CONTEXT BASED DYNAMIC CONTENT GENERATION, INTRODUCING A
NEW APPROACH AND A FRAMEWORK

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS INSTITUTE
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

BURKAY OZDEMIR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF MODELING AND SIMULATION

MARCH 2015

Approval of the thesis:

THE CONTEXT BASED DYNAMIC CONTENT GENERATION, INTRODUCING A NEW APPROACH AND A FRAMEWORK

Submitted by **BURKAY OZDEMIR** in partial fulfillment of the requirements for the degree of **Master of Science in Game Technologies**, **Middle East Technical University by**,

Prof. Dr. Nazife Baykal ____________________
Director, Informatics Institute

Assist. Prof. Dr. Hüseyin Hacıhabiboğlu ____________________
Head of Department, Modeling and Simulation

Prof. Dr. Veysi İşler ____________________
Supervisor, Computer Engineering

Examining Committee Members

Assoc. Prof. Dr. Alptekin Temizel ____________________
Modeling and Simulation, METU

Prof. Dr. Veysi İşler ____________________
Computer Engineering, METU

Assist. Prof. Dr. Hüseyin Hacıhabiboğlu ____________________
Modeling and Simulation, METU

Dr. Aydın Okutanoğlu ____________________
Simsoft

Dr. Erdal Yılmaz ____________________
Argedor

Date: 03 March 2015

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: BURKAY OZDEMIR

Signature :

ABSTRACT

CONTEXT BASED DYNAMIC CONTENT GENERATION, INTRODUCING A NEW APPROACH AND A FRAMEWORK

OZDEMIR, BURKAY

M.Sc., Department of MODELING AND SIMULATION

Supervisor : Prof. Dr. Veysi İşler

March 2015, 92 pages

Game Industry is growing every day with thousands of games are being released each month. Although game development tools are constantly become more easy to use, accessible and carry most of the workload of developers, serving fresh content is still a big issue for developers. Procedural Content Generation (PCG) is used as an alternative method to providing content manually. Despite its main purposes like reducing the memory usage are fading away with the recent developments, it has a huge potential to create truly never ending fresh content to players, which is one of the main problem that most of the games having these days. There are many researches being done in PCG, most of them relies on pseudo random generators or experience driven generators which are both stuck with the limits of users or testers who plays those games. In this thesis, we have proposed a new approach to procedural content generation which will help to feed the games with actual, up-to-date content by using social networks. We have generated a framework and demonstrated the usage of this generation method in a game. We have tested our games and applied a questionnaire in Appendix C by using four prototypes to understand how current dynamical approaches perceived by users. We have found out that

while our approach preferred best among prototypes, traditional random generation approach was always least preferred approach. We have also applied another questionnaire in Appendix A to measure player's experiences and forecast new possible applications of this approach. As a result we were able to pinpoint strength and weaknesses of our approach and target new genres to explore using this framework in the future.

Keywords: procedural content generation, PCG, dynamic modification, artificial intelligence, unity3d

ÖZ

ORTAM TABANLI DINAMIK ICERIK YARATIMI, YENI BIR YONTEM

OZDEMIR, BURKAY

Yüksek Lisans, Bilişim Sistemleri Bölümü

Tez Yöneticisi : Prof. Dr. Veysi İşler

Mart 2015, 92 sayfa

Oyun endüstrisi gün geçtikce gelişmekte ve her ay binlerce oyun piyasaya sürülmektedir. Oyun geliştirme araçlarının geliştirme yükünü, geliştiricilerin üstünden gittikçe daha büyük ölçüde almasına rağmen, taze ve güncel içerik, geliştiriciler için hala büyük problem arz etmektedir. Prosedürel içerik yaratımı (PIY), manuel içerik sunmaya iyi bir alternatif olarak kullanılmaktadır. PIY'nın, hafıza kullanımını düşürme olan ana ortaya çıkış nedeni gün geçtikçe gelişen dünyada önemini yitirmeye başlasada, sonradan görülmeye başlayan güncel, taze ve sınırsız içerik yaratma potansiyeli gün geçtikçe daha da göz önüne gelmeye başlamıştır. Bu konuda oldukça fazla literatür çalısması olmasına rağmen bunların yüksek çoğunluğu rastgele yaratılan veya tecrübeye dayalı girdilere dayanmaktadır. Bu tarz girdiler prosedürel içerik sistemlerinin genel olarak ya çok kontrolsuz veya yalnızca kullanan kişilerin limitlerinin kısıtlarında üretim yapmasına neden olmuştur. Bu tez çalısmasında, prosedürel içerik yaratımı için sosyal ağları kullanan yeni bir yaklaşım sunmuş ve böylece oyunların sürekli olarak güncel içeriklerle beslenmesini sağlamış bulunmaktayız. Aynı zamanda bu yaklaşım için bir framework geliştirilmis ve oyunlar üzerine entegre ederek test ve EK D'de bahsedilen anket ile savımızı destekleyen sonuçlar elde etmiş olduk. Bunun yanında EK C'de bahsedilen anket ile de sunduğumuz bu yaklaşımın artı ve eksilerini belirlemiş ve bu yaklaşımın incelenebileceği yeni oyun tarzları belirlemiş bulunmaktayız.

Anahtar Kelimeler: Prosedurel icerik yaratimi, sosyal aglar, yapay zeka , unity3d

dedicated to all Padawans who is in search of gaining true knowledge and patience, Daleks who failed to destroy our planet dozens of times and the Doctor who saved us from them, Elves that see farthest reaches of the Mordor and Leeroy Jenkins who killed all of his friends in search for adventure, and to all people who mindlessly read these words written on the little piece of universe!.

ACKNOWLEDGMENTS

I would like to express the deepest appreciation to my thesis supervisor Prof. Dr. Veysi İşler for his guidance and persistent help throughout my graduation and in this thesis. Without his help, this dissertation would not have been possible.
I would like to thank to my committee members, Assoc. Prof. Dr. Alptekin Temizel, Assist. Prof. Dr. Hüseyin Hacıhabiboğlu, Dr. Aydın Okutanoğlu, Dr. Erdal Yılmaz for their great support and concerns.
I also thank to my parents for their support and encouragement to be the best. It is because of them we are here on the first place.
Last but not least, a thank you to Emre Canbazoğlu, Ömer Avcı and Mert Gŭrkan for their substantial help and companionship throughout this journey.

TABLE OF CONTENTS

ABSTRACT iv

ÖZ vi

DEDICATON viii

ACKNOWLEDGMENTS ix

TABLE OF CONTENTS x

LIST OF TABLES xiii

LIST OF FIGURES xiv

CHAPTERS

1 INTRODUCTION 1

1.1 Scope 2

1.2 Outline 3

2 BACKGROUND AND RELATED WORKS 5

2.1 Background 5

2.1.1 Game Content 5

2.2 Procedural Content Generation (PCG) 8

2.2.1 Definition 8

2.2.2 Uses of PCG 8

2.2.3 Metaphors of PCG 9

2.2.4 Approaches to PCG 11

2.2.5 Analyzing Aspects of PCG 12

2.2.6 Inside PCG . . . 15
2.2.7 Evolutionary Algorithms and Computing (EA) in PCG . . . 15
2.2.8 Taxonomy of PCG . . . 18
2.2.9 Outline of PCG . . . 21
2.2.10 Search-Based Procedural Content Generation . . . 22
2.3 Case Studies And Related Works . . . 25
2.3.1 Procedural Content Generation in Charbitat . . . 25
2.3.2 Automatic Content Generation in Galactic Arms Race (GAR) 27
2.4 Sentiment Analysis and Sentiment Strength Detection . . . 30
2.4.1 Sentiment Analysis Steps . . . 30
2.4.2 Different Levels of Analysis . . . 30
2.4.3 Sentiment Strength Detection . . . 31
2.5 Environments . . . 31
2.5.1 Unity Game Engine . . . 31
2.5.2 Photon Framework . . . 32
3 PROPOSED METHOD . . . 33
3.1 Concept . . . 33
3.2 Conceptual Design of Context Based Approach . . . 33
3.2.1 Heuristic Model . . . 34
3.3 Generation Framework . . . 37
3.3.1 Architecture . . . 37
3.3.2 Application of the Approach: Football Manager . . . 40
4 RESULTS AND DISCUSSION . . . 53
4.1 Questionnaire 1 : Dynamical Context Generation in Games . . . 53
4.2 Questionnaire 2 : Preference test for Context Based Procedural Content Generation Implementation . . . 55

5 CONCLUSION AND FUTURE WORK 59

REFERENCES 63

APPENDICES

A Appendix A 69

B Appendix B 75

B.1 Questionnaire 1: Answers 75

C Appendix C 81

D Appendix D 83

D.1 Questionnaire 2: Answers 83

LIST OF TABLES

Table 2.1 Outline of PCG 21

LIST OF FIGURES

Figure 2.1 Evolutionary Algorithm in pseudocode [1] 16
Figure 2.2 Evolutionary Algorithm in flow-chart [1] 17
Figure 2.3 Interactive Evolutionary Computation example *PicBreeder* 18
Figure 2.4 Taxonomy by Hendrix et al. [2] 20
Figure 2.5 Three approaches to PCG including Search-Based by Togelius et al. [3] . . . 23
Figure 2.6 Content Generation Examples 25
Figure 2.7 How CPPNs Represent Particle Weapons [4] 28
Figure 2.8 Galactic Arms Race Weapon evolution examples with different number of generations [4] 29

Figure 3.1 Desert & Ice World Dungeon Map 35
Figure 3.2 Natural Disaster Events, Volcano, Tornado, Tsunami 36
Figure 3.3 Natural Disaster Events, Fire, Tornado, Blizzard 36
Figure 3.4 Sandbox User Interface 37
Figure 3.5 Core Generation Module, Internal Items 39
Figure 3.6 Core Generation Module, External Items 39
Figure 3.7 England vs. Uruguay morale and match result after modification 41
Figure 3.8 Turkey Key Players vs. Brazil Key Players after 0-4 match on 16th Nov 2014 43
Figure 3.9 Sentiment analysis of Soccer Players: Messi, Ronaldo 44
Figure 3.10 Sentiment analysis of Soccer Players: Ibrahimovic, Persie 44
Figure 3.11 Sentiment analysis of Soccer Players: Hazard Ramsey 45
Figure 3.12 Sentiment analysis of NBA Players : LeBron James, Dwayne Howard 45
Figure 3.13 Sentiment analysis of NBA Players : Kobe Bryant, Noel 46
Figure 3.14 Sentiment analysis of NBA Players : Paul, Waiters 46
Figure 3.15 Sentiment analysis of NFL players: Brady, Johnson 47
Figure 3.16 Sentiment analysis of NFL players: Manning, Vasquez 47
Figure 3.17 Sentiment analysis of NFL players: Washington, Witten 48
Figure 3.18 Sentiment analysis of F1 drivers: Alonso, Revson 48
Figure 3.19 Sentimental Analysis of Dota player Dendi 49

Figure 3.20 Brazil vs. Germany (Morale 100%-100%) , Brazil vs. Germany (Morale 100%-170%) 50

Figure 4.1 Pairs in T-Test 56
Figure 4.2 Results of Paired Samples T-Test 57

Figure B.1 Questionnaire 1 Answers 75
Figure B.2 Questionnaire 1 Answers 76
Figure B.3 Questionnaire 1 Answers cont. 76
Figure B.4 Questionnaire 1 Answers cont. 77
Figure B.5 Questionnaire 1 Answers cont. 77
Figure B.6 Questionnaire 1 Answers cont. 78
Figure B.7 Questionnaire 1 Answers cont. 79
Figure B.8 Questionnaire 1 Answers cont. 79
Figure B.9 Questionnaire 1 Answers cont. 80
Figure B.10 Questionnaire 1 Answers cont. 80

CHAPTER 1

INTRODUCTION

People of all ages play games. According to ESA, the entertainment software association, 2014 reports, 59% of Americans continuously play video games. As games join hundreds of people's lives everyday, the game content becomes more and more important[5]. Players demand for fresh, rich and constantly changing content rises, however, manually adding novel content is still expensive[6]. That is obviously one of the main reasons why Procedural Content Generation in games become more frequently used in today's games that takes a lot of effort to include more in-game content.
Procedural Content Generation (PCG) means creating game content automatically using algorithms, artificial intelligence, planners or similar ways. Game content, consists of all aspects in the game from more abstract parts such as textures, sounds, behaviors to more concrete parts, some of which are characters, terrains, buildings, maps, vegetation, story, story-line, ecosystem, quests, dialogues, levels and more..

