015). These participants work together in a given time to complete a game project, most of the time with people they have never met (Pirker et al. 2015). A small study concluded that if these participants fail to balance their composition it may affect their jam experience and result of the game project (Pirker et al. 2015). If they succeed to compose a balanced team for their project, the experience is much more satisfying, the outcome of the project is affected by this (Pirker et al. 2015). Insights are provided for team building based on their experiences in different scaled game jam events (Pirker et al. 2015).

In this work overview of game jams is provided which includes identifying problems of game developer students developing in the same physical community (Scacchi, 2015). This team identified different kinds of game creation organizational forms and conducted five field studies of different game development organizational forms (Scacchi, 2015). They directed their attention to problems in team-based software development projects like teamwork processes for computer game software engineering, game software requirements engineering, game software design, game making tools, game software

reuse, and global computer game software engineering (Scacchi, 2015). Studies and results showed that it may be possible to identify an outline eight observations or learnable lessons for future computer game jam research opportunities that may be useful for consideration by game development scholar, students or jam organizers (Scacchi, 2015).

Game jams are increasingly becoming a popular way of starting, working on and completing games (Toprak, 2014). Other than being very short events (generally between 8-48 hours), game jams allows people to collaborate, experiment and socialize (Toprak, 2014). In this paper writer uses game jams to make games that he explores and write about his doctoral studies as a game designer and a Ph.D. candidate (Toprak, 2014). Writer aims to understand the role of games jams in research and practice by making games and contribute to the discussion (Toprak, 2014).

In this paper a pilot study presented about connectivity of ideas at the Global Game Jam (Ho et al., 2014). After the game development completion, a survey presented to the participants (Ho et al., 2014). The authors investigated the idea formation of teams in a given time with data given by the participants (Ho et al., 2014). Authors presented a visualization demonstrating the connectivity of ideas and their findings for an inspiration network (Ho et al., 2014).

This study of Global Game Jam indicates how gathering people has become a new collective ritual for those who love video games (Sampugnaro et al., 2014). GGJ has become a place for sharing and practicing video game culture, even simple players are qualified as a participator provided by organizing committee (Sampugnaro et al., 2014). According to the analysis it is understood that complexity of motivations, individual interests and diversified character of the audience drive people to participate (Sampugnaro et al., 2014).

The writer states that concept of a game jam discussed in this paper (Annakaisa, 2015b). 20 research papers have been analyzed from 2006 to 2014 and it is presented that an advanced definition of a game jam proposed (Annakaisa, 2015b). "A game jam is an accelerated opportunistic game creation event where a game is created in a relatively short

timeframe exploring given design constraint(s) and end results shared publicly.” (Annakaisa, 2015b)

It is reported that Global Game Jam was introduced in Japan (Yamane, 2013). The writer mentions in promoting the benefits of new methods and technologies to the developers, educators and students throughout the world, introducing events like Global Game Jam plays an assisting role (Yamane, 2013). In their case before the Global Game Jam Japanese jam attendees were not well-acquainted with the practice of the participatory design or prototyping (Yamane, 2013).

In this paper selection of the picture of Ouroboros as the theme for Global Game Jam 2012 and discussion of the Global Game Jam 2012 Theme Committee has been explored (Annakaisa, 2015a). It is stated that GGJ Theme Committee discussed the topics of the role of GGJ and GGJ Theme and the ideas themselves (Annakaisa, 2015a). There was more than one reason for each idea to support or declining, and reasons for refuting the ideas consisted of the theme being too open, too hard, spurring one, dominant interpretation, or setting a wrong tone for the event, reasons for supporting the ideas consisted of the theme itself being experimental, promoting diversity, representing personally interesting and important domains of game design (Annakaisa, 2015a). The discussion circled around the role of the theme being a creative stimulus or a design constraint and how to cater to both experienced and inexperienced participants, it echoed differences in design paradigms: whether design activity is framed as problem-solving or reflection-in-action (Annakaisa, 2015a).

3. DATA & METHOD

In this section, three games will be explored in terms of their game development cycles. Mentioned games are friendSHIP, Shlack! and Bestow. They are all created in game jams, in particular; friendSHIP created in Global Game Jam 2015, Shlack! created in Windows10 Game Jam 2015 Istanbul and Bestow created in Global Game Jam 2016. All games developed under 48 hours and teams that have small number of people. All games will be explored under certain game development stages such as goals, audience, space, materials, opening ceremonies, brainstorming, prototyping, building the game, testing the game, refining the game, closing ceremonies, follow up, audience (Macklin et al., 2012).

3.1 CASE STUDY 1: FRIENDSHIP

Friendship is a local multiplayer co-op game, where each player can affect the ship's movement with using any digital mobile device connected to internet by giving directional orders to reach the end of the level.

Team Member	Assignment	Used Tool	Previous Experience
Programmer 1	Coding, Server - Client	Unity(first time)	Freelance writer at Oyungezer, 2 years of software engineering
Programmer 2	Coding, Level Design, Sounds	Unity(first time), www.bfxr.net(first time)	Master's degree on software engineering

Visual Designer	Game Art	Pyxel Edit(first time)	Visual communication design for 5 years
Game Designer	Coding, Game Design	Unity(experienced)	Game developer/designer for 1.5 years

3.1.1 Goals

Our goal was to integrate with the game development community as well as challenging ourselves to complete a game in given time, with the people we may or may not have met before. Also, it was expected to learn and improve our knowledge about game development.

3.1.2 Audience

The audience in Global Game Jam 2015 had a great energy with people both experienced and inexperienced. Mostly young adults with different backgrounds on game design, arts, programming, sound design etc.. Global Game Jam community motivated us to make something that has high energy and we can play together as a whole.

3.1.3 Space

The Global Game Jam2015 Istanbul, that is participated took place in Bahçeşehir University Communication Faculty Game Design Laboratory called BUG (Bahçeşehir University Game Lab) Kitchen as well as Bahçeşehir University VR1 Laboratory located next to BUG Kitchen. Since one of the team members was already a student in Bahçeşehir Game Design Department, working in that space was not a foreign experience.

3.1.4 Materials

After deciding the team, although team programmer was not familiar with the “Unity” game engine, we considered using that game engine to both to learn and to explore its features. An Amazon server is used to handle the player traffic in the game. Pyxel Edit is used for creating the graphical user interface and game art in pixel art style.

3.1.5 Opening Ceremonies

In opening ceremonies the theme of the Global Game Jam was presented as “What do we do now?”. Usually this period is used to share your ideas and finding team mates to create the game idea. In this scenario the team was already established ahead of time, so this period is also used for brainstorming. The team consisted of 2 programmers, 1 game designer and 1 artist. Although we put names of the jobs of the team members, in some point everybody helped each other with different tasks that they are not responsible of.

3.1.6 Brainstorming

Since the theme was “What do we do now?” and the audience had high energy, our desire was to create multiplayer game that all the jam participants can enjoy. While thinking about the mechanics of the game we considered that all the people in the jam should be able to get involved. After some brainstorming “twitch plays pokemon” event came to the mind. Twitch plays pokemon is an event where game livestream site called twitch allowed its viewers to give commands to the game Pokemon Generation 1 from chat window and for each given time most voted action was chosen to play in the game. Therefore, it was decided to a mechanic that character moved with the actions of people who are playing the game. Theme of the game inspiration was come from the song that “What do we do with a drunken sailor?” a well-known sailor song with a high energy and funny lyrics. Friend concept is close to the theme and FriendSHIP name is popped out into the heads since “we” in the theme can mean friends (which the team consisted of) and our sailor song dictates a ship to be used in the game as a main character.

3.1.7 Prototyping

After mechanic of the game was decided both art style and programming aspects of the project started to form. Since there is a time constraint for the game jams, it sometimes much easier to create a pixel art game as in art form to be able to complete in the given time. Given these ideas, to make a prototype boxes and circles were used. First, the thought of a team fight game came in to the mind and started to form. After programming the core mechanic with Unity Game Engine, team realized that server-side problems may occur if a team fight mechanic introduced in the game. Also, since as it foreseen by the team, participants may not be get used to the playing style if team fight mechanic be a frustrating factor. Team tried to put team fight mechanic in to the prototype and tested. Result to this was the fun factor of the mechanic was not high as the expectation. Prototyping also showed us even in a constrained time it helps to try out the product's features before going into a final form. Another mechanic choice was a rowboat style control which has two inputs and turns the ship either right or left. With this control more complex inputs were introduced. Since angular movement may result in frustration among participants team decided to eliminate this feature. For spamming reasons, team decided that to put an input count limit for each player. After some playtesting team realized that spamming is much more fun compared to limited input availability.

3.1.8 Building the Game

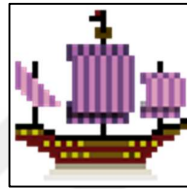
After designing what to do, team started programming, creating the art, level design and sound design. As can be seen in Appendix 1 Programming of the game started with programming the movement of the ship. Team decided to have 4 (north, west, south, east) directions as inputs and 8 directions (north, northwest, west, southwest, south, southeast, east, northeast) as outputs to the movement. The inputs taken from the server transferred to the Unity in which the movement of the ship decided. Also, some sort of sea waves integrated into the game (figure 1). Ship lose control when entered wave areas according to the direction of the wave.

Figure 1.Wave Picture



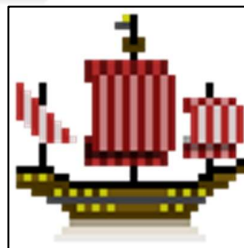
An enemy type also integrated to the game which can be found in Appendix 2. The enemy is a ship which makes different movements as player tries to get past the enemy (figure 2).

Figure 2.Purple Enemy Ship



For the art design, ships used in 15th Century was an inspiration for the team. The pixel art type of these kind of ships designed in Pyxel Edit to be used in the game (figure 3).

Figure 3.Player Ship



Level design of the game kept simple as possible since the team could not foresee the feedback from a huge audience playing at the same time. Complexity in levels may have been resulted in deadlock situations. Also, simple enemy ships and obstructions added for increasing multiplicity in the game (figure 4).

Figure 4. Enemy Ship and Obstacles



Sound design is produced by <https://www.bfxr.net/>'s design tool which is a really fast prototyping tool. Also, music from the song “Drunken Sailor” by Irish Rovers used as a background music.

3.1.9 Testing and Refining the Game

Since the team wanted to keep the first impressions natural no outside testing done. In the process of the game development phase team members tested the game for different features mentioned in prototyping section above. Feedbacks from each team member considered separately and given the right fixes that are decided together. No further testing is done due to time limitations and server problems to the last hour of the jam.

3.1.10 Closing Ceremonies

In this period, participants first met with our game, nearly a hundred people played the game together. Reactions of the players are amazing, laughing and clapping participants were in majority. everybody really enjoyed playing friendSHIP. In this time, we realize that decisions we made about the game is seen right for a game jam environment, level design complexity and control strategy of the player ship was in the right notch. An offer is made to us to use our game in team building exercises in a company training. This offer shows that game has an impact and has promise.

3.1.11 Follow-Up

Team considered to monetize this game, but due to busy schedules of the team members this idea had yet to be implemented to this day.

3.1.12 Audience

To this day the game still on the itch.io website in global game jam 2015 games. Anyone who can rent/build a server and configure the settings can play the game.

Subject	Time Spent
Brainstorming (4 person)	2 Hours
Mechanics programming (2 person)	20 hours
Pixel art (1 person)	15 hours
Level design(1 person)	6 hours
Server-side programming (1 person)	7 hours
User Interface (1 person)	2 hours
Sound, Music (1 person)	2 hours
Testing (4 person)	3 hours

For friendship, the team consisted of two programmers, a visual designer and a game designer. All the team members had some level of understanding and experience of coding. The team members knew each other before but it was the first time they worked all together on a project.

Team sacrificed from visual design to make time for the coding and game design.

After deciding what to do, base mechanics of the game implemented by the programmers at the first 20 hours since programmers never used unity before and wanted to try it. In the meantime, visual designer started to make pixel art models of the game assets. Game designer started with level design sketches for the game. For the second part of the jam, server need to be implemented to play the game with many people therefore, programmers focused on the server-side implementation while visual designer draw the user interface for the game and game designer implemented in to the game. Game designer also implemented the sound and music of the game in the last 2 hours while testing was done for the game.

3.2 CASE STUDY 2: SHLACK!

Shlack! is a local multiplayer single screen deathmatch game where chubby mustached men slap each other with wet towels in a Turkish bath. It supports two to four concurrent players with controller inputs.

Team Member	Assignment	Used Tool	Previous Experience
Programmer	Coding	Unity(experienced)	Experienced jammer, programmer
Visual Designer 1	Game Art, Animations	Adobe Photoshop(experienced)	Visual design

Visual Designer 2	Game Art, Level Design, User Interface	Adobe Photoshop, Unity	Cinematographer
Game Designer	Unity, Game Design	Unity(experienced)	Game developer/designer for 2.5 years

3.2.1 Goals

As four game design master students of Bahçeşehir University our main goal was to create a local multiplayer game that it can also be used as a term project. Since it was a competitive Windows 10 Game Jam, our motivation also included the prizes that are promised.

3.2.2 Audience

Even though it was a competitive event, the audience was very friendly and energetic. Being a competitive game jam, audience consisted of more experienced participants compared to other game jams.

3.2.3 Space

As one of the Global Jam Istanbul places, BUG Kitchen in Bahçeşehir University Communication Faculty building was chosen as the host of the Windows10 Game Jam 2015. Since all team members studied there, space was familiar to work in.

3.2.4 Materials

Most of the team members had an understanding how Unity Game Engine works. Unity is mostly used to build the game. Artists were fluent with Adobe Photoshop and 2D game art.

3.2.5 Opening Ceremonies

As its name suggests Microsoft made an opening statement and gave some information about Windows10. There was no theme, only constraint was the games had to be made for Windows10 Operating Software. The team was already formed, therefore brainstorming started.

3.2.6 Brainstorming

As the team joined Windows10 Game Jam with a goal to make a local multiplayer deathmatch game all team members had some idea about what kind of mechanics to be added as a feature in to the game. What the team needed was a theme to put together. Normally in a jam this phase can take some time to be able to come up with a theme/idea that all team members agree. Surprisingly team agreed quickly upon on the usage of a local concept. “Turkish Hamam” (Turkish Bath) concept was a strong one which everyone agreed on. After this brainstorming about the game mechanics started. Local multiplayer games like Overcooked, Crawl, TowerFall Ascension, Gang Beasts and Nidhogg were inspiration for all team members. An idea come up with mustached guys in a Turkish Hamam hitting each other with loincloths (a kind of towel to cover your lower body in Hamams) and it seemed like fun and team agreed upon it to start developing the game.

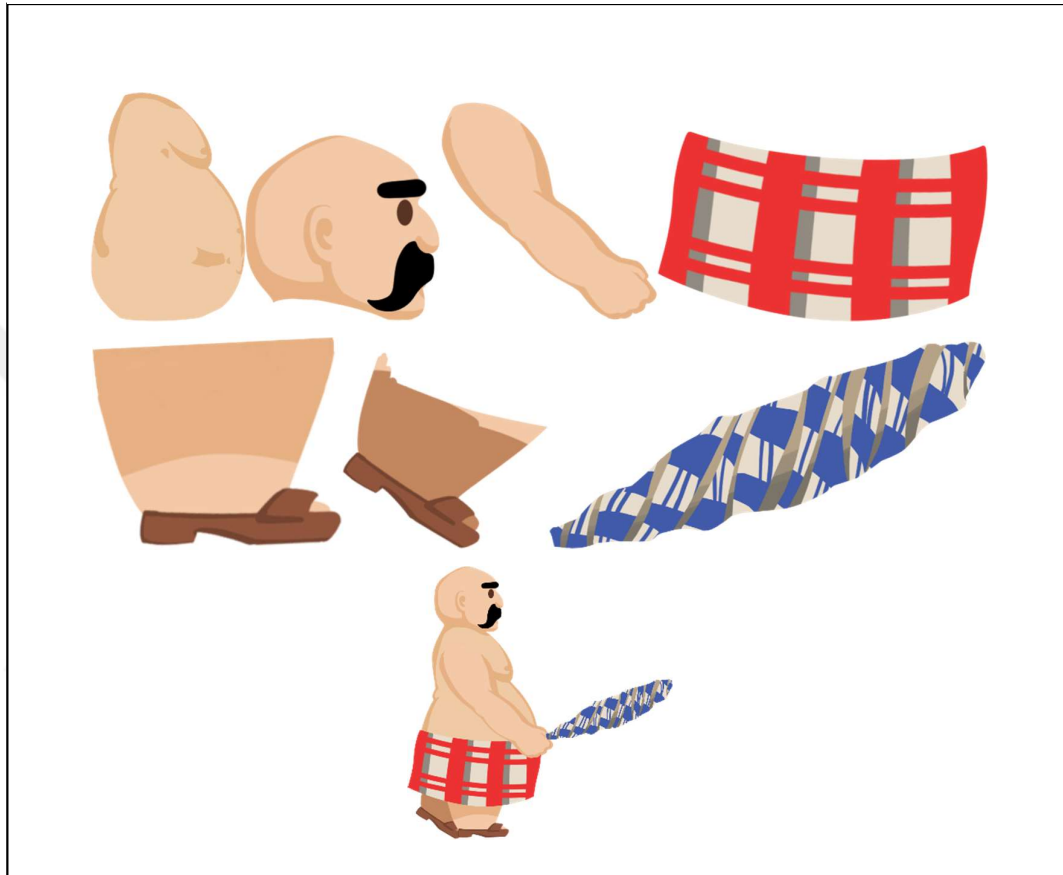
3.2.7 Prototyping

After mechanics and theme decided by the group, programmer started to code the main controllers and the scene. Artist started with concept art of a guy the idea represents and concepts of level design started to come out. As soon as main controllers finished team started testing the jump and hit mechanics with placeholder boxes and circles to as a first prototype. The mechanics tuned for making the game more fun centric and tested couple of times.