The need of PCG can be explored under several points. The first point, which also gave birth to PCG, is memory consumption. PCG implicitly requires less memory since it can keep the content compressed until needed. PCG typically needs numbers, keywords or seeds to generate content when needed, that's why in example of Elite, (Published by Acornsoft) the game was able to keep hundreds of star systems in couple of ten kilobytes of memory by storing only numbers. As available memory on the hardwares increased, the need for memory consumption became less critical but still important. Another point, which we can say more important these days is the expense of manually creating game content, and keeping the game world alive. This problem becomes more evident in the production of complex AAA games. As an example, in 2008 ComplexityGaming reported that, World of Warcraft was consisting of over 2,000,000 words of text, 1,400 geo-locations, 7,600 quests, 30,000 items, 5,300 interactive objects created during 5 years of development. Most of the AAA commonly use procedural generation software like *SpeedTree* and more to reduce the expense but still create the level of detail they would like to see in games.
In addition to the expenses, procedural content generation has a potential to generate truly fresh

and rich content and thus produce an endless games with many possibilities. If game content can be generated according to the style of a particular player or group of players, it might be possible to create close to infinite replay value[3]. PCG might help to dynamically modify properties, trigger real events, narratives, stories, rules and others based on core criteria given by game designers.
Aim of this thesis is to build a new approach and introduce a framework for not only generating content using randomized seeds or algorithms but also generate and modify the content dynamically using a context driven by real world action and events by using an incredible data flow of the current social media environments. This approach differs from other randomized generations with the fact that it's been created by living people and big enough to satisfy needs of the generating mechanisms.
The outcome of using such an environment to generate content is that, the games using this framework will have the power to acquire truly up-to-date actions and events, thus games will be able to generate a living content depending on the real life. In other words, the real life that is effecting gameplay such as, when there is an earthquake happening in any location of the world, the earthquake also happening in the game or when football player is playing bad in real world, it also affects virtual player in football game. Not only such events but also, generating stories, quests, generating new species, depending on the seed keywords extracted by the social media and finding keywords that are related to the seeds and core foundation to more possibilities will be introduced with this framework.
To develop a framework that has this kind of capabilities, a pilot network, Twitter, is chosen with the reason of having large amount of public accessible data on contrary to other social networks. However, other networks that have a huge pool of public data and give access to its data flow can also be used as a core of this generation framework.
Ultimately, we aim to apply our approach in few small-wide games to show the effects of dynamically mining these kind of data and generating new content from Twitter.

1.1 Scope

The main focus of this thesis is to introduce a new dynamical content generation approach, develop a framework that is using a social media in its core to generate in-game dynamic content such as events, actions, in-game props and so on. Our main contribution is introducing a new method for procedural content generation showcase initial applications and gather feedback from people by applying questionnaires to forecast it's boundaries, detect it's shortcomings, explore new ways to implement our approach. We plan to show the effects of this contribution on a game developed with Unity engine. Generation Framework is built with .NET and C# and a photon server application is used to connect with popular game engines such as Unity.

1.2 Outline

This thesis has been organized into the following chapters:

- Chapter 2 provides a background and an explanation of ”game content”, ”procedural content generation (PCG)”, ”evolutionary algorithms & computation(EA & EC)”, discusses various types of PCG in the literature. It also gives detail about ”sentiment analysis” and brief information about Photon Framework and Unity game engine.
- Chapter 3 describes the concept of our proposed method, explains the generation framework and its architecture in detail, shows how the generation framework works, show the games developed to demonstrate the approach.
- Chapter 4 shows the questionnaire applied to the selected subjects then discusses the results and the effects of using this approach and talks about following future works to be done.
- Chapter 5 concludes our thesis, summarizes the approach, generation framework and its effects in game and general scope.

CHAPTER 2

BACKGROUND AND RELATED WORKS

2.1 Background

The main focus of this thesis is to create a dynamic content generation framework which uses the huge data flow of external social networks and generate content depending on the context of the data. In this section, we will first describe game content defined by Hendrikx et al. [5]. After game content is defined, we will introduce what Procedural Content Generation is and discuss the main uses of PCG. Then we will discuss Procedural Content Generation metaphors created by Khaled et al. [7]. We will look into each metaphor in detail. Then, we will look at the main aspects of PCG to understand the role of PCG in games. Next, we will move on to dissecting PCG and define the taxonomy of PCG. We'll go through the related works and researches made in PCG and delve into the variety of forms. While talking about PCG, we will be addressing commonly used evolutionary computing and interactive evolutionary computing to generate content. Then we will explore industrial and academic examples of games that uses PCG mechanisms. The proposed framework make use of sentimental analysis in its core processes. Therefore we will explain sentimental analysis and sentiment strength detection. Finally we will give brief information about Unity, Photon Framework the game engine and server framework used in this thesis to develop the dynamic content generation framework.

2.1.1 Game Content

Procedural Content Generation has been used for couple of decades. What can be procedurally generated in games? In order to understand what we are capable of doing with Procedural Content Generation, we have to define what game content is and dissect the elements that's forming the game content as a whole. Content means almost all the things that game contains like levels, maps, rules, rule-sets, textures, items, quests, sounds, vehicles, characters etc
Nitsche et al. argues that the game content can be defined in abstract and concrete ways [8].

Concrete bits defines the elements that are closely related to the human perception, in other words objects in the world that are interacted. We can give example of the concrete parts such as, forests, mazes, buildings, vegetation waters and so on. Abstract bits are the parts that should be combined to form a concrete part. Textures, sounds, behaviors are examples of abstract parts of games.
On his survey, PCG Nitsche et al. gathers commonly used elements in the game content and defines the game content under 5 main groups namely, Game Bits, Game Space, Game Scenarios, Game Design and Derived Content which are discussed in the sections below.

2.1.1.1 Game Bits

Textures (Abstract Bit) are basic images used to give visual representation to the objects in the game. Textures generally define the art style of games. Most commonly they used with material of the object which actually has different properties that define physical properties of the objects such as reflection, scattering, luminosity, opacity and such.
Sound (Abstract bit) is another important part within the games. Musics, soundtracks and sound effects are used to give both atmospheres and feedback to the players according to the type of different objects. Although there is notable resistance to the algorithmic compositions from musicians [9], importance of procedural generation in sound should not be ignored.
Behavior (Abstract bit) defines how objects interact with each other and with the environment. Giving behavior to the objects may make the game more interesting, interactive and lively. Procedural behaviors generally used to create object behaviors.
Environmental elements, Vegetation, Trees, Fire, Water, Stone and Clouds (Concrete) is commonly used in many games for realistic terrains and environments. Lots of procedural generation tools are available for these kind of objects to give the detail artists want, without losing precious resources.

2.1.1.2 Game Spaces

Indoor Maps are illustrations of all the structures and objects relatively positioned in the rooms of indoor spaces. Rooms might be connected by corridors, stairs, objects that are commonly seen in real world environments and they can be grouped in dungeons. In real world, dungeon means cold, dark, terrifying places but in computer games they mean maze or labyrinths which adventurers enter, fight, slay monsters, collect hidden treasures, rescue people and clear and finally leave dungeon from end point. They can be abstract spaces like in Sokoban or concrete spaces.
Outdoor Maps are the illustrations of the structures of outdoors environments. Many successful commercial games has comprehensive outdoor maps or mixed indoor outdoor maps like in the ”Commander Keen” series. They can be abstract spaces like chess board or concrete spaces.

Water Bodies (Concrete Space) means embodiment of waters like seas, oceans, rivers, lakes which are commonly used in games.

2.1.1.3 Game Systems

Ecosystems defines the general placement, evolution of the flora and fauna. Ecosystems generally effect player's powers, resistances and psychologies. Ecosystems can be abstract or concrete systems.
Road Networks are structures of outdoor maps, made for purposes such as transportation. Road networks can be abstract or concrete systems.
Urban Environments are clustered building where variety of people live together and interact. Urban environments can be abstract or concrete systems.
Entity Behavior defines interactions, behaviors of the entities like NPCs, other intractable environment objects and so on. Entity Behaviors are abstract systems.

2.1.1.4 Game Scenarios

Puzzles are abstract part of game scenarios which players try to find a solution based on his skills or hints and tips systematically embedded in the problem [10]. Many quests, objectives consists of one or many puzzles and reward players who solve them.
Storyboards are usually used by game developers, designers and sometimes players. Storyboards are quick sketches with visual and textures, that describe some action, event or a story. They might guide players or developers depending on how its make and its purpose. Depending on the context, storyboards can be abstract or concrete.
Story is one of the key and core part of the games which motivate players to keep playing and reveal the content, goals within a logical means. Stories can be abstract or concrete parts.
Level are the abstract or concrete type of concepts are separators between game play and used in most of the games.

2.1.1.5 Game Design

World Design is abstract parts of the game design that can be defined as "the design of a setting, story and theme" [11] **System Design** usually the concrete parts of the game designs that means the "creation of mathematical patterns underlying the game and game rule sets" [11]

2.1.1.6 Derived Content

News and Broadcasts are usually concrete side contents of the game that give players reveals more information on the story or reveal irrelevant information regardless of the story **Leaderboards** are the abstract parts that ranks player according to their scores they made in games.

2.2 Procedural Content Generation (PCG)

Since we've explored what we mean by game content, we can move on to the definition and details of procedural content generation in literature.

2.2.1 Definition

Procedural Content Generation means the creation of game content using algorithms with or without help of parametric inputs. Togelius et al. made short concrete list of what they consider to be PCG by giving examples[12]

- A tool that creates dungeons for games
- A system that creates new weapons in games
- A system that creates new weapons in games
- A program that generate complete playable and balanced games such as board games
- A middle ware that populates game world with content such as vegetation
- A visual design tool that lets user design strategy maps while constantly evaluates the designed map depending on the domain, properties and suggest improvements.

2.2.2 Uses of PCG

Procedural content generation in games is used for wide variety of different reasons. The PCG first appears in the 1980s in the Rogue game. First need in PCG is more on the storage and memory since the capacity of hard-wares were pretty low back in the days. Rogue uses PCG to increase the game replay-ability by generation environments automatically instead of storing manually on the memory. Elite (Acornsoft 1984), space trading and adventure game uses PCG to keep hundreds of star systems in few kilobytes of memory by only storing few numbers.
Currently, after increase in the storage sizes, memory and CPU power, probably the most

prominent reason why PCG is out there is to lift the large burden on the game content makers in a limited amount of time when making complex game worlds. After three decades of research, Many PCG methods exists that play the role of alternative creation tool for almost every content we see in the game. *".kkrieger"* uses PCG to generate textures, meshes, sounds to create complex games and demo videos within 100KB of memory. Tools like *SpeedTree* (Interactive Data Visualization, Inc), *WorldMachine* , *Spore Creature creator* are used to lift the heavy burden and populate the world in detail the designers want.
Another very popular reason the use PCG is to personalize, enhance and naturalizing the content of games. In *Borderlands*, weapons are generated procedurally by evolving the guns in the game. Charbitat by Nitsche [2] generates worlds as players play through the game with the purpose of exploration and procedural generation of game spaces. Another good example to personalization is Galactic Arms Race by Hastings et al. [4] which uses interactive evolutionary computation and cgNEAT [4] to generate new weapons according to user preferences.
PCG methods are also commonly used in games to generate in-game dungeons [13], in the examples diablo sequels, rogue-like games.
Creating rule-sets and game mechanics, planning quests and stories, creating mazes and puzzles can all be seen as examples that doing both designer support and personalizing content. There are also experience-driven examples [14] that specifically focusing on the player's experience on games and how to craft better or specular experiences.
In theory, there is a possibility to create truly endless game with the use of PCG that offers fresh, new and motivating quests, contents, terrains and other game elements. Although general generation framework which does that task, current PCG systems does lots of work to support developers, enrich the game content and reduce the hardware consumption.

2.2.3 Metaphors of PCG

The definition of the game content shows us the parts that can be generated procedurally. In recent years, the huge increase in storage, memory and CPU power, made PCG one of the main parts of the games. However, since many people from different perspectives work on development of games, they need a shared language when using PCG. To make the procedural content generation more meaningful for all the developers from different disciplines, Khaled et. al. made a research to create metaphors in a design centric way to group the PCG exercises semantically and highlight the limits and qualities of PCG. These metaphors developed by surveying the PCG literature and finding their systematic relations to design process [7]. Four proposed metaphors are TOOL, MATERIAL, DESIGNER, EXPERT.
PCG systems can be viewed as one or multiple of these metaphors.
Lets discuss these metaphors in detail.

2.2.3.1 TOOL

TOOLs can be defined as instruments used for the purpose of achieving specific design goals. TOOLs assist designers and optimize the performance of their users. TOOLS should be able to reach goals that designers want to achieve, should minimize cost of adaptation and should be less costly then doing the actual goal manually. For example, tools can help to design gameplay mechanics [15], levels [16], quests [17] and more. TOOLs can aid designers, developers or even non-technical users [15] to create games or game designs.

2.2.3.2 MATERIAL

Materials reflects physical properties or forms of objects. Materials can be shaped, manipulated and re-configurable to give expected physical forms.
MATERIALS, in the context of metaphors dynamical, procedurally generated, re configurable parts that can be used and shaped by their users. MATERIALS can be seen as TOOLS in the means of giving the users support to achieve goals. MATERIALS can also be completely generated procedurally and do not require further use. Unlike tools, MATERIALS are generally passive instances. MATERIALS usually generated offline but it can also be generated online. *SpeedTree*, which is very common software in game production can be seen as MATERIAL. Weapons that are procedurally generated in Borderlands (Gearbox Software 2009), levels generated in Rogue-like games also considered as MATERIALS. MATERIALS generally generated depending in the parameters.

2.2.3.3 DESIGNER

Designer can be defined as PCG algorithm that solves specified design problems in the game and issue game design tasks with only little, if not none, interaction from the actual designer. DESIGNERS are not limited to perform any tasks. They can design gameplay mechanics, adding new plots to the game story, adjusting the difficulty, making decisions about art style and more. DESIGNER, defined by Khaled et al. can be ASSISTANT to the real designer or LEAD DESIGNER with more responsibilities and capabilities. LEAD DESIGNER can design rules, rule-sets, entire aesthetics or alternatively investigate how the design works. Puzzle-Dice made by Fernandez-Vara, generate narrative puzzles by using puzzle map made by game designer and game objects from database. Puzzle-Dice can be seen as ASISTANT in this context. Togelius and Schmidhuber's experimental work which they use evolutionary computing (EC) to evolve rules and agent logics to automatically design game [18] can be seen as LEAD DESIGNER.

2.2.3.4 EXPERT

EXPERTs are authoritative systems that has high knowledge within AI forms that emulate the problem solving and decision making by using their knowledge. PCG works that are related to monitoring, analyzing, interpreting, assessing data is main focus of EXPERTS. Khaled et al. proposes that there are two main forms or experts, PLAYER EXPERTS and DOMAIN EXPERTS. While both forms of experts can use same bases like interpreting game state, they divide on their analysis and interpretation.
PLAYER EXPERT interpret, analyze depending on experience of the player. The results given by PLAYER EXPERT is used in personalization. Player Experience Modeling term in PCG defines this usage[14]. PLAYER EXPERT's usual usage is to support DESIGNERS.
Personalizing race tracks to the driving style of players of racing games [19] by Togelius et al. can be an example to PLAYER EXPERT. As should all experts, PLAYER EXPERTS should have high knowledge in their area. They are often trained on player data which is representing the targeted audience. With training in place, PLAYER EXPERTS gain the ability to analyze and interpret how player behaves and output the results of the analysis. PLAYER EXPERTS can also be DESIGNERS to have the responsibility of designers as in Kazmi and Palmer's system [20].
DOMAIN EXPERTs outputs domain-specific analysis. Domain-specific means that the expert know the appropriate measures depending on the domain. In the example of the learning game that teaches fractions, Refraction, leveling up occurs when players master the specific problem. DOMAIN EXPERT here makes analyses to find return the practice and difficulty needed to master by player according to the domain specific needs. Correct interpretation, presenting tasks, difficulties, challenges that are appropriate depending on the domains shows the value of the DOMAIN EXPERTS.
These metaphors are needed to be absorbed by game developers to be able to share same terms when they are using PCG in game development. PCG metaphors also gives the ability to check and expand the quality, limits and usabilities of game PCGs. These metaphors has shown to serve as a generative function which suggest ways to extend existing uses of PCG in better supporting in the large-scale projects[7].

2.2.4 Approaches to PCG

As mentioned before, Procedural Content Generation has been used in wide variety of context thus there are many approaches to the PCG. Smith et. al. summarizes these approaches under six main points[21]; Optimization, Constraint Satisfaction, Grammars, Content Selection, Constructive.

Optimization Optimization approach tries to find best fit for combination of basic elements by generally using search methods and evolutionary algorithms and aids the design processes. These kind of methods generally works slow and used offline.

Constraint Satisfaction This approach defines the PCG algorithms specifically involving with the producing satisfactory end results. In this approach designer has the power to define what end goals should be and how these end goals appear to the users. One problem with the constraint satisfaction approach is that it may be difficult to represent the facts. Modeling and simulation of Complex Systems, *described below*, can be used in this context.

Grammars This approach mainly relies on creating sentences and turning one string to another in a right way using set of rules. Grammars try to maintain a balance between designer and rule-sets while expanding the small contents to fit them together. *SpeedTree*, vegetation generation, wall/building generation can be example to this approach.

Content Selection This approach is very simple form of PCG that decides what content should be selected or not selected when generating content.

Constructive Constructive approach fits building blocks together and finishes the generation. It doesn't test the end result. Constructive approaches should be able to satisfy some acceptable quality and generally are game-specific systems.

2.2.5 Analyzing Aspects of PCG

To understand the role of PCG in games, Smith et al. proposes a framework that contains 3 main aspects: Mechanics, dynamics and aesthetics[21].

Mechanics Mechanic aspects of PCG can be divided to Building Blocks, Game Stage, Interaction Type and Player Experience.
Building Blocks defines the presentation layer of generators.
Building blocks, defined within 5 categories.

- **Experiential Chunks**, are very large blocks that authored by humans.

- **Templates** are generic forms of experiential chunks, beside human authoring, computers also fill some blanks left by users. These blanks are generally very small frames. Most of the content still generated by humans.
- **Component Patterns**, are relatively small blocks and does not dictate the experiences a human will be having. These kind of patterns are still identifiable as a human authored blocks.
- **Sub components** are internal representations that are using building blocks like human designer.
- **Game Stage**, is the stage where generation is being done in the game.

As described in Togelius taxonomy [3], *game stage* consists of the online and offline content. Briefly,

- **Online** means that the generation is being done in run-time.
- **Offline** means that the generation is done either in the development or offline stage of the game

Interaction Type, is the type of interaction generator mechanisms uses as input from users

- **None**, means that the generation mechanism doesn't take any input into account. All of the content is being generated by the PCG and there is no interaction by player or designer. MineCraft, Endless Runner Games, Rogue-like games are great examples to this kind of control.
- **Parameterized control** means that there is a form of indirect control where generators use some parameters from players to manipulate the generation process.
- **Preference control** means that the generation is being done by preferences of the players, playing the game. Galactic Arms Race is great example to this control type.
- **Direct Manipulation** means that the players are purposefully molding the content. This can usually be seen on PCG design tools such as *SpeedTree*, **WorldMachine** and so on.

Player Experience, means that generation is driven by the game experience of players or crafts specular kind of experience for the players.
Indirect, means that the generator has indirect control over players. Like in Borderlands weapon generation system, system doesn't act to change player experience, instead contents that generator creates change the experience in some unforeseen way.
Compositional control over players means that not whole but some parts of the content has

some affect over players. Compositional control doesn't let players to control for the experience. Experimental control is type of a control that generator is able to manipulate the experience players are having. In this type of control, players can interact with the mechanism.

Dynamics

Dynamics aspects are game rules interact with each other in gameplay. Dynamics are categorized in 6 categories: Relationship to Other Game Mechanics, Memorization vs. Reaction, Strategizing, Searching, Practicing and Interacting.

Relationship to Other Game Mechanics

- **Core relationship**, is a type which games are relying on PCG in their core such as rogue-likes. In these kind of relationships game design, level design mostly gets done by the PCG mechanism.
- **Partial Framing**, is a kind that players are not exposed completely by PCG but the game still relies on PCG to some extent. Game that use that type of PCG, such as Civilization, generally use PCG in early or late content but the total experience doesn't rely on that.
- **Decorative type** of relationship with PCG in games means that PCG is purely cosmetic, in other words generates the optional content, like buildings, populations, vegetation and so on.

Memorization and Reaction

Games that has the dynamics of memorization, means that game plays with player's memory to overcome goals such as finding paths within the game or mastering the game by beating best scores by finding hidden objects.

Reaction, usually used in "Endless Runner" type of games that plays with reaction of players to reach goals in the game or beat high scores.

Strategizing

Players building strategies triggering the generators and generators producing outputs depending on the trigger sums up this kind of dynamics. This kind of dynamic is only possible if the generator has the ability to respond reasonably to the developer or player. There should be apparent mapping between generator and player in this mapping.

Searching

Creating huge world that would be either not possible or very hard to create for developer or designers, creates the dynamic of searching. Searching means generator create huge and surprising content that players can enjoy exploring this huge space and mold and manipulate objects easily as they please. Popular examples include Minecraft, Charbitat.

Practicing and Interacting

- **Practicing dynamic** either gives players chance to manipulate the settings of the content by changing how the generator work, such as generating dungeons using parameters gathered directly from players, or give chance to players experience the same setting with new strategies such as playing different combination of skills, items and so on.

- **Interacting dynamic** gives player the ability to feed the system and the lets player community talk over differences in those experience, like in Civilization game prompts letting players discuss different settings.

Aesthetics
Aesthetic experiences created by the dynamics is summarized in 3 categories.

1. PCG mechanisms that creates discovery as a main aesthetic element. Such as in the *"Searching"* dynamics creating huge worlds to explore.

2. PCG mechanisms that creates challenge as its main aesthetic experience. Such as in the *"Reaction"* and *"Practicing"* part, creating highly reactive dynamics to experience;

3. PCG mechanisms that creates fellowship ass its main aesthetic element. Such as in the *"Interacting"* part, creating player communication inside or outside of game environment creating the ecosystem for the players to discuss their experience between themselves.

2.2.6 Inside PCG

While developing a PCG algorithm, there are commonly and well-known algorithms that we need to aware of. Following subsections describes these algorithms in detail.

2.2.7 Evolutionary Algorithms and Computing (EA) in PCG

2.2.7.1 Aim

Evolutionary computation is basically an abstraction from the Darwinian Evolution that is used for optimization and solving complex problems. Evolutionary algorithms are subset of evolutionary computation such as genetic programming. EA is often applied in PCG. Although evolutionary algorithms should not be married to PCG, it is worth to look into definition and components evolutionary algorithms to understand why they are commonly used in PCG and what their uses are.

2.2.7.2 Definition

Although there are wide variety of Evolutionary algorithms, core idea behind the evolutionary algorithm is ”survival of the fittest”, means that natural selection of the individuals depending on the environmental pressure on the vast population of individuals[1].
The usual way is to have a initial population, applying a quality or fitness function and let only fittest to survive. From the survivors, apply recombination using off springs of two parents or mutation to create the next generation. During this process there are two principles. First, the variation operators should create the required diversity and help novelty, and the second is selection should put pressure and push quality. After that, we have a next generation with their new fitnesses. This whole process can be iterated until we reach to the acceptable fitness value. It should be noted that the selection here gives higher chance to the fitter candidates, however there are still chances to the less fittest become parent for recombination. We can also see the pseudo-code of EA in Figure 2.1 and Figure 2.2

```
BEGIN
  INITIALISE population with random candidate solutions;
  EVALUATE each candidate;
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    1 SELECT parents;
    2 RECOMBINE pairs of parents;
    3 MUTATE the resulting offspring;
    4 EVALUATE new candidates;
    5 SELECT individuals for the next generation;
  OD
END
```

Figure 2.1: Evolutionary Algorithm in pseudocode [1]

Evolutionary algorithms use generate-and-test method mentioned in the taxonomy. They mostly use recombination to create new generations.

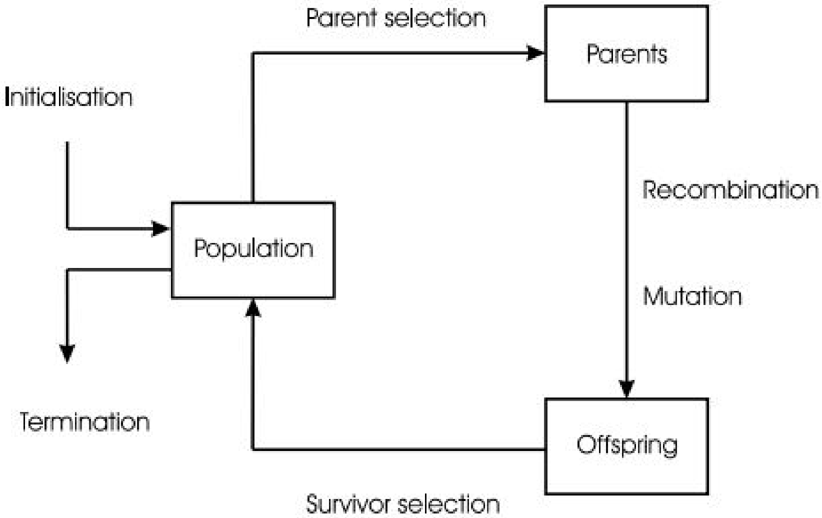


Figure 2.2: Evolutionary Algorithm in flow-chart [1]

2.2.7.3 Fitness or Evaluation Functions

Fitness function is used to set the requirements or quality for the problem that the original context should be fitting. This function typically represents quality measures for the problem, means the phenotype or genotype spaces that differs according to the use of representation. There are three key classes for fitness functions[3]:
Direct evaluation function extracts some features from the content and maps directly to the fitness value. Mapping can be linear or non-linear and doesn't have large computation. Direct evaluation functions usually designed for specific game or content. According to Yannakakis et al. there are two variance of direct evaluation functions[14] which are theory-driven evaluation functions and data-driven evaluation functions.