3.2.8 Building the Game

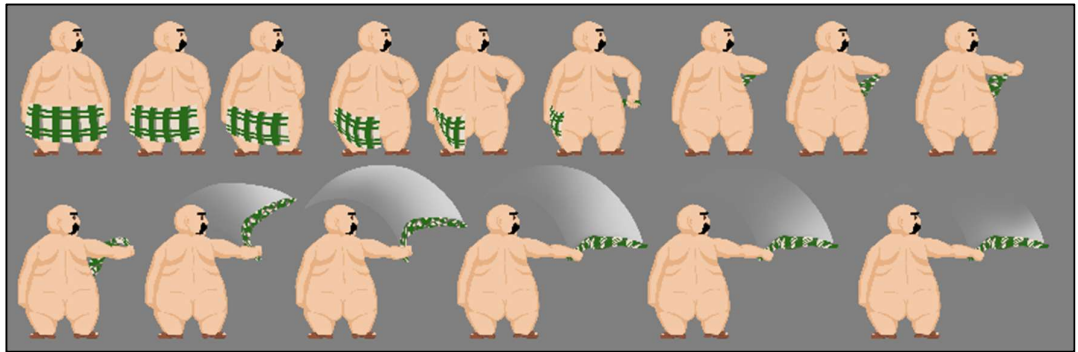
When creating the character, to be able to manipulate in animation 7 different parts produced which made easier implementation to Unity (figure 5).

Figure 5.Character Parts



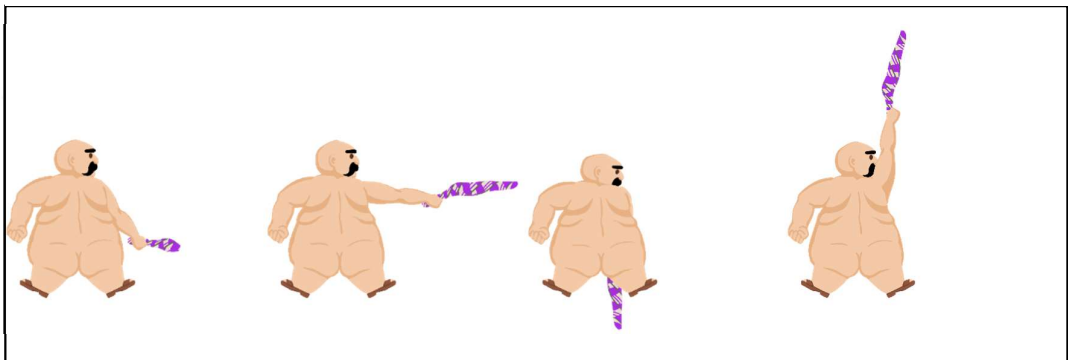
Furthermore, hit animation was created using these parts and some added sprites (figure 6). Team thought that while hitting to the other players, as a downside, the guy would need to expose himself which would make a funny figure that would entertain the players while playing the game.

Figure 6.Hit Animation



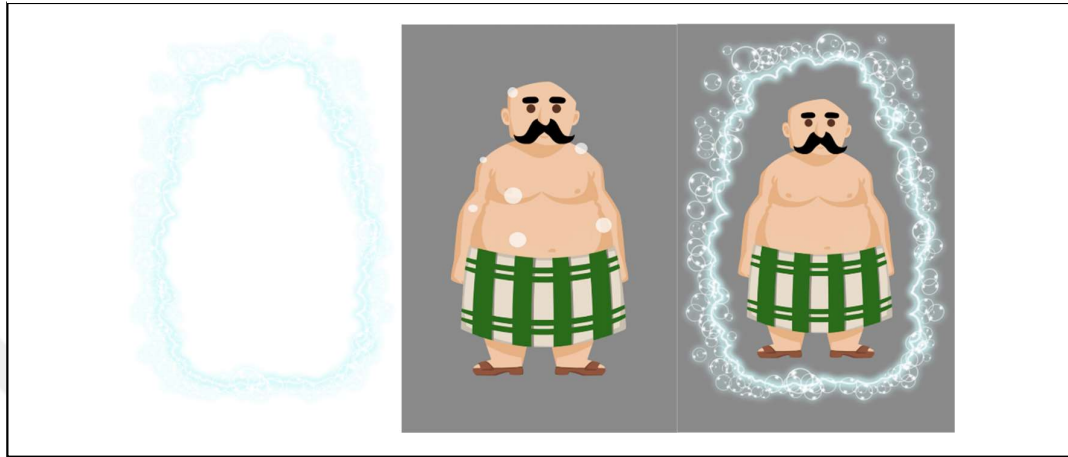
Since game includes jumping and hitting mechanics team decided that player should be able to move and hit 4 different directions (up, down, left, right) (figure 7).

Figure 7.Jump-Hit Animation



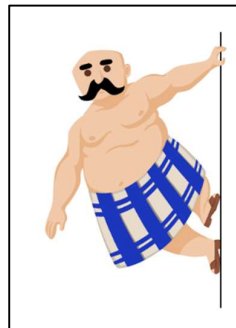
Protection power up added to the game to make it more challenging by putting the power up to an open area so that players interact and take risks to be able to get the power up since there are only 10 lives before you lose/win the game (figure 8).

Figure 8. Protection PowerUP



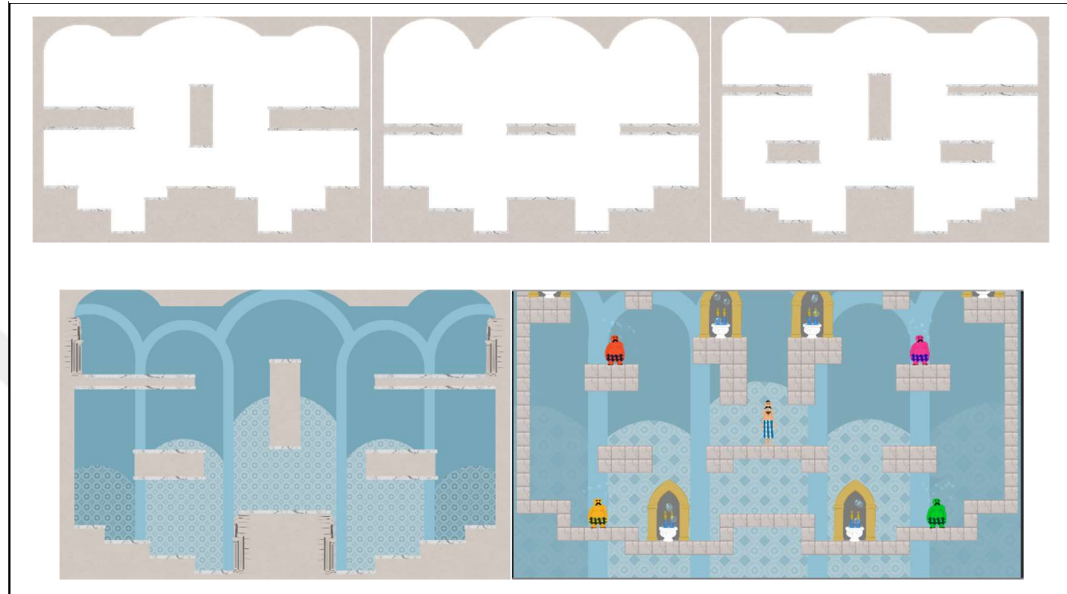
Wall hold/slide and wall jump features added to the game to make the characters more mobile and increase the movement styles thereby allowing players to be more versatile in the game (figure 9).

Figure 9. Wall Hold/Slide



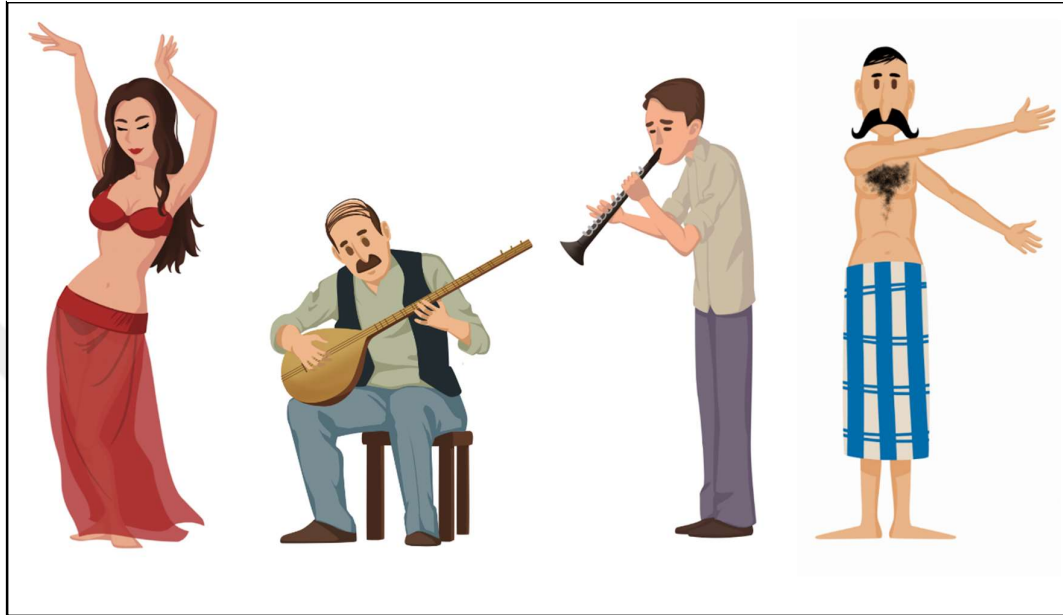
For level design, couple of different level styles tested with two, three and four players. Level had to be able to provide some walls, open areas, spawn points, some edges for frustrating fights and movements, at the same time it should fit in a window and players need to be able to distinguish their characters. After testing some radical changes to the level design, team chose more conservative design types of 2D deathmatch games (figure 10).

Figure 10.Level Design



To emphasize local hamam content, characters of different kind put in the scene. A belly dancer, a person who plays a Turkish instrument with three double strings, a clarinet player and a hamam attendant added to the background scene with a background Turkish music especially used in hamams in Turkish movies. These helped to show some oriental and authentic tones (figure 11).

Figure 11. Background Characters



3.2.9 Testing and Refining the Game

Since the game is a local multiplayer experience, a lot of tests done between each iteration of the game in given time. With each test and given feedback, game developed further and better. For the last couple of hours, team fine-tuned, cleaned some bugs and glitches.

3.2.10 Closing Ceremonies

The reaction from audience was great when the game presented and played by the participants. Shlack! chosen as the winner of the event.

3.2.11 Follow-Up

After the event, improvements, fine-tuning and bug cleaning made for the game since it was also a term project for the team. Later, Shlack! presented in the JOIN Local Multiplayer Summit 2015 in Berlin. The game got great feedbacks from this event. Unfortunately, due to the busy schedules of the team members development of the game did not continue.

3.2.12 Audience

The game can be found in itch.io website or its own website <http://www.shlackthegame.com/>.

Subject	Time Spent
Brainstorming (4 person)	2 Hours
Programming (1 person)	27 hours
Concept art (1 person)	2 hours
Character design (1 person)	15 hours
Level design(2 person)	5 hours
Animation (1 person)	4 hours
User Interface (1 person)	2 hours
Sound, Music (1 person)	2 hours
Testing (4 person)	5 hours

For Shlack!, the team consisted of a programmer, visual designer, visual/game designer and a game designer. None of the team members worked together before but they studied together.

Since the team had an experienced programmer with controllers in unity and experienced visual designer in 2D team did not sacrifice much in quality for time compared to the other games.

For the first 2-3 hours, base of the code established to check if the game will work with intended mechanics and features. After integrating the movement in the game, for the first 24 hours, features added to the game. For the last 24 hours, programmer cleaned bugs and adjusted the game. As for the visual side, the idea is developed for the game and started to draw concept arts. The concept art was decided to go forward by the team after first 1-2 hours. The visual artist then started to draw characters in game for the first day. The animations and the background drawings were done in the last 24 hours. Visual/Game designer started with UI design of the game and helped the visual designer when needed, also helped with the level design. As for the game designer, level design concepts created in first couple of hours to implement in the base game for testing. Creating the levels designed in unity and testing done iteratively until the end after the programmer finished with the base in 2-3 hours.

3.3 CASE STUDY 3: BESTOW

Bestow is a two-person cooperative survival game where a person fights monsters with a sword and a shield controlled by a keyboard and the other person tries to match physical movements given by the game verified by a Microsoft Kinect to keep the other person's shield up to stay alive.

Team Member	Assignment	Used Tool	Previous Experience
Programmer 1	Coding	Unity	Freelance writer at Oyungezer, 4 years of software engineering

Programmer 2	Coding	Unity	Master's degree on software engineering, PhD Candidate
Visual Designer 1	Game Art, User Interface	Photoshop	Art director, illustrator
Visual Designer 2	Background Visuals	Photoshop	Visual communication designer
Sound Designer	Music, SFX	*	Sound Artist
Game Designer	Game Design, Animations	Unity	Game developer/designer for 2.5 years

3.3.1 Goals

Our goal was to make a game with a diversifier of Global Game Jam 2016 while meeting new people to work and have fun with.

3.3.2 Audience

The headcount was higher than the other Global Game Jams hosted there. All the participants were excited to be there and started bonding even before the icebreaker event.

3.3.3 Space

BUG Kitchen, VR1 Lab and BUG Lab of Bahçeşehir University Communication Faculty building in Karaköy was chosen for the event. Of the team members, one was already studying there where the others joined a game jam before at the same space therefore, space was not a new challenge for the team.

3.3.4 Materials

Unity Game Engine was used for creating the game. Microsoft Kinect was used for capturing and verifying the second player's movements. The art of the game was done with Adobe Photoshop.

3.3.5 Opening Ceremonies

The theme for the Global Game Jam 2016 was announced as "Rituals". As mentioned in the goals section team explained that they wanted to make a game with a diversifier and that would be a Microsoft Kinect. After the ceremonies 2 artists and a sound designer joined to the team.

3.3.6 Brainstorming

Since the team wanted to use Kinect on their game, the theme "Ritual" needed to have a physical impact on the game. A story was thought about praying to the gods, so the gods would empower the people with their might. For the player one a simple villager is thought to be empowered and for the player two who makes rituals to be able to defend their home. Team decided to put an art style of Greek mythology for the character on top of these things.

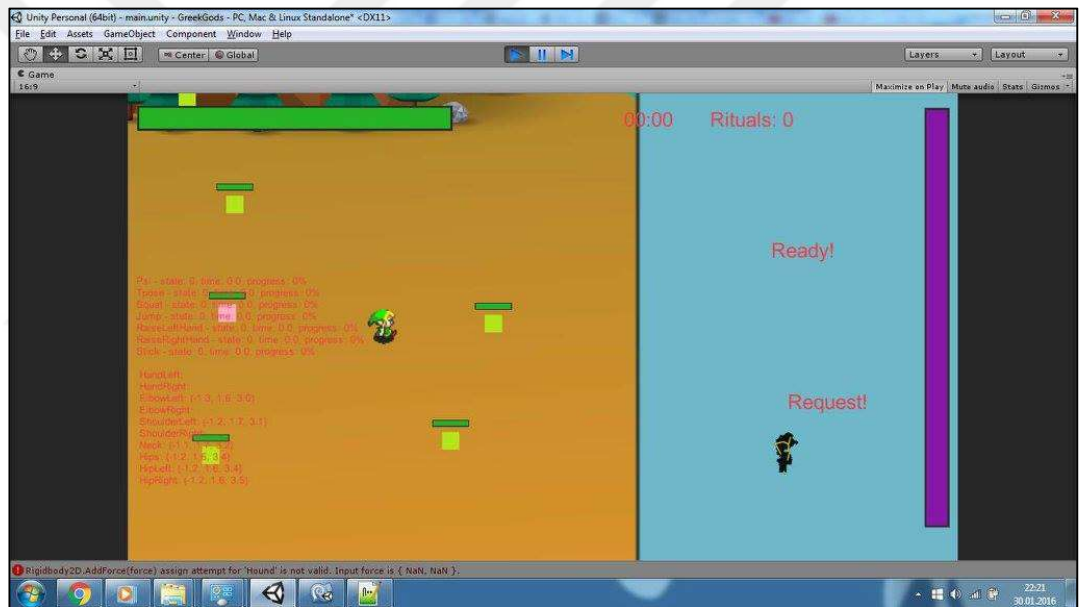
3.3.7 Prototyping

When the concept decided and agreed upon, programmer started to work on the game's core features. As soon as core features was handled, working on the connection with Kinect started. At the same time art for the background and the characters started and first prototype was made, then the team started to test it.

3.3.8 Building the Game

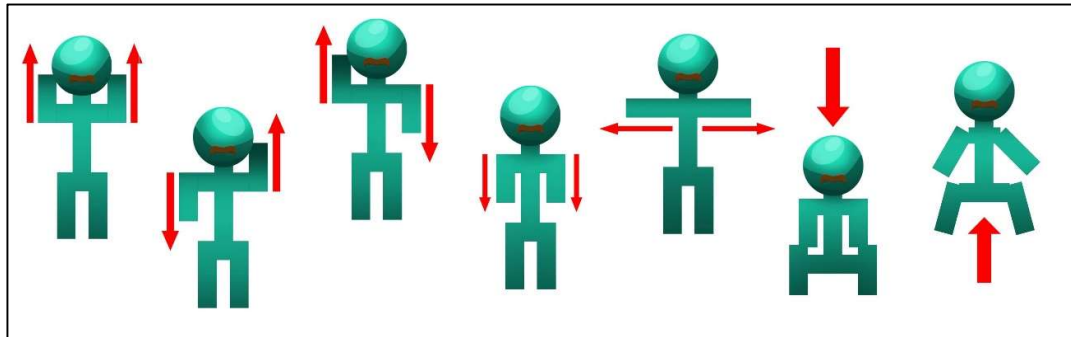
First, core features of the game implemented into the it. Movement for the character to all directions were allowed for easier controls. Health bars for character and enemies and hit action implemented to test the system in its basic form. The character Link from Legend of Zelda used as a placeholder as well as boxes and different shapes to be able to test the game mechanics (figure 12).