- In **theory-driven evaluation functions** designer is almost always guided by a intuition, qualitative emotion theory or player experience. Examples include [22, 23, 24, 25, 26, 27]
- In **data-driven evaluation functions** evaluation is always based in collected data and various content examples, like questionnaires, physiological measurements. Examples include [28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38]

Simulation based evaluation functions take place when it is hard to design an evaluation function for specific games or contents. Simulation-based functions involves with an AI agent playing part of the game or content like clearing dungeon without dying or successfully finishing the race track and outputs the quality of the test space. AI agent can either be hand-coded or

self learning agent based on behavioral models. Simulation based functions are usually more needs more computation power than the direct functions. Main idea behind the simulation based evaluation is to have an agent that is playing the content very similar to humans. Thus the functions can only be connected with gameplay based experience models[14].
Yannakis et al. describes key distinction in the simulation based functions as :

- **Static evaluation functions** which is one agent playing through the content. Related researches for static functions include [19, 39, 40]
- **Dynamic evaluation functions** which changing agents playing through the content. Studies related to dynamic functions include [41, 42]

2.2.7.4 Interactive Evaluation Functions

Interactive Evaluation Functions gather data from users inside the game and evaluate during the gameplay. These collected data are used to enhance player experience and it affects the evaluation function. In Figure 2.3 pictures are generated completely by getting external inputs from users.

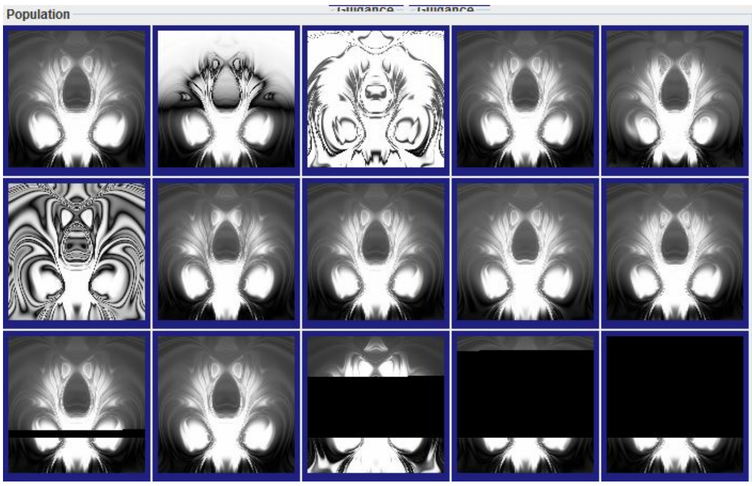


Figure 2.3: Interactive Evolutionary Computation example *PicBreeder*

2.2.8 Taxonomy of PCG

Now that we have explored approaches, aspects and algorithms used in PCG, we can take a look at taxonomy of PCG.

2.2.8.1 Online and Offline PCG

This distinction in PCG refers to whether the generation is performed online, during run-time, or offline, in development[3]. Example for online PCG would be generating the next part of the game world when player passes the specific positions on a map or game world. Or generating the inside of the building when players enters the building. Example for offline generation would be generating the puzzle in a level according to some parameters defined by designer. Mixed cases can also be seen in the PCG such as suggesting new maps in strategy games depending on player's play styles and letting them choose the new map. Online PCG should be fast that it shouldn't take too much time in the process and results should be in acceptable or reasonable quality.

2.2.8.2 Necessary and Optional Content Generation

Another distinction in generated content is whether the content is necessary or optional[3]. With necessary content, we mean the game content that is necessary to be able to play the game such as monsters, dungeons, game rules and so on. In contrary, optional content refers to the content that populates the game but doesn't necessarily be interacted or completed by the player to progress in the game. Houses, buildings, side quests, huge variety of weapons can be examples of optional contents.

2.2.8.3 Random Seeds and Parameterised, Complex Algorithms

Content generation in the games might take random numbers to as seeds to its random number generator or use complex algorithms to generate content. An example of random generation would be generating dozens of stars or solar systems in the universe that you went through. On the other hand, algorithm might take parameterized vectors such as number of rooms that dungeon should contain, minimum number of chests that player should find or artificial intelligence algorithms and simulations of complex systems to generate stories and quests and so on. Hendrix et. al. gathers the following Figure 2.4 by making research over 50 research papers to show what kind of algorithms is used to generate content[5].

Generative Grammars, as described before are sets of rules that can generate grammatically correct sentences and rewrite strings, turn one string to another.

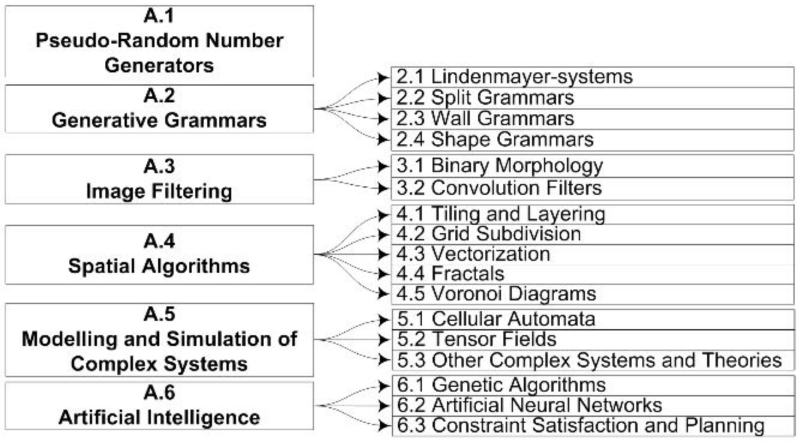


Figure 2.4: Taxonomy by Hendrix et al. [2]

1. **Lindenmayer** systems consists of set of symbols made by the application of certain rules.

2. **Split Grammars**, are three dimensional design grammars that works on encoded shapes closely to the L-Systems. The grammar generates a new shape by rewriting rules based on a basic set of shapes and doing shape-to-shape conversions.

3. **Wall Grammars**, designed specifically to generate building exteriors. Shapes generally molded to create building exterior similar to the split grammars but wall grammars has the ability to generate much more advanced shapes.

4. **Shape Grammars**, are sequential, context-sensitive grammars that is able to generate complex structures like wall grammars. It rewrites the symbols like wall grammar but in each step, symbol and it's neighbor decides what or which symbol should replace the original one.

Image Filtering is set of commonly used methods such as convolution filtering, binary morphology to enhance, measure or extract features from images.
Spatial Algorithms are ways to manipulate spaces like tiling, layering, subdividing on grids, Voronoi diagrams [43], fractals, which means recursive figures that consist of copies of themselves.
Modeling and Simulation of Complex systems are needed when mathematical equations are not enough to describe or define the cases. These systems includes, cellular automata [44], tensor fields [45], agent based simulations based on using individuals in modeling complex situation [46].
Artificial intelligence which is a large field to simulate real intelligence. Procedural generation with AI contains, genetic algorithms [47], Artificial neural networks [48], planning [49].

2.2.8.4 Stochastic and Deterministic PCG

A completely deterministic algorithms outputs the identical content when given inputs are the same. These kind of algorithms can usually be seen in the compression and extraction algorithms. Good example of this is *.kkrieger* which compresses textures, objects, musics and levels to under 100 KB of memory. Another good example is Elite space adventure game which compresses solar systems. Stochastic approach does exact the opposite. Nowadays, most of the PCG algorithms, if not all, works stochastically to some extent. Stochastic algorithms can bee seen in the recent works such as Borderlands, Galactic Arms race[4] and so on.

2.2.8.5 Constructive and Generate and Test PCG

Finally, constructive method means that the generation algorithm or mechanism generates the content for once and finish the job. Constructive algorithms make sure that the generation is good enough and have the acceptable quality when the generation finishes. One example is generating fractals to generate terrains [50]. On the contrary, generate-and-test method, as it can be seen in it's name, does sequence of tests to make sure its in good quality after the generation. If test fails, algorithm discards the generated content and continue to testing until the final content verified. Evolutionary algorithms are candidates of `generate-and-test` type of PCG because fitness function or evaluation function of some kind is applied after generation and appropriate content is selected from candidates depending on the tests.

2.2.9 Outline of PCG

Here we summarize and outline the taxonomy of PCG under five main groups.

Table 2.1: Outline of PCG

CONTENT GENERATION				
TIME	**TYPE**	**FORM**	**OUTPUT**	**LINE**
* ONLINE * OFFLINE	*NECESSARY *OPTIONAL	*RANDOM SEED *PARAMETERIZED and COMPLEX ALGORITHMS	*STOCHASTIC *DETERMINISTIC	*CONSTRUCTIVE *GENERATE&TEST

2.2.10 Search-Based Procedural Content Generation

2.2.10.1 Definition

Research made by Togelius et al.[3], proposes a *"generation and test"* approach and a survey of this approach.
Search-Based content generation has two main functions:

- Test function or evaluation function doesn't rejects or accepts based on the evaluation, instead it grades the results using number or vectors.
- Production of new content always aimed to have higher value by generating new content depending on the fitness value of the previous.

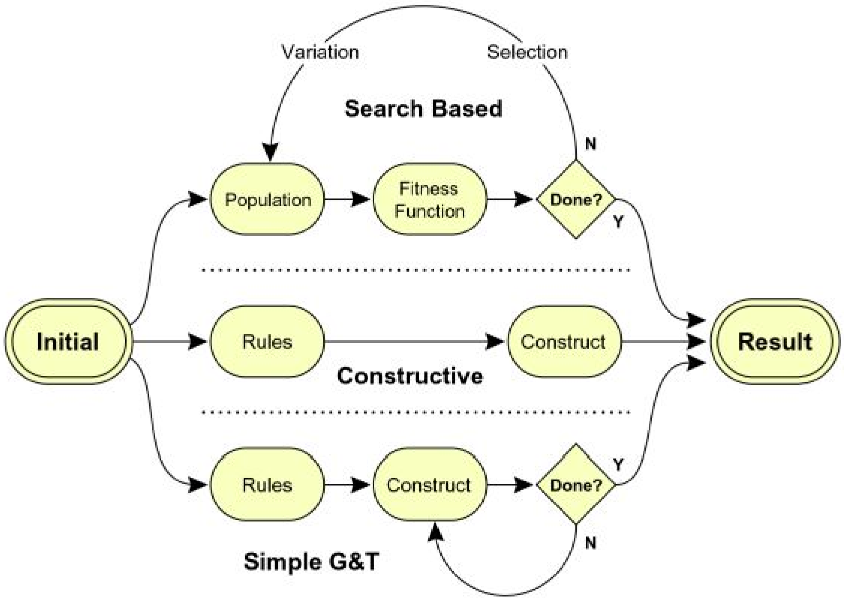


Figure 2.5: Three approaches to PCG including Search-Based by Togelius et al. [3]

2.2.10.2 Properties and Standing of Search-Based PCG

According to Togelius research, *Search-Based PCG* consist of generation approaches that use **Generate-And-Test Approach**. Choosing the right way of representation is an important issue, theoretically chosen representation should be capable of representing most possible solutions within a Search-Based PCG context. Indirect representation to a degree by storing list of positions, properties of the objects or different patterns of objects and a free space would be good candidate for search-pcg in dungeons. Usually there is no guaranteed completion time. Time that it takes mostly depends on the evaluation function. There is no guarantee that it will provide good results. Most examples takes more than a day to run which implicitly means that its less suitable for online content generation.
It can demonstrably be used for both necessary content and optional content generation Necessary content generation examples include,

- Ludi, a system for offline design of rules for board games by Browne [42]
- Zillions of Games game description language, by Hom and Marks [51]
- Genetic algorithm based puzzle generator for Shinro games by Oranchak [52]
- Puzzle generation for chess mazes and chromatic puzzles by Ashlock [53]
- Online/offline generation of tracks by Togelius et al [54, 19]

- Terrain generation for Chapas by Frade [55]
- Generating maps for RTS games by Togelius and more[56]...

Optional content generation examples include,

- Weapon generation in Galactic Arms Race by Hasting et al. [4]
- Interactively evolving buildings by Martin et al. [57]
- Controlling in game camera movements by Burelli and Yannakakis [58, 59]
- Offline manual exploration of design spaces of 3D models by Talton and more[60]...

Example implementations of Search-Based PCG can be found in following figures 2.6

(a) Ludi [42]

(b) Zillions of games [51]

(c) Offline/Online generation of racing tracks [54, 19]

(d) Galactic Arms Race [4]

Figure 2.6: Content Generation Examples

2.3 Case Studies And Related Works

To understand state of the art in PCG we have to explore the literature for applications of Procedural Content Generation. The following subsections presents, well-known case studies made in literature about procedural content generation.

2.3.1 Procedural Content Generation in Charbitat

2.3.1.1 Overview

Charbitat is an experimental game and also a case study for the PCG field. In *Charbitat*, players creates the world as they run through it[2]. There are no win or lose conditions however there are quests given to players to differ this game from open-ended simulations like Second-Life (Linden Lab). Game uses combination of random elements to shape the terrain, positioning and authored and pre-modeled objects which combined in run-time. *Charbitat* is about a princess who get poisoned and in a coma state. Like in coma state, *Charbitat* can be infinite. It has the goal of overcoming obstacles and master the world, balance the elements and leave the coma.

2.3.1.2 Mechanics

Characters steps into a border less world and when the characters steps into the void, new part of the world is generated and materialized.
Back-end is responsible for managing seed values, world generation, stabilization and saving the world.