Figure 12. Bestow with Placeholders



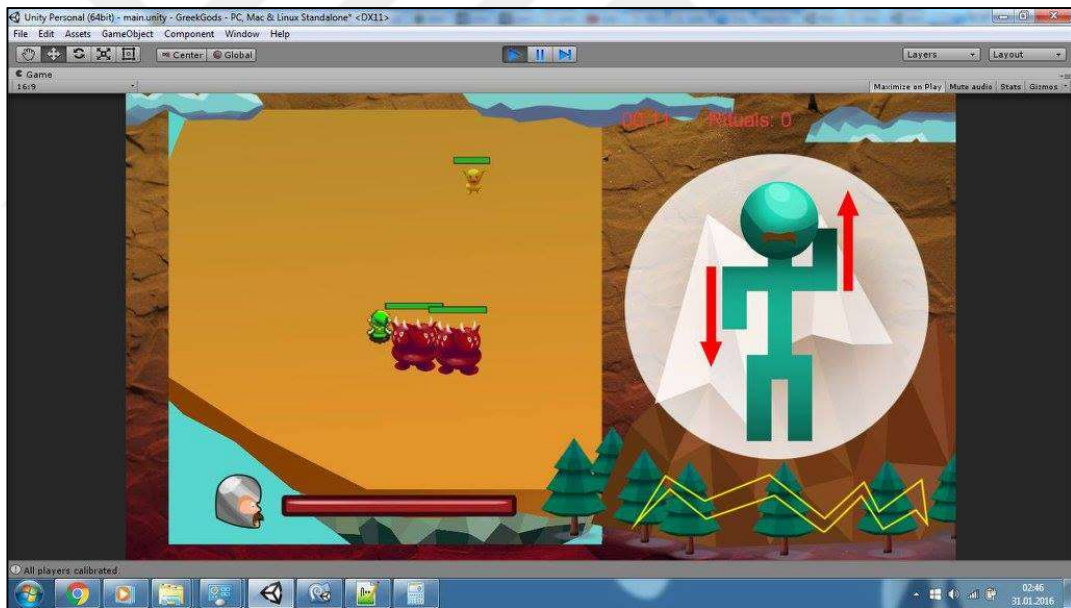
As the theme suggested “Ritual” was decided to be different arm and leg movements. For the sake of the jam, difficulty of actions of player two were chosen to be easy to moderate (figure 13).

Figure 13.Player Actions



In the meantime, environment was decided to be a 3D low-poly like 2D visuals and implemented into the game with the movement figures (figure 14).

Figure 14.Environment



A timer (small white bubble in figure 15) added to the game to see how long the players survived. Combo counter (small yellow bubble in figure 15) added to be able to see the consecutive successful movements. An enemy counter (small red bubble in figure 15) added to be able to count the enemies in the area so that if counter reaches 65, the players would lose. Also, the finished character sprite and the protective shield added to the game. As last, animations, music and sound effects were added to the game (figure 15).

Figure 15.Timer, Combo Counter, Enemy Counter



3.3.9 Testing and Refining the Game

As soon as Kinect connected to the game team tested the movements and core mechanics. Some participants outside of the team asked the try the game and give feedbacks. Iterations done by the given feedbacks. Kinect had some difficulties reading the T human shape in the game, therefore the odds of T shaped actions reduced to improve the player experience.

3.3.10 Closing Ceremonies

Our game was the only Kinect game done within the Global Game Jam 2016 Istanbul. People laughed and giggled when they see the player two making weird movements. Since it was not a competition no grading done yet the feedback from the audience was good. The game was satisfactory for the team.

3.3.11 Follow-Up

There was no follow-up for this game because it was created just for the event and team had no reason to chase it afterwards.

3.3.12 Audience

Game created in this jam is the website itch.io. Anyone who has a Kinect and a computer to run can download and enjoy the game.

Subject	Time Spent
Brainstorming (4 person)	4 Hours
Programming (2 person)	9 hours
Concept art (1 person)	2 hours
Character design (1 person)	8 hours
Level design(2 person)	4 hours
Animation (1 person)	3 hours
User Interface (1 person)	2 hours
Sound, Music (1 person)	3 hours
Testing (6 person)	4 hours

For Bestow, the team consisted of two programmers two visual designers, a sound designer and a game designer. Only 3 of the team members worked together before.

Team thought that game would appear much better with 3D modelled, at the same time they realized that they did not have the resources or the time to complete that achievement for the game. That is why 2D visuals which seemed like a low poly 3D models implemented in the game.

For the first 6 hours programmers implemented the Kinect to Unity and after that started to put game mechanics in it. At the same time one of the visual designers started to make background assets and the other made the characters. Sound design was not ready 5-6 hours before finish since the game was not fully understood by the team to make a sound design decision. In the meantime, mechanics and level design created by the game designer for the first 16 hours. For the last 8 hours team members polished iterated and made the game better.

4. RESULTS

In this chapter, three different games mentioned in the Data & Method section will be summarized in order to explain Game Jam Game Development Cycle processes. It can be said that all games' main goal was self-improvement, along with creating a fun game for the audience. For the audience, it is explained as energetic, purposeful and determined. All the games created in the same space which is a university campus lab that is allocated temporarily for these events. Mentioned games used Unity as their game engine and Adobe Photoshop for making the visual art. For the opening ceremonies two of the games used given theme where the other one had no theme but constraint as an operation system. All the games used brainstorming phase fully to create fun experiences. This phase also used in deciding some mechanics of the game while discussing about the story behind the game along with the sound visual design. All the games processed similar kind of prototyping phase where their game was tested and iterated through the feedbacks. The longest phase for the three games was building the game. All the games went through with programming their game, creating their art with the decided concept, testing core features after implementing them and designing sound and music in short periods. While one of the games kept their game as a secret when it came testing, other two games went on and tested with different people on other teams. This resulted in having tested the game to a different audience helped refining the game. All teams reported the audience was fond of their game. Although one had an offer to be played in another place and second one won the competition in the jam. None of the jam games mentioned resulted in as a follow-up with their products. Even though they did not follow up their projects, to this day all the projects can still be downloadable and playable.

In GGJ criteria, figure 16 shows successful games that were created during game jams (GGJ Success Stories., n.d.). Common factors between these games are first, follow up work after game jams. Originality is another key factor, game ideas are unique in almost every game. In game jams time is another suppressing factor therefore graphic works on games are usually simple and colourful. Game plays are easy to learn, it can be seen that fun factor are more emphasized than polishing the game.

Figure 16.GGJ Success Stories

Name of the Game	Status
Resonance	Greenlit on Steam
A Normal Lost Phone	Released for IOS, Android and Steam
Pentagrab	BAFTA Ones to Watch Award Nominee 2017
Reflection at Sunset	Was shown at Different Games Toronto, Playcrafting Expo, and Play NYC
Curvish	Won the grand prize in the UK-wide Transfuzer game development program, greenlit on Steam.
Home Improvisation	Winner Global Game Jam 2015 Atlanta Location, Winner IndieBits Festival 2015 Best Overall Game
Hide-Dot(·)-Seek	Converted into a successful online game with an active player base.
Blockpocalypse	Released on Steam
Marooners	Available on Steam, got featured in a Humble Bundle and is soon to be released on PS4 and xbox one.

In literature, in order to create a successful game in game jams some factors are emphasised. First a balanced team is a key factor (Pirker et al. 2015), in other words distributing game work between multidisciplinary team members should be close to equal to have a pleasant game jam experience which would affect the quality of the game product. Communication between peer group members is another key factor to create a successful game (Ho et al., 2014). Since game jams have time constraint, fast and successful communication plays a key role on sharing ideas and creating an end result. In addition to that time constraint also affects the scope of the planned game (Zook, Riedl, 2013. Groups should keep level of detail in an optimal point to able to finish their game

in constrained time. If one of these factors are not met in a team, resulted game could be affected severely. Lack of communication or lack of equality in work distribution can be a reason to not be able to finish the game in a game jam. Also introducing a lot of mechanics and features in a game could exceed the desired range of the scope which would result in an unfinished game at the end of the game jam.



5. DISCUSSION

As mentioned in introduction, game development has many phases. They generally take long times and large teams, mostly resulted in highly costly projects. In contrast game jam game development cycle has small teams and a constrained small-time window. In this chapter a discussion about differences between game development cycle and game jam game development cycle will be made.

In game development process main phases are concept, pre-production, prototype, production, alpha, beta, gold, and post-production. In game jam rapid game development cycle has phases as; goals, audience, space, materials, opening ceremonies, brainstorming, prototyping, building the game, testing and refining the game, closing ceremonies, follow up and audience.

In the sake of argument game jam rapid game development's stages; goals, audience, space, materials, opening ceremonies and brainstorming are equivalent to classical game development cycle's concept phase. There is no pre-processing stage in between game jam game development's stages. They have both prototyping phases. Production phase is equivalent to building the game. Alpha and beta testing phases are equivalent to testing and refining the game. Gold testing phase is like closing ceremonies. And finally follow up phase is equivalent to post production stage. There is not any second audience phase in game development cycle.

First comparison is going to be between pre-production phases. In game jams goals are based on self-improvement. Corporation games' goals are mainly economical. These are completely opposite aims. In game jams audience is ready, they are mainly participants in jams. Usually they are young and energetic people who has shared love in games. On the other hand, for classical game design audience have to be chosen before deciding game mechanics, theme and game arts to reach maximum number of game consumers, this is also a result from economical expectations from the created game. In addition to that, game companies have stable and familiar working environments but for game jams usually a university campus is temporarily used for "48 hours to complete a game.

Materials used in classical game development are widely ranges, since in game jams materials must be self-sufficient, material variety is decreased significantly. In classical game development, theme decision is between one of the jobs of game developing team. However, in game jams theme is usually given to the teams in opening ceremonies. Brainstorming exists in both classical game development and game jam game development, probably the biggest difference among this part is time. In classical game design it can take weeks or months to decide on levels, concept of game arts, deciding mechanics of the game, discussing about the story behind the game and the sound design.

Secondly like mentioned above, there are no pre-processing stage in game jams because in this phase mainly decisions are typed into written form and a proposal is prepared, since whole game producing team is generally maximum of 4 or 5 people in game jams there is no need for written documents between team mates.

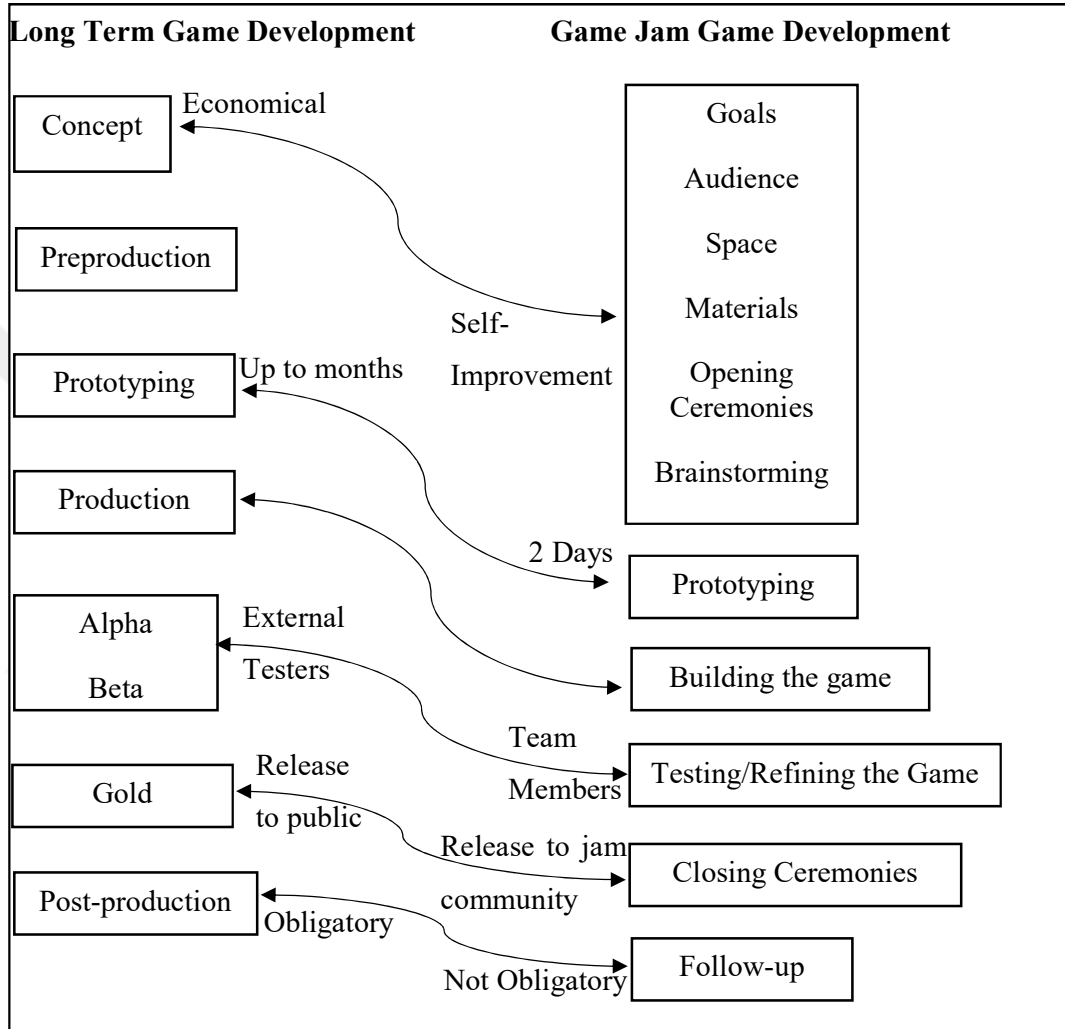
Prototyping stage and production phase are mainly similar for both game design cycles. Similar to brainstorming phase main difference between them is time. In classical game design cycle prototyping phase can take up to months but in game jams there is only 2 days to finish the game so in maximum a day or so prototyping phase has to finished. Also, in production stage this idea is valid. Time difference between two game design strategies are significant. In addition to time difference, in a game production, production teams can consist of up to hundreds of people. In game jams teams are up to 5 people. This makes resulting games simple and have short playing times in game jams.

In testing and refining game phase, team members play the game and debug it in game jams. In classical game design cycle alpha and beta testing are done. There is a significant difference between them because alpha and beta testing is done by the outside company players for classical games. Bugs are reported to the game developing team and debugging process is conducted by the. For this stage, time difference and difference between team members is a factor like all other phases too. Gold testing phase is like closing ceremonies, in order to be a final testing stage for both game design cycles. In game jams other game jam participants and jam committee meet the produced game for the first time but in game design cycle, players are familiar with the game as a result of alpha and beta testing phases.

Post production stage can occur in both game development cycles. The game jam teams are not obligated to follow up their games for post-production. Although it is highly encouraged to work on games after game jams to publish the created games for outside of the game jam community, many games are not continued to be worked on. However, for classical game design cycle it cannot be possible to not to follow up the created game in post-production. After selling a game, consumers might need help in many game related issues and company should deal with problems in the game.

These are main differences between game jam game development and classical game development cycles. Results are explaining why game jam games and classical games differ from each other significantly. This work can be expanded with surveys which will be conducted among game jam community and classical game producing communities as a follow up study to expand the content and have more accurate results.

Figure 17. Comparison Table



6. CONCLUSION

Game development life cycle uses software process models, which enclose the processes, methods and tools that assist the development cycle required to complete the design. Four different phases used generally in the development which are analysis, design, code and testing. Unlike typical game development cycles, in game jams due to the time restrictions some of the phases or processes may be excluded. The rapid increase on game jam researches and literatures assists on understanding game jam development cycles which open possibility to improve learning processes, future development cultures, compressed development processes. Moreover, they play a big role in the social aspect of the game industry.

In order to explain game jam development cycle three games (friendSHIP, SHLACK!, Bestow) were explored in terms of the mentioned processes. These games examined under the following game development stages: goals, audience, space, materials, opening ceremonies, brainstorming, prototyping, building the game, testing the game, refining the game, closing ceremonies, follow up, audience. Detailing the games' aspect on these stages a discussion about differences between game jam development cycle and game development cycles is made.

Software development life cycle also known as software process models is a strategy that encloses the process, methods and tools which is used to develop the software. Typical software development life cycle contains four phases such as; analysis, design, code and testing (Ramadan et al. 2013). In game jams participants work on programming, graphic production, interactive media design, writing narrative texts, creating sound effects or soundtracks in a limited time. Perspective research and literature on game jams are rapidly increasing. Their effect on the learning process of game developments, shaping the future development cultures, being a window to compressed development processes, they also have a giant role in the social aspect of game industry which make game jams also an important subject to research. Three games explored in terms of their game development cycles. Mentioned games are friendSHIP, Shlack! and Bestow. All games explored under

certain game development stages such as goals, audience, space, materials, opening ceremonies, brainstorming, prototyping, building the game, testing the game, refining the game, closing ceremonies, follow up, audience (Macklin et al., 2012). On these three different games Game Jam Game Development Cycle processes is explained. Game development has many phases. They generally take long times and large teams, mostly resulted in highly costly projects. A discussion about differences between game development cycle and game jam game development cycle made.



REFERENCES

Books

Novak, J., 2011. *Game Development Essentials: An Introduction, Third Edition*, Cengage Learning.



PERIODICALS

- “Findings from the GGJ six-month follow-up survey” 2014 Global Game Jam
- A. Godoy and E. Barbosa, "Game-Scrum: An Approach to Agile Game Development, " in Proceedings of SBGames 2010 Computing, I. S. F. SC, Ed., November 8th-10th 2010, pp. 292-295.
- Aggarwal, Praveen & L. OBrien, Connie. (2008). Social Loafing on Group Projects Structural Antecedents and Effect on Student Satisfaction. *Journal of Marketing Education - J Market Educ.* 30. 255-264.
- Annakaisa, Kultima. (2015a). An Autopsy of the Global Game Jam 2012 Theme Committee Discussion: Deciding on Ouroboros.
- Annakaisa, Kultima. (2015b). Defining Game Jam.
- Fowler, A., Khosmood, F. & Arya, A. 2013 “The Evolution and Significance of the Global Game Jam” in Workshop Proceedings of the 8th International Conference on the Foundations of Digital Games.
- Fowler, Allan & Khosmood, Foad & Arya, Ali & Lai, Gorm. (2013). The Global Game Jam for Teaching and Learning.
- Fullerton, Tracy & Chen, Jenova & Santiago, Kellee & Nelson, Erik & Diamante, Vincent & Meyers, Aaron & Song, Glenn & Dewese, John. (2006). That cloud game: dreaming (and doing) innovative game design. 51-59.
- GGJ Success Stories. (n.d.). GGJ Success Stories. Retrieved from <https://globalgamejam.org/ggj-success-stories>
- Goddard, William & Byrne, Richard & Floyd' Mueller, Florian. (2014). Playful Game Jams: Guidelines for Designed Outcomes.
- History. (n.d.). Global Game Jam History. Retrieved from <https://globalgamejam.org/history>
- Ho, X., Tomitsch, M. & Bednarz, T. 2014 Game Design Inspiration in Global Game Jam. Proceedings of DiGRAA 2014: What is Game Studies in Australia?
- J. Musil, A. Schweda, D. Winkler and S. Biffel, "Synthesized essence: what game jams teach about prototyping of new software products," *2010 ACM/IEEE 32nd International Conference on Software Engineering*, Cape Town, 2010, pp. 183-186.
- Macklin, Colleen & Martin, John & Dikkers, Seann. (2012). Planning your game jam: game design as a gateway drug. 203-218.
- Mitgutsch, K., & Alvarado, N. (2012). Purposeful by design?: a serious game design assessment framework. *FDG*.