2.3.1.3 Elemental Seeds

Whole world of *Charbitat* based on Taoist elemental system which is fire, water, earth, metal and wood. Areas reflect their properties such as fire area has bright red fire flowers and wood are has lush trees. Every entity in the world is connected to these five elements. Each entities can either be infected or healthy. Whenever player kills a predator in the world they gain strong tendency against that material. In order to guarantee coherence, elements can't jump towards contrary areas, such as tiles can't jump from water to pure fire environment. While player always influence the system, generator defines the limits and stabilize the world all the time.

2.3.1.4 Space generation

Development of *Charbitat* world depends on the player. Playing the game and changing elemental variables is the only way for the world to materialize.
Quests
Charbitat differs from open-ended simulations like Second Life by having quests and goals. Players often have to find healthy elements to cure heroine and heal poisoned elements. The player has to find a unique generated path to each ore cells. **Informing the Player**
To let players make logical choices in game **HUD** (Heads-up Display) is provided by game which shows the tendency of elemental values and relationship between the elements. Throughout the game, quests keep players motivated and procedural generation object positioning supports continually growing world. *Charbitat* answers number of design challenges in the PCG by building bridges between different approaches[2].
One of the most innovative part of *Charbitat* is mapping the character behavior into the space. *Charbitat* also gives meaning to the world generation by issuing new quests and giving players goal to accomplish.
Finally, *Charbitat* shows one example to a player-shaped procedural generation.

2.3.2 Automatic Content Generation in Galactic Arms Race (GAR)

Galactic Arms Race is another case study which is one of the most prominent examples of using IEC with Neural Networks and naturally Procedural Content Generation. The main contribution made by GAR is to show the great potential on generation based on player preferences.

2.3.2.1 Overview

GAR is single and multi-player a game where players control a space-ship and kill aliens in order to gain money and find new weapons. Every weapon found in a game is unique and generated using evolutionary programming. Players has 3 slots for weapons, can choose what weapon they can use but changing weapon in the slot removes the weapon. So they have to be careful choosing what weapon they'll use.Destroyed enemies may drop unique generated weapons. In GAR, weapons are generated based on user preferences, which means that weapon type or properties that player frequently use will be chosen as parent in the evolutionary process, and generated weapon have good chance to include some of these properties in it.

2.3.2.2 Particle System Weapon CPPNs

In GAR, each player weapon contains a single evolved CPPN. CPPNs are variation of artificial neural networks that has slightly different activation functions and applications. The main difference between ANN and CPPN is, ANN's use sigmoid and gaussian as activation functions. CPPN has the ability and advantage to use these existing effective methods in ANN and many more functions as their activation functions such as sine, cosine, tan, other periodic functions, linear functions and so on. CPPNs are applied in large spaces to represent complete images and patterns. CPPNs can output patterns in any number of dimensions by giving it's inputs in the right number of orthogonal axes[4, 61].
In every frame of animation weapons give inputs to CPPNs (as shown in Figure 2.7 as their local positions, distances from the ship. The CPPN outputs particle's new velocity and it's new color.

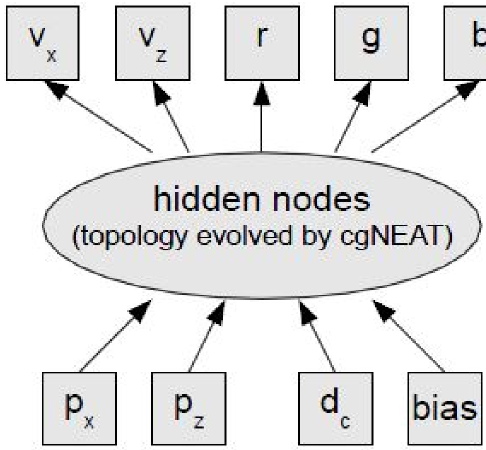


Figure 2.7: How CPPNs Represent Particle Weapons [4]

In GAR, weapons are created using **cgNEAT** methodology. *NEAT*, neuro evolution of augmenting topologies, is a method developed to solve control and sequential decision issues. NEAT outputs complex behaviors by sophisticating the network topology into diverse species[61]. cgNEAT method automatically generate computer graphics and video contents based on user behaviors in real time. cgNEAT uses implicit usage statistics to give decisions for generating new content. Especially in GAR, this method corresponds to the need of producing new weapons in real time. cgNEAT algorithm described in Galactic Arms Race implementation is as follows[4]:

- Each content is represented with using CPPN.
- During the game each item is assigned a fitness value
- Players begin to the game with new weapons that is either in spawning pool or as random content.
- Content is spawned in the game world and placed for players to pick it up. Content is not able to reproduction if players do not pick up.
- Content is reproduced and evolved in cgNEAT
- Content acquires very high fitness and increases or decreases depending on player's usage

2.3.2.3 Fitness of Weapons

As described in IEC, the fitness function here replaces the user preferences. In GAR, fitness is calculated according to the usage of weapons. When player fires a weapon that weapon is given with a constant fitness value, and other weapons in the slots lose constant fitness value.

2.3.2.4 Evolving New Weapons

New weapons are spawned when players kill enemy or stations. Weapons are spawned in three ways:

1. Reproduction using current weapon by using roulette system[62]. This way has given 80% probability. As in evolutionary algorithms, higher weapons always has higher change to selected as parent.
2. Choosing between set of pre-evolved weapons chosen by designers called spawning pool. This way has 10% probability.
3. Random weapons in the hidden nodes and with random weights. This way also has 10% probability.

Using this method novel weapons are created by *cgNEAT*. In single player mode, total weapon population is limited by player's inventory. In multiplayer version population is a very broad space of all users inventories. Results presented for automatic content generated in Galactic Arms Race can be seen in Figure 2.8.

Figure 2.8: Galactic Arms Race Weapon evolution examples with different number of generations [4]

So far, we have analyzed Procedural Content Generation in detail, now we move on to the process that's used in our core framework.

2.4 Sentiment Analysis and Sentiment Strength Detection

In our core generation framework we have performed sentiment analysis and sentiment strength detection to be able to analyze power of the keywords. In the following subsections, we'll give a background to sentimental analysis and strength detection. Sentiment analysis tries to identify the opinions from underlying text i.e. identifying movie review as "thumbs up" or "thumbs down"[63]. Opinions are central to almost all human activities. Opinions and it's related concepts are main subjects of study in sentiment analysis and opinion mining[64]. Computational analysis of opinion, sentiment and subjectivity is very important because of it's potential applications. Information extraction, question-answering, recommender systems, editorial sites, data mining services are examples of usages for sentiment analysis[63]. For the first time in human history, social media now has incredible opinionated or biased data. Sentiment analysis is now at the heart of social media research[64].

2.4.1 Sentiment Analysis Steps

Usual sentiment analysis steps can be described as;

- Classification using supervised or unsupervised learning
- Sentiment Rating prediction
- Cross-Domain sentiment classification
- Cross-Language sentiment classification

2.4.2 Different Levels of Analysis

- **Document Level Analysis** tries to analyze or classify whole opinion on the document. This kind of analysis can be too crude for most applications. In this kind of analysis task is usually meaningful if only one entity is evaluating in the document. If there is more than one entity then sentiments on the entities can be different.
- **Sentence Level Analysis** tries to determine whether sentence expresses positive, negative or neutral opinion. This analysis is closely related to subjectivity classification[65]. Fundamentally methods to analyze sentiment in document or sentence level is same. We can see sentences as a short documents. However usually sentences contain one opinion while documents contain multiple opinions. Main specific tasks on sentence level analysis is subjectivity and sentence classification, dealing with conditional sentences and sarcastic sentences.
- **Entity and Aspect Level Analysis** performs finer-grained analysis to discover what exactly people like and did not like. Aspect-based analysis deals with many phrase and

word sentiments depending on aspect contexts and focuses mainly on product/service reviews and tweets from Twitter. Entity level analysis focuses on compiling word lists and extracting opinions or polarity on words.

2.4.3 Sentiment Strength Detection

SentiStrength detection algorithm tried to detect strength of polarity in a given informal English text[66]. SentiStrength algorithm developed to address this problem is able to predict positive emotion with 60.6% and negative emotion with 72.8% accuracy based on 1-5 scale. The core algorithm uses sentiment word strength list which is a collection of 298 positive and 465 negative terms classified with values from 2 to 5. Default manual word strengths are modified by **training algorithm** to optimize the strengths. **Spelling correction algorithm** identifies standard spelling errors such as hello¿helllo. **Booster word list** contains words to boost or reduce the emotion like ”very, extremely”. **Negating word list** analyses words that invert subsequent emotion like ”very happy = not very happy”. **Repeated letters** give 1 boost to the emotions.**Repeated punctuation** gives 1 strength boost immediately. **Negative emotion is ignored** in questions like ”Are you angry?”.

2.5 Environments

We have explored all scientific background that we have based our thesis. Finally we will give a bit of information of the development environments we have used throughout this research.

2.5.1 Unity Game Engine

Unity is a modern high-end game development tool. This game engine integrates, coding, visual design, animation, sound and couple of other suites out of the box. Unity uses Mono which is an open source .NET alternative, c#, Javascript, boo languages which are relatively the most dominant ones in programming languages. Unlike most of the other game engines. Unity game engine's development pipeline is built upon the easiness that small studios or even single person can develop games very quickly. Unity is accessible for free and pro license is also accessible for low budget studios, and doesn't take royalty fee from it's users. With it's easiness, integrated programming languages, it's integrated design suite Unity became highly popular in few years between game developers. In this thesis, client of the generation framework is built with Unity game engine with the reason of it's popularity, easiness and usability.

2.5.2 Photon Framework

Photon is one of the most powerful multi-player middleware and service specifically for Unity. Photon's notable features are:

- Realtime, photon uses fast and lean protocols to use minimal bandwidth and allow fast serialization
- Highly Scalable, load balancing offered by Photon helps to scale applications to thousands of concurrent users
- Multiplayer, Photon enables you to make the choice between HTTP, TCP, UDP, RUDP

CHAPTER 3

PROPOSED METHOD

3.1 Concept

Procedural Generation Methods are used as alternative for adding manual content. While it does takes burden of adding all contents, the other potential of PCG helps cannot past unnoticed. As mentioned in Chapter 2, procedural content generation has a potential to create truly endless content for games which can produce an ever fresh game for all the players. The reason we propose this new approach for the PCG is to reveal new ways not only generating content using randomized algorithms, but also let games to generate new content based on the actual content rendered from real world.
The tremendous data flow of social networks can help us to generate content like triggering & spawning events, choosing and altering themes based on irrelevant reaction from masses to the events happening in the world such as natural disasters, politics, seasonal changes and much more.
In the scope of this thesis, we will perform data mining on Twitter and build a framework for the purpose of showcasing usages of this new approach and apply a questionnaire to find its potential applications. In this section we will explain the concept of this approach, architecture of developed framework that has the ability to trigger events, generate new themes, quests, stories, new materials, textures and such based on seed keywords and associated words defined by framework.

3.2 Conceptual Design of Context Based Approach

The very first thing to consider before diving into details and architecture of the generation framework is the new approach of context based dynamical generation. We propose an approach that potentially has wide variety of usage to generate truly refreshing content.
Randomized approaches has been used as a main generation approach for decades. One of the main problem we find with the randomized content generation is irrelevancy of pseudo ran-

domization. There are other methods such as experience driven, interactive evaluation based methods but those are restricted into the small space which is inputs gathered from in-game or players. However, the flow of the real world actually far more extensive and can be much more meaningful if we consider today's actuality. We use social networks to instantly get actual data happening in the world. From that perspective, content generators can also be able to get all the events happening in much more larger space and project it into the games and applications. In this thesis, we use Twitter for this approach because of its high public accessibility that is the main requirement for this approach.
Using twitter, we expect to get most of the events posted throughout the world. Firstly, we define an example heuristic model to classify events as discussed in 3.2.1. Then we let twitter to take the turn and cluster bunch of streaming tweets into these predefined classes. With this way, we are not only protecting the keywords and classes from extinction but we are also expanding our dictionary and keywords and generate them in our game space. Finding related and trending keywords is discussed in detail in the architecture of generation framework section.

3.2.1 Heuristic Model

3.2.1.1 EVENTS

Natural Events & Disasters
Natural disasters are events that are results of natural processes.

- Volcano
- Earthquake
- Tsunami
- Blizzard
- Hailstorm
- Tornado
- WildFire
- Star Rain
- Avalanche are basic event examples we used on our game.

We use these events in their own natural environment. In the examples on Figure 3.1, we used tornado, earthquake in Desert Dungeon and blizzard, hailstorm in Ice dungeon.