- Pirker, Johanna & Voll, Kimberly. (2015). Group Forming Processes -Experiences and Best Practice from Different Game Jams.
- Ramadan, Rido & Widyani, Yani. (2013). Game development life cycle guidelines. 95-100.
- Ruonala, H., & YLIOPISTO, H. (2016). Agile Game Development: A Systematic Literature Review.
- S. Aslan and O. Balci, "GAMED: digital educational game development methodology," Simulation: Transactions of the Society for Modeling and Simulation International, vol. 91, no. 4, 2015, pp. 307-319.
- Sampugnaro, R., Mica, S., Fallica, S., Bonaiuto, A., and M. Mingrino, M. 2014. Participation at the global game jam: a bridge between consumer and producer worlds in digital entertainment. GAME, 3:35-45, 2014.
- Scacchi, Walt. (2015). Case Studies and Practices in Local Game Jam Software Development Organization: A Software Engineering Perspective.
- Toprak, C. 2014. Game Jams: A Method for Starting, Working On and Completing Games. CHI 2014. Workshop submission for Game Jam [4Research] workshop
- Yamane, S.R. (2013). Adaptability of the Global Game Jam: A Case Study in Japan.
- Zook, A., & Riedl, M.O. (2013). Game Conceptualization and Development Processes in the Global Game Jam.



APPENDICES

APPENDIX 1: friendSHIP Controller Code

APPENDIX 2: friendSHIP Enemy Code

APPENDIX 3: friendSHIP Content

APPENDIX 4: SHLACK! Content

APPENDIX 5: Bestow Content



APPENDIX 1: FRIENDSHIP CONTROLLER CODE

```
using UnityEngine;

using System.Collections;

public class ShipController : MonoBehaviour {

    public GameObject bir, ki, uc, dort, bes, alti, yedi, sekiz;

    public CurrentScript gogogo;

    public Sprite Right, Left, Up, Down;

    public float speed;

    void FixedUpdate () {

        float moveHorizontal = Input.GetAxis ("Horizontal");

        float moveVertical = Input.GetAxis ("Vertical");

        Vector2 movement = new Vector2 (moveHorizontal, moveVertical);

        GetComponent<Rigidbody2D>().AddForce (movement * speed *

Time.deltaTime);

        if (Input.GetKeyDown (KeyCode.RightArrow))

            ChangeSprite ("right");

        else if (Input.GetKeyDown (KeyCode.LeftArrow))

            ChangeSprite ("left");

        else if (Input.GetKeyDown (KeyCode.DownArrow))

            ChangeSprite ("up");

        else if (Input.GetKeyDown (KeyCode.UpArrow))
```

```

        ChangeSprite ("down");
    }

    void ChangeSprite (string directiongg){

        if (directiongg == "right")

            gameObject.GetComponent<SpriteRenderer>().sprite = Right;

        if (directiongg == "left")

            gameObject.GetComponent<SpriteRenderer>().sprite = Left;

        if (directiongg == "up")

            gameObject.GetComponent<SpriteRenderer>().sprite = Up;

        if (directiongg == "down")

            gameObject.GetComponent<SpriteRenderer>().sprite = Down;

    }

    void OnTriggerStay2D(Collider2D other) {

        if (other.tag == "Currenter"){

            this.transform.rotation = bir.transform.rotation;

            this.GetComponent<Rigidbody2D>().AddRelativeForce(new
Vector2 (0, 10));

            this.transform.rotation = Quaternion.identity;

        }

        if (other.tag == "Currenter2") {

            this.transform.rotation = ki.transform.rotation;

```

```
        this.GetComponent<Rigidbody2D>().AddRelativeForce (new Vector2  
(0, 10));  
  
        this.transform.rotation =  
Quaternion.identity;  
    }  
}
```

```
    if (other.tag == "Currenter3") {  
        this.transform.rotation = uc.transform.rotation;  
        this.GetComponent<Rigidbody2D>().AddRelativeForce (new Vector2  
(0, 10));  
        this.transform.rotation = Quaternion.identity;  
    }  
}
```

```
    if (other.tag == "Currenter4") {  
        this.transform.rotation = dort.transform.rotation;  
        this.GetComponent<Rigidbody2D>().AddRelativeForce (new Vector2  
(0, 10));  
        this.transform.rotation = Quaternion.identity;  
    }  
}
```

```
    if (other.tag == "Currenter5") {  
        this.transform.rotation = bes.transform.rotation;  
    }  
}
```

```

        this.GetComponent<Rigidbody2D>().AddRelativeForce (new Vector2
(0, 10));

        this.transform.rotation = Quaternion.identity;

    }

    if (other.tag == "Currenter6") {

        this.transform.rotation = alti.transform.rotation;

        this.GetComponent<Rigidbody2D>().AddRelativeForce (new Vector2
(0, 10));

        this.transform.rotation = Quaternion.identity;

    }

    if (other.tag == "Currenter7") {

        this.transform.rotation = yedi.transform.rotation;

        this.GetComponent<Rigidbody2D>().AddRelativeForce (new Vector2
(0, 10));

        this.transform.rotation = Quaternion.identity;

    }

    if (other.tag == "Currenter8") {

        this.transform.rotation = sekiz.transform.rotation;

        this.GetComponent<Rigidbody2D>().AddRelativeForce (new Vector2
(0, 10));

        this.transform.rotation = Quaternion.identity;

    }

```

}

}



APPENDIX 2: FRIENDSHIP ENEMY CODE

```
using UnityEngine;
```

```
using System.Collections;
```

```
public class Enemyofthestate : MonoBehaviour {
```

```
    public float walkSpeed = 2.0f;
```

```
    public float wallLeft = 0.0f;
```

```
    public float wallRight = 5.0f;
```

```
    bool facingRight = true;
```

```
    float walkingDirection = 1.0f;
```

```
    Vector3 walkAmount;
```

```
void Update () {
```

```
    walkAmount.x = walkingDirection * walkSpeed * Time.deltaTime;
```

```
    if (walkingDirection > 0.0f && transform.position.x >= wallRight) {
```

```
        walkingDirection = -1.0f;
```

```
        Flip ();
```

```
    }
```

```
    else if (walkingDirection < 0.0f && transform.position.x <= wallLeft) {
```

```
        walkingDirection = 1.0f;
```

```
        Flip ();
```

```
    }
```

```
    transform.Translate(walkAmount);
```

```
}
```

```
void Flip(){
```

```
    facingRight = !facingRight;
```

```
    Vector3 theScale = transform.localScale;
```

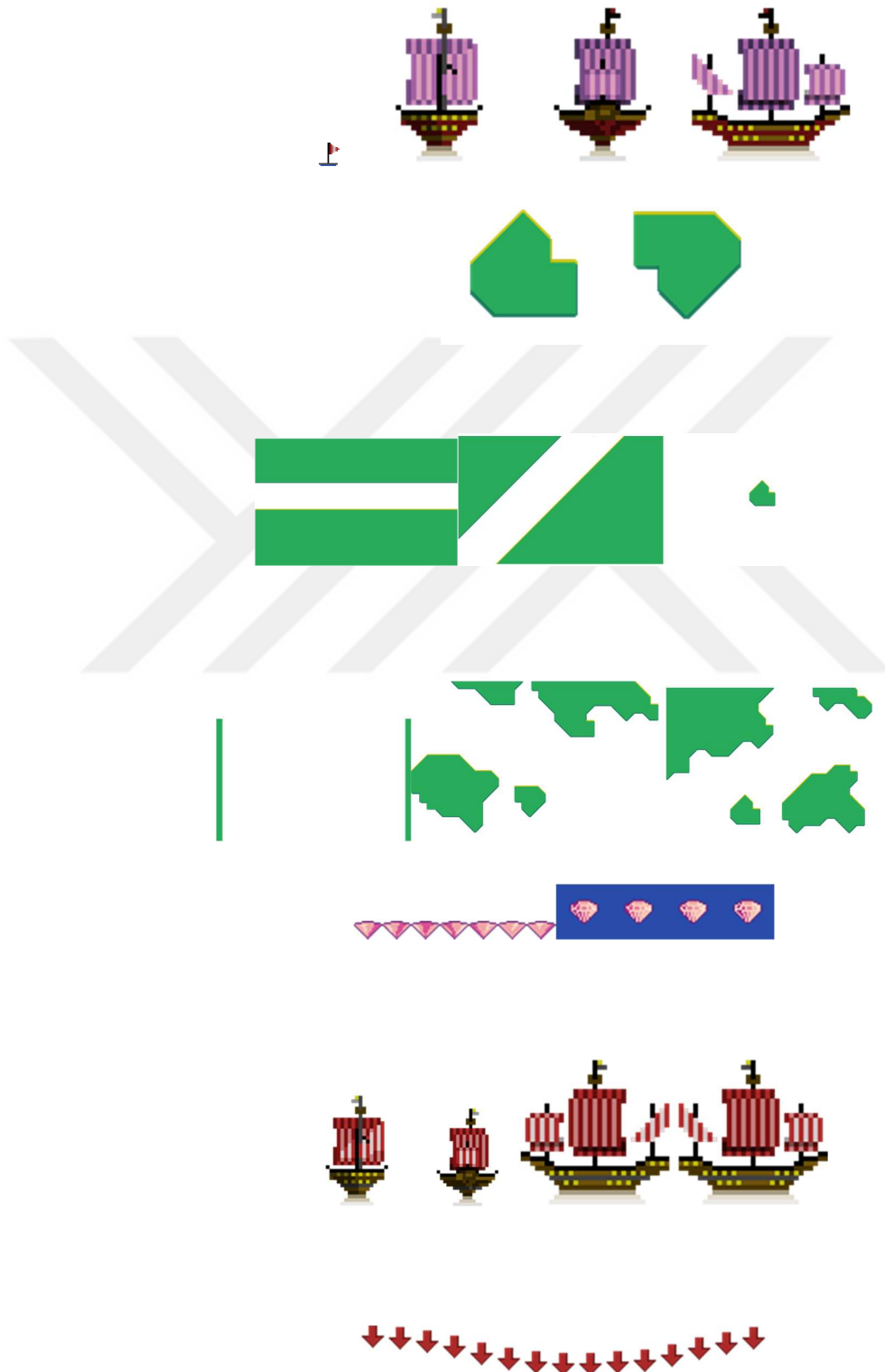
```
    theScale.x *= -1;
```

```
    transform.localScale = theScale;
```

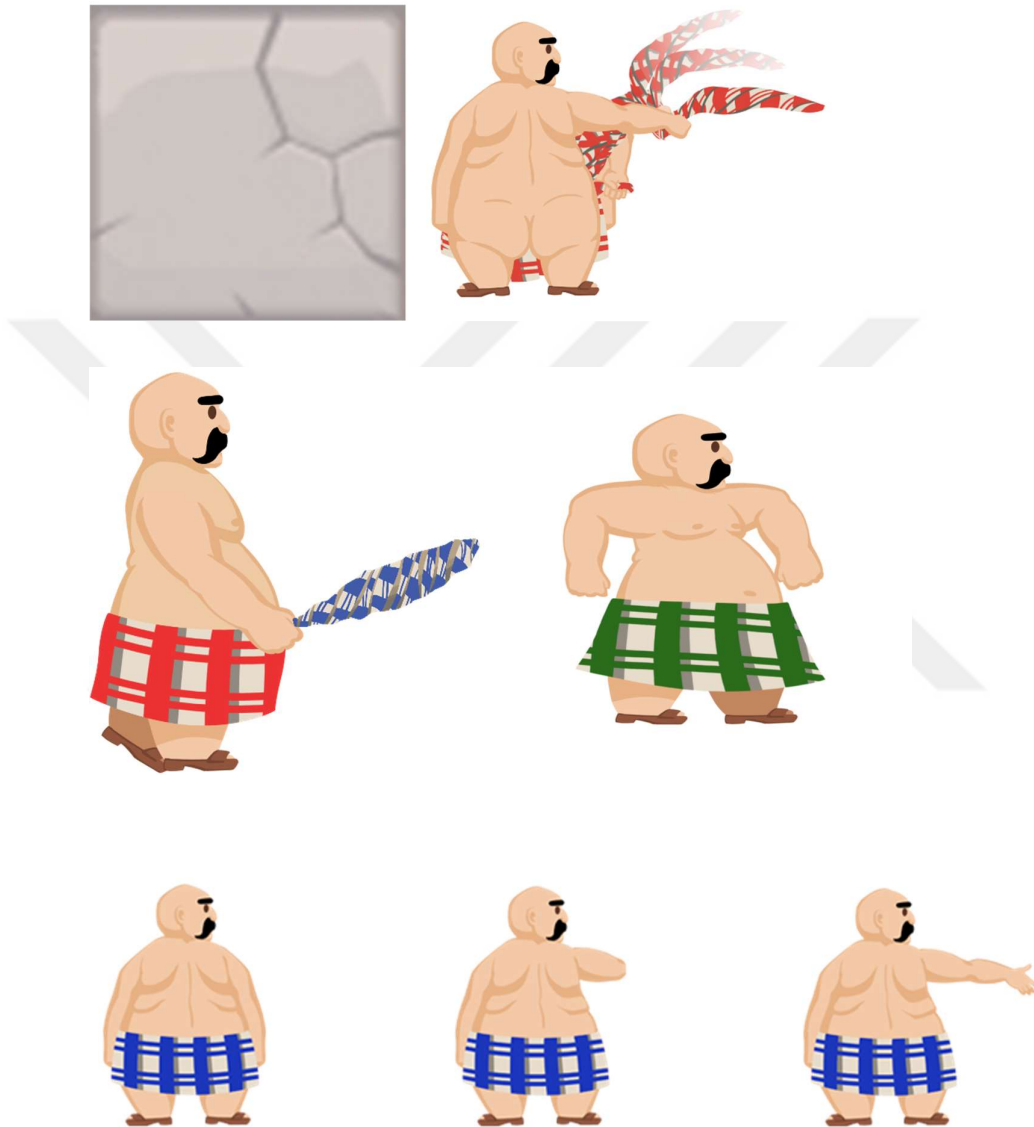
```
}
```

```
}
```

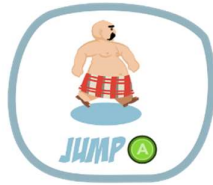
APPENDIX 3: FRIENDSHIP CONTENT



APPENDIX 4:SHLACK! CONTENT

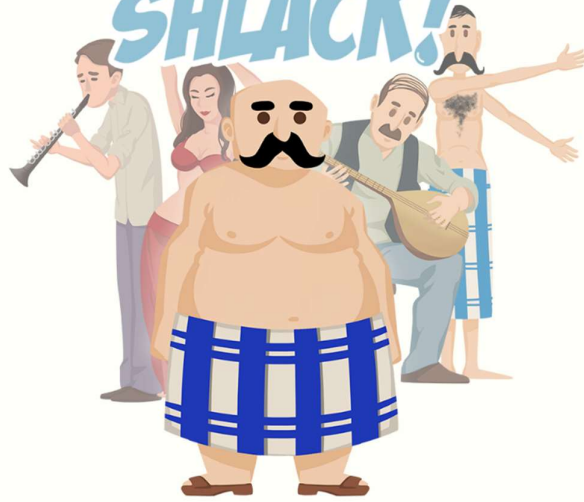


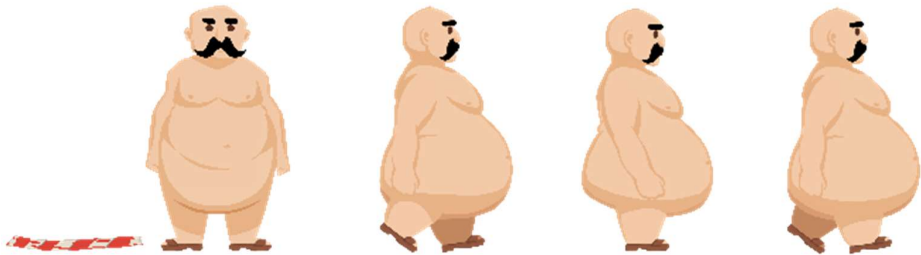
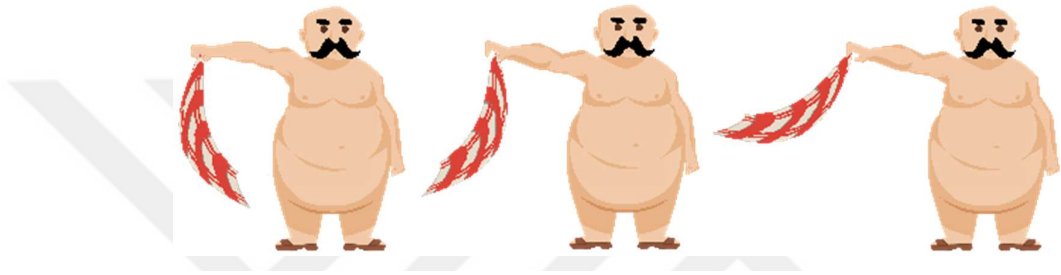
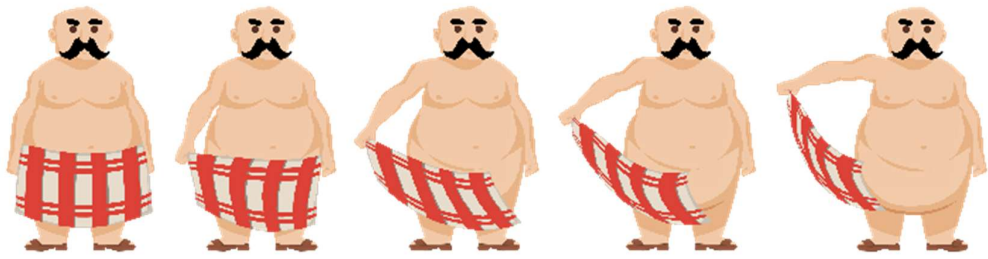
PAUSED!



CONTINUE
RESTART
CREDITS

SHLACK!





APPENDIX 5: BESTOW CONTENT