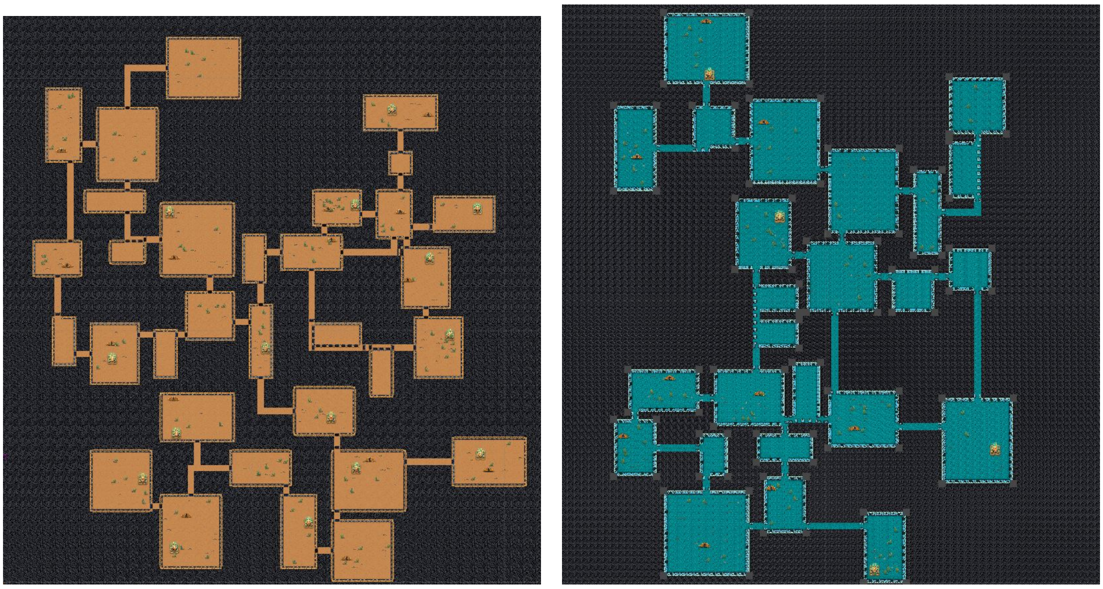


Figure 3.1: Desert & Ice World Dungeon Map

After selecting keywords we add this keywords to our library and generate new properties for these keywords. This generation of properties is able but not limited to produce colors, textures. materials, vectors, parameters. With this approach we do not only pre-define every potential keywords but we ensure continuity of these keywords most of the time. Only exception appears when we enter keywords that all of the synonyms of this keywords can't be found or used very little in the twitter. But we can argue that if this keywords are not appearing in twitter anymore these kind of keywords are starting to fade away from everyday life. We continue to follow these keywords for a chance that they might re-appear and started to be used frequently.
Examples of events being triggered in game can be seen in Figure 3.2 and Figure 3.3.

Figure 3.2: Natural Disaster Events, Volcano, Tornado, Tsunami

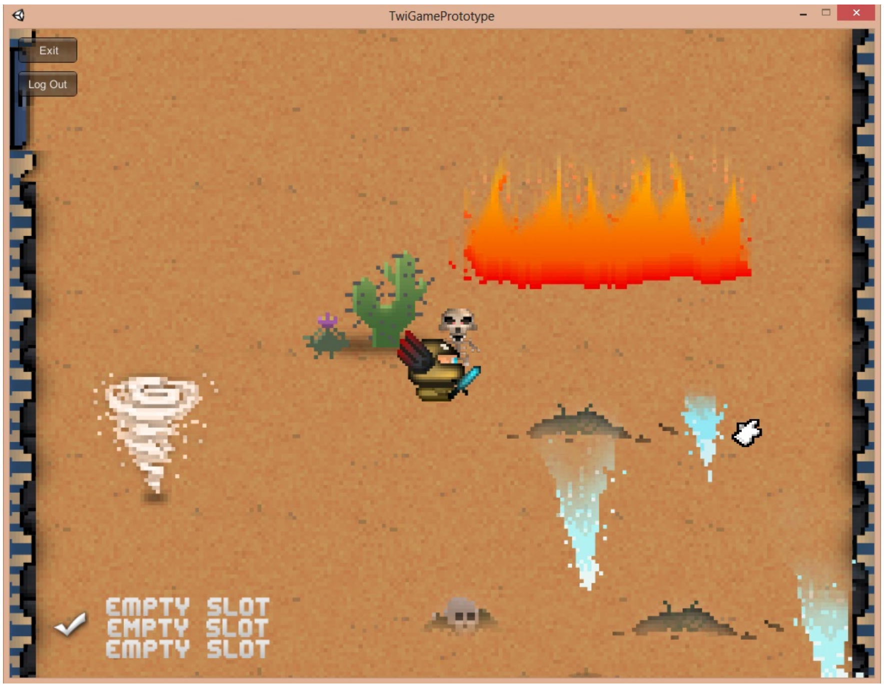


Figure 3.3: Natural Disaster Events, Fire, Tornado, Blizzard

3.3 Generation Framework

Generation framework is developed to demonstrate some of the potential usages of the generation approach.
Sandbox is an example framework (interface in Figure 3.1) application used as an authoring tool for applications that uses the new approach we are introducing. Developers can enter new seeds, inputs, modify current ones and changes made by authors happens in game real-time.

Figure 3.4: Sandbox User Interface

3.3.1 Architecture

Sandbox is able to communicate required application or game server that is developed using Photon Framework to transfer input parameters given by game developers. After input parameters given, the process starts on the server side.
Before going into generation process in detail, we will explain base process that take place in each generation.

3.3.1.1 Step1 : Keyword streaming & association

Generation framework generate content based on keywords that is either given by developers or associated keywords mined by twitter. That is why keyword streaming and association is one of the most common and core process in our content generation pipeline.
When keyword is entered and sandbox transferred these keywords to the server, server starts to search for synonyms on the thesaurus for each keyword. When the synonyms are found, we search the keyword and its synonyms on Twitter using Twitter Search API. While doing this search, developer has the ability to restrict the search in whereabouts of selected location, language and time interval. From twitter search we get (if there is that much data) between

60-100 latest tweet, which is a 140 character string posted by individuals to twitter, for each keyword.

3.3.1.2 Step2 : Sentimental Analysis

After tweets are gathered from twitter, each tweet is analyzed sentimentally and it's sentimental strength is returned depending on that tweet's context. Sentimental analysis and strength detection is processes is discussed in Section 2.4. Mean polarity is calculated for each context and associated final polarity result is associated with each keyword.

3.3.1.3 Step3 : POS, Part-of-Speech, tagging

After sentimental analysis we run POS-Tagger on each of these tweets.
A Part-Of-Speech Tagger (POS-Tagger) is a piece of software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc., although generally computational applications use more fine-grained POS tags like 'noun-plural'. [67]
After all the strings are tagged by POS-Tagger, we exclude some parts from these strings to select only reasonable keywords in our arsenal. Frequencies of all these keywords are also calculated in this process. Finally, by running a k-means frequency election, we gather keywords that are used more than an average frequency threshold. In this process, it's implied that, after excluding of meaningless words from strings, remaining words, that used very frequently, has some kind of affinity or relationship with our seed keywords. Thus, using these new keywords to generate new content can more likely create new content depending on our context. Additional to generating related content, mining this associated keywords also prevents the seed keywords triggering the events if they extinct or does not appear in twitter. For example let's say lucifer is an enemy spawning whenever lucifer keyword is spotted on twitter. In time, lucifer may not appear on twitter, but demon, devil etc. may still used very frequently. If these keywords are associated to the lucifer, we, usually, prevent lucifer from not spawning ever again.

3.3.1.4 Generation Pipeline

Generation Module handles main processes in generation pipeline to generate events and themes. Basically events are actions that happen dynamically in games. Using Sandbox, developers are able to search and stream keywords through twitter and direct the in-game event generation to twitter. Using this feature, developers has the power to connect all in-game events with real world events. For example, when there is earthquake happening on the twitter, earthquake can also happen inside the game. Additionally with the location restriction, developers can even make earthquake happen in X, Y location in game world, when earthquake is happening in

Japan, or even Fukishima.
Keyword streaming and association is in the core of this process. After keyword streaming is started, event generation mechanism only responsible to send message from game server to the game clients about happening event and does not generate any additional content except new keywords generated in the keyword association. It is developer's responsibility to handle these messages coming through the Sandbox Servers. For gathered keywords that's associated with seed keyword, developer can choose whether to send the same message or message with new keyword to server. This can help include additional context mined from social network such as starting a war irrelevant to the which war happening on twitter or starting a war depending on the war happening in real world.
Core generation module also responsible for handling connection between in-game events and in-game assets like in 3.5. Developers can browse and select external/local prefabs or enter name of the internal prefabs to be used when the event is happening which can be seen in 3.6. If external prefabs are selected, these prefabs are uploaded to server and photon server handles the updating progress in all clients.

Figure 3.5: Core Generation Module, Internal Items

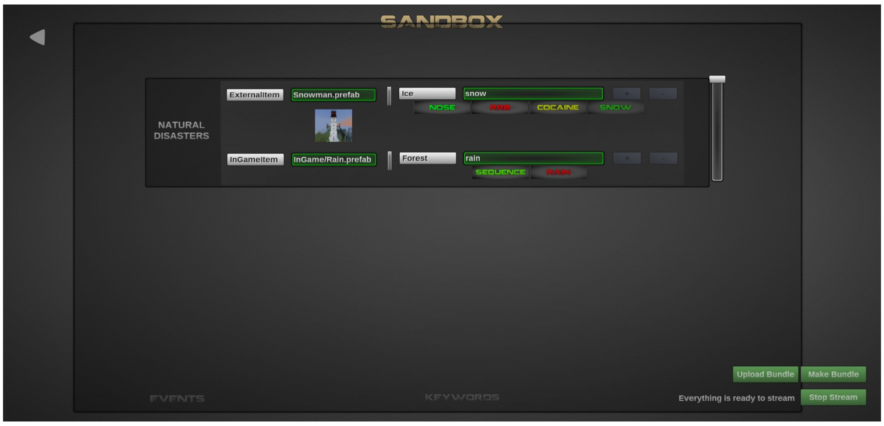


Figure 3.6: Core Generation Module, External Items

3.3.2 Application of the Approach: Football Manager

3.3.2.1 Description

With the purpose of transferring our approach into practice, we have chose to modify an Open-Source football manager game, and alter the performance of each players in a team depending on the data we gathered from twitter.

3.3.2.2 Details

In this implementation, each player's performance is altered by dynamically gathering the tweets on twitter that mentions the player's first and last name. Weather forecasts based on team home locations also changes specific skills of football players in the team.
In the football manager application, skills of a player in the team is as follows;

- Morale
- Passing
- Teamwork
- Ball Control
- Throw In
- Dribbling
- Crossing
- Zonal Marking
- Man Marking
- Rushing Out
- Handling
- Shooting
- Pace
- Heading
- Right Foot
- Left Foot

3.3.2.3 Altering Player Skills

Each skill does different contributions to football game.
Using sentimental strength gathered for each player in the team, we alter the Morale of the players. This way, when crowd talks about any player, we are using that data to alter the morale of the player. When morale of player is altered, it effects player's overall performance since morale directly effects all of the skills.

Heuristic

While altering skills using morale, we used heuristic value which we defined by our experiences in following equations.

$$skill_{alt} = skill_{def} + (skill_{def} * morale/2) \tag{3.1}$$

Where:

- $skill_{alt}$: is altered player skill
- $skill_{def}$: is default skill of player
- $morale$: is the mean polarity value gathered from our generation pipeline

$$skill_{def} + skill_{def} * (1/weather_{coeff}) weather_{coeff} = \begin{cases} weather_{coeff}, & \text{if } team_{away} \\ weather_{coeff} * (weather_{coeff}/2), & \text{otherwise} \end{cases} \tag{3.2}$$

Figure 3.7: England vs. Uruguay morale and match result after modification

3.3.2.4 Dynamic Weather

Weather forecast on city that match is played is designed to effect some specific skills of player.
Rainy Weather, negatively effects Ball Control, Passing, Dribbling
Snowy Weather, negatively effects Ball Control, Passing, Dribbling, Rushing Out, Shooting, Pace
Windy Weather, negatively effects Ball Control, Passing, Dribbling, Throw In
Sunny Weather, positively effects Ball Control, Morale, Rushing Out

3.3.2.5 Testing Results

We have divided testing part into 2 sections.
First, we have tested the applicability of our approach and implementation. To do that we have tested reaction of twitter users, rate of change in sentimental data over time and our application
In second part, we have applied two questionnaires. First questionnaire that is shown in Appendix A is applied to wide range of players and developers that consist of 120 people. Main purpose of this questionnaire was understanding how current dynamical approaches perceived by users, does our approach makes any difference, in what genres and what ways we can explore this approach in the future. Second short range questionnaire that is shown in Appendix C is applied to initial test group of gaming experts which consist of 20 people with the purpose of understanding preference over traditional methods and our applications.

3.3.2.6 Part-1

Testing Reaction: To test how crowd reacts on real football we checked the match data and overall player reaction of Twitter users based on those days. In the Figure 3.8, data gathered for Turkey football team players are shown after Turkey - Brazil match on Nov, 12 2014 6.30 PM GMT. Final score of the match is Turkey 0 - Brazil 4. After this match, we can see that sentimental analysis for Turkish players are negative overall except some players who shown good performance in the game, and sentimental analysis for Brazil seems to be positive.

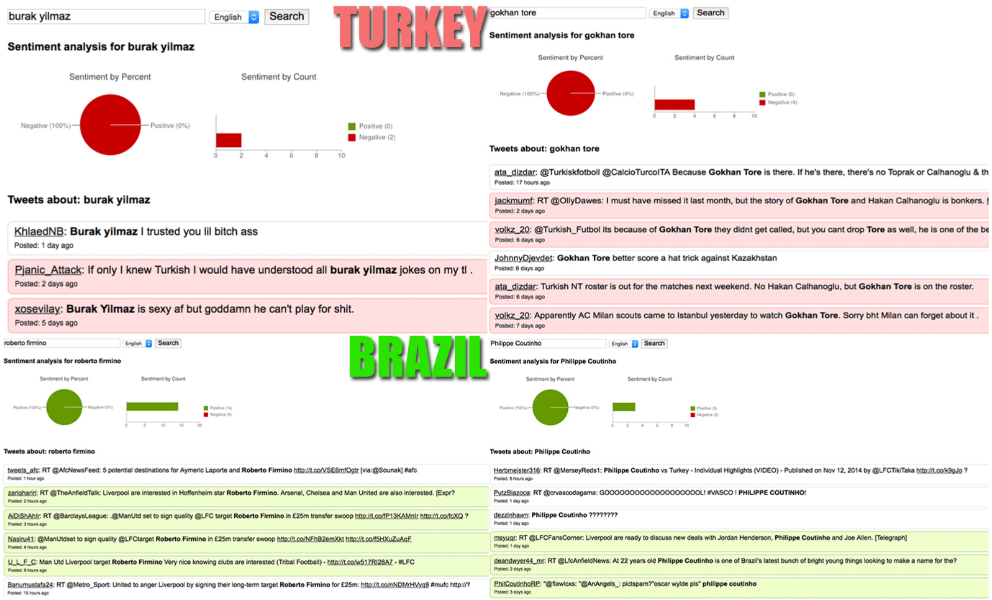


Figure 3.8: Turkey Key Players vs. Brazil Key Players after 0-4 match on 16th Nov 2014

Testing the Rate Of Change in the Sentimental Data Over Time

To understand usability of this approach, we need to find out the range we can reasonably measure changes on the sentimental analysis data. To test the changes in sentimental analysis results, we gathered sentimental analysis results for 6 football players (3 well known, 3 moderately known), 6 nba players (3 well known, 3 moderately known), 5 nfl players (3 well known, 2 moderately known) 3 well known F1 drivers and finally 1 dota e-sport player. In first 8 run, we have gathered 1 sentimental data in 3 days, in the next 19 run, we have gathered 1-4 sentimental data for each day. Results of these data is shown in the figures below.

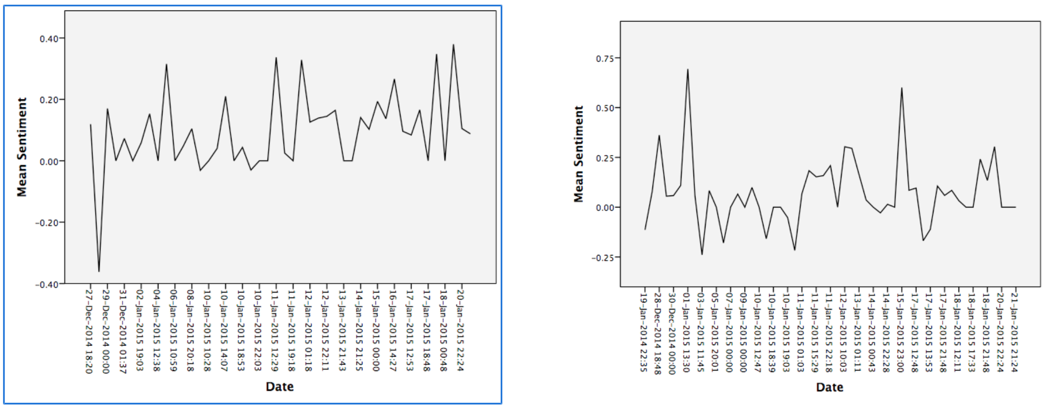


Figure 3.9: Sentiment analysis of Soccer Players: Messi, Ronaldo

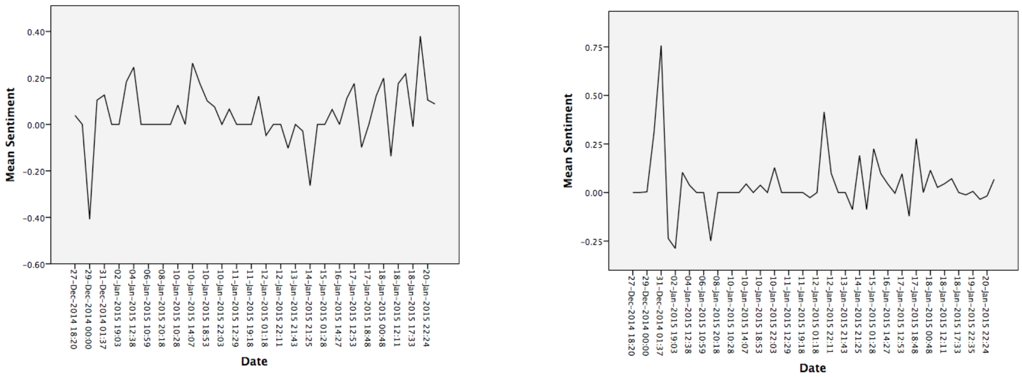


Figure 3.10: Sentiment analysis of Soccer Players: Ibrahimovic, Persie

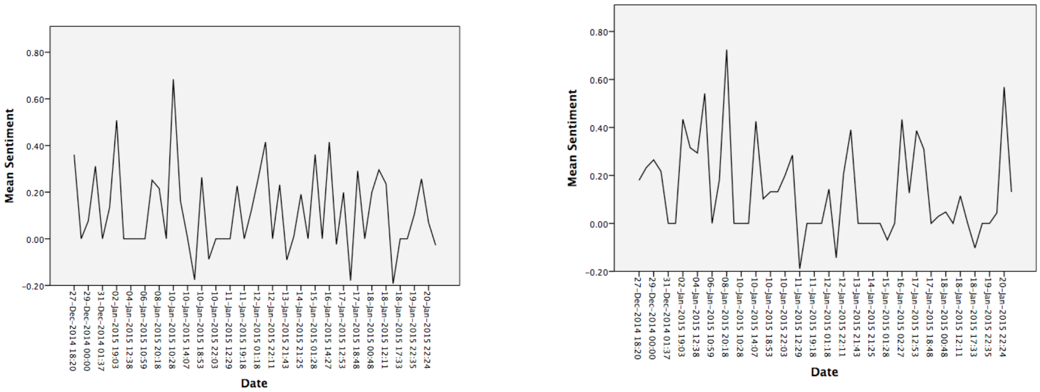


Figure 3.11: Sentiment analysis of Soccer Players: Hazard Ramsey

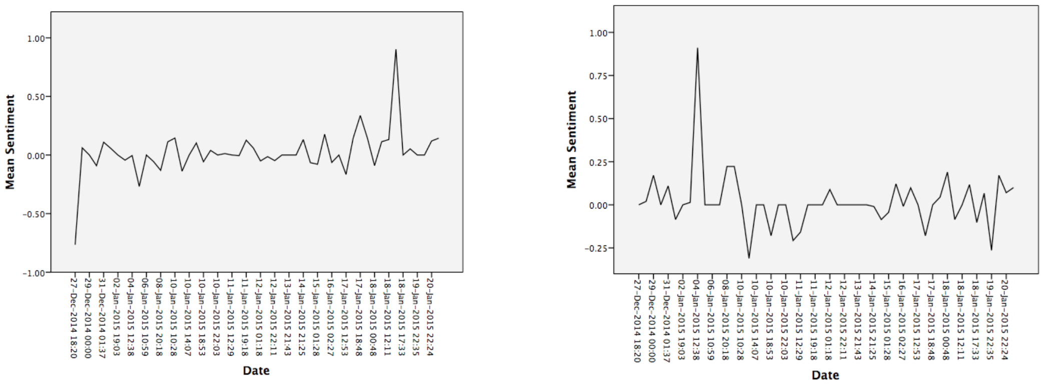


Figure 3.12: Sentiment analysis of NBA Players : LeBron James, Dwayne Howard

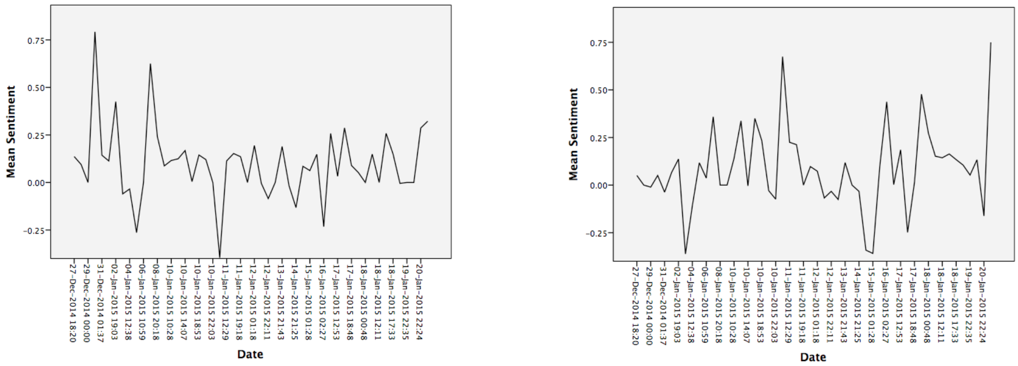


Figure 3.13: Sentiment analysis of NBA Players : Kobe Bryant, Noel

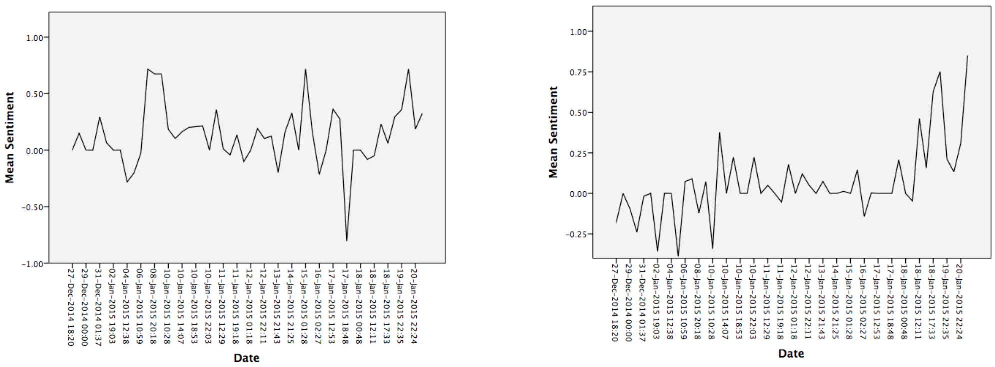


Figure 3.14: Sentiment analysis of NBA Players : Paul, Waiters

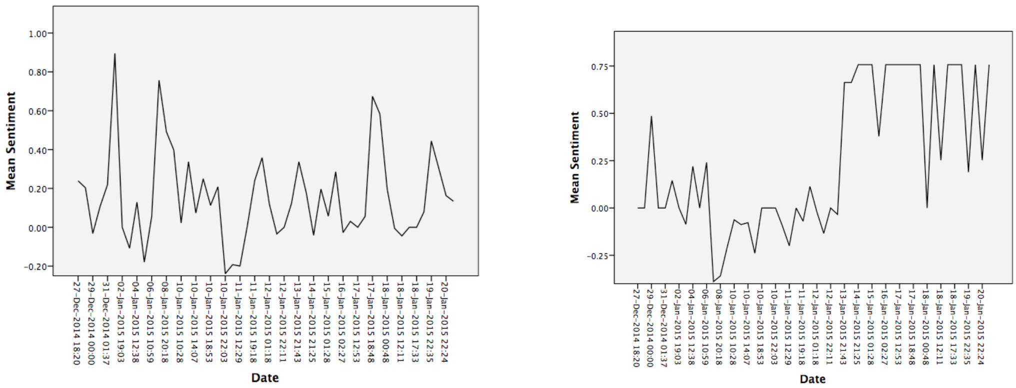


Figure 3.15: Sentiment analysis of NFL players: Brady, Johnson

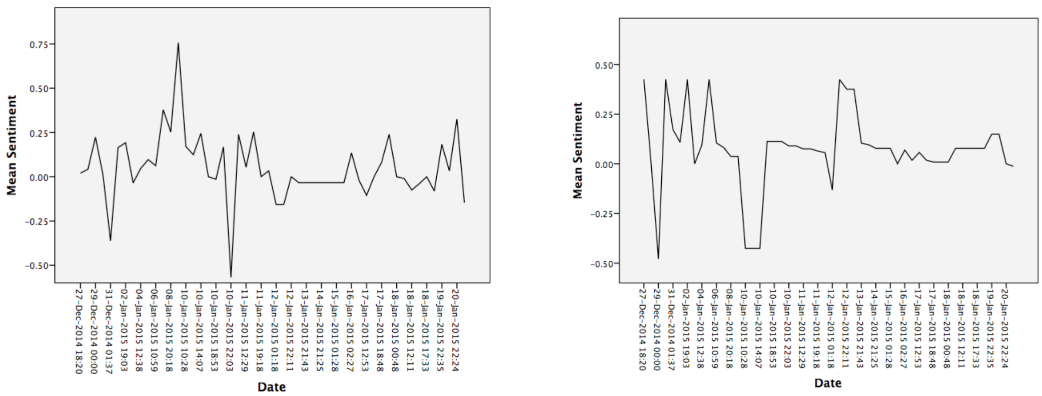


Figure 3.16: Sentiment analysis of NFL players: Manning, Vasquez

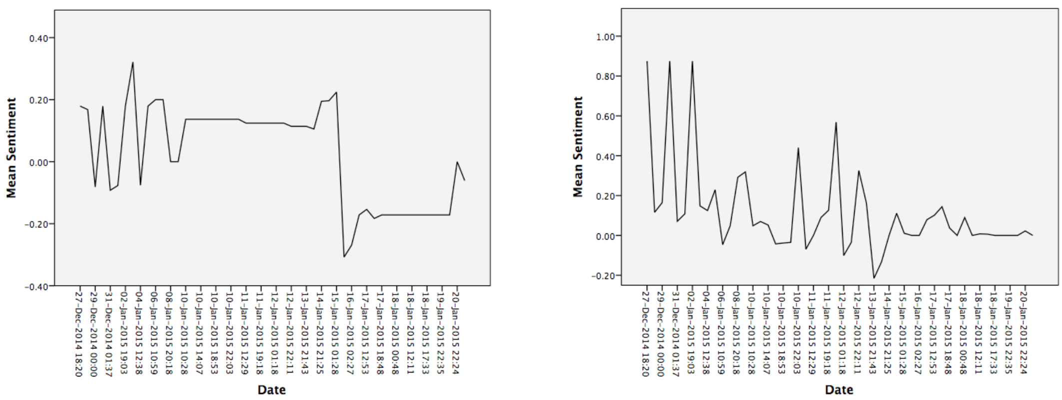


Figure 3.17: Sentiment analysis of NFL players: Washington, Witten

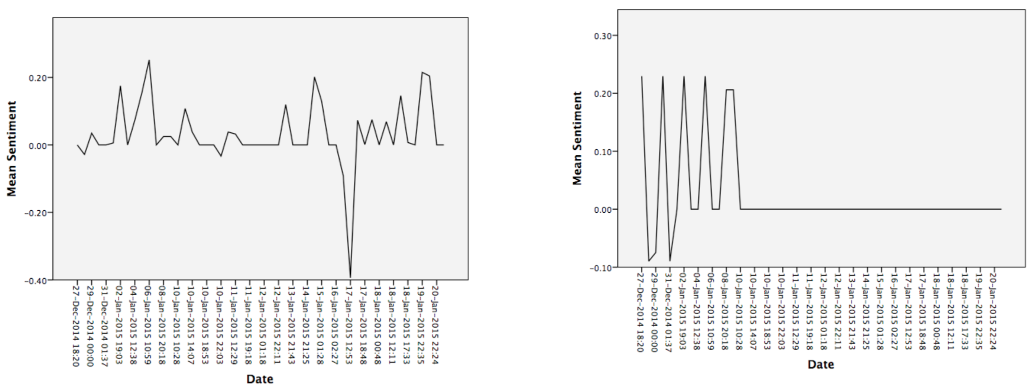


Figure 3.18: Sentiment analysis of F1 drivers: Alonso, Revson

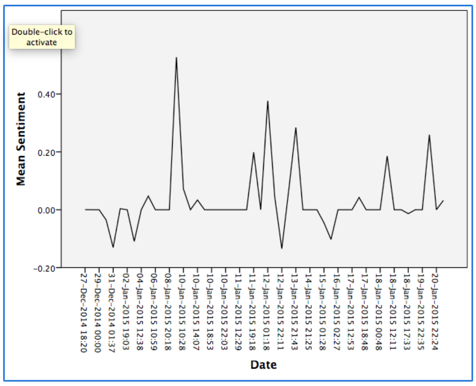


Figure 3.19: Sentimental Analysis of Dota player Dendi

By looking at the charts, we can see that the rate of change is varying a lot even when we get data in each hour a day. Which we can say, it would change the performance of the players very quickly if it's implemented in any product which can be played continuously like Fifa or Football Manager.
We have also tested our application's reaction based on the altered morale. We ran the football simulation 40 times with same teams and gathered the results which we can see in figure 3.20.

	BRAZIL	GERMANY			BRAZIL	GERMANY
	2	2			2	3
	2	1			1	3
	1	1			1	3
	1	2			1	1
	3	3			2	2
	0	1			0	2
	0	0			1	2
	1	0			1	2
	2	0			1	3
	2	2			2	4
	2	1			4	4
	3	2			2	0
	2	1			1	3
	2	1			2	5
	2	2			3	4
	2	2			1	4
	1	1			2	3
	0	2			1	2
	2	2			1	1
	2	0			2	2
TOTAL SCORE	32	26			31	53

Figure 3.20: Brazil vs. Germany (Morale 100%-100%) , Brazil vs. Germany (Morale 100%-170%)

In first 20 simulations, morale of all players are set to default 100% . In next simulations, second team's morale is boosted offline to be able to test how morale effects overall game. Looking at charts, we can see that while Germany scored 26 goals in the first simulation, they were able to score 53 goals in the second simulation.

3.3.2.7 Part-2

Questionnaire 1: Dynamical Context Generation in Games
In this part, our main aim was to understand how current dynamical approaches perceived by users. How our approach makes any difference related to other ways of dynamical generation and in what genres it worth to explore additional ways to apply our approach.
Questions can be found in Appendix A.
Questionnaire 2: Preference Test

In this part we tried to understand the preference on our approach vs. traditional approaches over small group of gaming experts.
Questions can be found in Appendix C.

CHAPTER 4

RESULTS AND DISCUSSION

In this part, we have analyzed and discussed the results of two questionnaires applied to test subjects. By doing this, we will try to understand effectiveness of our approach and also find new ways that our research can advance and implemented.

4.1 Questionnaire 1 : Dynamical Context Generation in Games

This first questionnaire is applied to 120 subjects that consist of game developers and people who plays game at least one hour a week.
Question answers can be found in Appendix B.
Looking at initial questions, we can see that most of our test subjects are mid-core players with 48% who play video games between 10-20 hours in a week. Following with 31% casual players who plays 1-10 hours a week, 21% hardcore players who play 20+ hours a week.
More than 30% of players share posts on social networks frequently. Mostly they like to share about games, family, news, tech and sports. They like to follow movies, tech, historical events, science and politics from social media.
By making a relationship test between questions "How often do you share posts on social networks?" and "What type of content would you like to share on social networks?", we found out that 93% of the subjects that always share on social networks, share posts about games. Looking at the most loved genre, 25% of the subjects like Action genre. Puzzle and RPG games following this trend by 16% and 15% respectively.
Next we tried to understand subject's perspective on dynamic content generation in games.
In the open-ended question about using dynamical generated content for everyone's most loved genre, we applied sentimental analysis using iDolOnDemand (http://www.idolondemand.com) and found out that overall reaction was positive with aggregate score of 0.581.
By looking at the details, most of the people find procedural game generation innovative and refreshing if it is done right. They emphasize on adaptive game mechanics, free character development. Some believe it adds the sense of immersion and realism. On the other hand, some of them says auto generated content lacks originality and can be repetitive and generated content needs to be more original and different. Generally, most people like the idea of making

the game less repetitive more original and as less static as possible. To explore our research through different genres, we tried to find out which game genres people think it needs more dynamic content. People find MMO as the most important genre. Following by Sport games, Action & Puzzle games, Adventure, RPG, Moba, Hack & Slash and Arcade.
Since our approach is tested on sports and RPG games, we wanted to learn subject's perspective on Sports and RPG games.
Most of our subjects find plays sports game on their spare time. 26% of them find it really boring and 18% claims to always discover new things in a game. Only 15% of players play long time without getting bored.
They think changing team performance or player performance is one of the most important option after visuals following by events and weather conditions and find environment to be the least important.
Dynamically modifying team performance & weather based on real data from social networks found to be by 51% of subjects. 28% given neutral response, following by 17% that's thinking that there's some room for improvement.
We have applied Chi-Squared test between the players who think that Sport is one of the most important game genre that needs to dynamic content to be added and enhanced and their opinion on dynamically modifying team performance and weather changes based on real data. **P-Value** which measures statistical significance, indicating whether the relationship between two variables is consistent enough that it is unlikely to be a coincidence was *0.468*, so we can say that there is no statistically significant relationship. We found out that, 41.5% of subjects who thinks adding dynamic content most important to sports game also finds the idea of modifying team performance and weather fascinating.
We have also explored relationship between the choice of **sports** in game genres people think that needs more dynamic content and the current status of Sport games using recommended Chi-Squared test. **P-Value** was 0.146 which we can say there is no statistical significance. However, similar to previous analysis we found out that people who like sports game and play it all the time, finds the idea of dynamically adding new content most important, on the other hand people who gets bored quickly from sports game, do not really care about adding dynamical content to this genre since they're more attracted to other genres.
33% of our subjects were playing RPG long time without getting bored. 23% always discover new things and other 23% play on their spare time. Only 9% finds RPG games really boring.
Our subjects believe that story and the mechanics are most important things to change based on real life consequences. Following with Events and Enemy and lastly Environment. 50% of the subjects find dynamically generating events and triggers idea fascinating. Following with 23% who thinks there's some room for improvement and 18% have neutral opinion.
We have explored relationship between the choice of **RPG** in game genres people think it needs more dynamic content and subject's opinion on dynamically generating events such as earthquake, blizzard, tsunami etc using Chi-Squared test. **P-Value** was 0.00831 which we can say that there is a strong statistical relationship. Similarly to sports genre, we can see that subjects who thinks RPG is the most important genre that needs dynamical content finds the idea of dynamically generating events fascinating. Finally, we have applied Chi-Squared test between the subject's opinion on dynamically generating events such as earthquake, tsunami, eclipse.

blizzard and the question how do you find current content of RPG games today. **P-Value** was 0.000149, which we can say there is statistically significant relationship between the questions. We can also see that people who likes the current status of RPG games, finds the idea of dynamical generation fascinating although, people who don't like RPG games does not really care about dynamically adding content.

4.2 Questionnaire 2 : Preference test for Context Based Procedural Content Generation Implementation

This second questionnaire is applied to 22 game experts that's interested in Football and RPG games.
During these tests, subjects are chosen and given information about the games. From 4 prototypes, 2 of the prototype's information was given correct and 2 of them incorrect. There are two reasons that we are not doing blind test. First one is to really understand their preference on such games, we believe it is not possible to getting bored from any game initially, our approach works eventually when players start to feel that game is monotonous and always acting randomly. Second one, most of the PCG games are marketing the idea of PCG and fresh content, we also believe that marketing game adds great value to product and changes the user's preference. That's why we used 4 prototypes. In these prototypes;
Prototype-1 : was randomly generated and given information to all subjects that it's randomly generated,
Prototype-2 : was using our approach and implementation and also told all subjects that it's using our approach,
Prototype-3 : was using random generation but the information given to users was it was using our approach
Prototype-4 : was using our approach but the information given to users was it was using random generation.
During the tests, all subjects answered choices randomly to make sure changing the order doesn't effect the overall result. The template of the question was, *please rank the prototypes from close-to-real to random* and *please rank the prototypes according to your preference* Answers to these questions can be found in Appendix D By looking at the charts, for Trending Dungeon application we can see that 54% find the events happening close to real world, at the same time Prototype 2 is the most preferred prototype as expected.
However when we look at the prototypes 3 and 4, we find that answers of prototype-3 and prototype-4 really close. 40.91% of subjects find Prototype-4 interesting and 31.92% of people find it **random**. At the same time, 40.91% find Prototype-3 random and 36.36% of subjects find it **more real than random**. From this, we can deduce that in Trending Dungeon, if we were to apply traditional methods and tell the people that we're using Context-Based Dynamical Generation, it'd only affect 9-10% of people at most if they would prefer the game which generation is being done closer to reality. However, when we look at their preference over games, subjects seemingly prefer Prototype-4 over Prototype-3. From this, we can understand that even though subjects didn't know that Prototype-4 is using our approach, they liked our

approach better than randomly generated approach. We can also easily see that if there's an option between two games, none of them was preferring the first prototype compared to others. When we look at the football manager application, we can see that it becomes a lot more apparent that people find the fourth prototype much more closer to real than prototype three, even if we given the complete opposite information. Unlike Trending Dungeon, subjects seem like they're interested a bit more from Prototype-1 in the Football Manager application. Most subjects still prefer Prototype-2 as their best prototype which we can realize that if done right, our approach brings a new excitement and more preference to both game genres.
Additional to the analysis above, we have applied Paired Sample T-Tests using IBM-SPSS with following hypothesis H_0 : Given info does not affect the results
H_a : Given info does affect the results In first paired T-Test, we have tested answers given to Prototype-1 & Prototype-3,
In second paired T-Test, we have tested answers given to Prototype-2 & prototype-4 and tried to understand if user choices affected by our info.

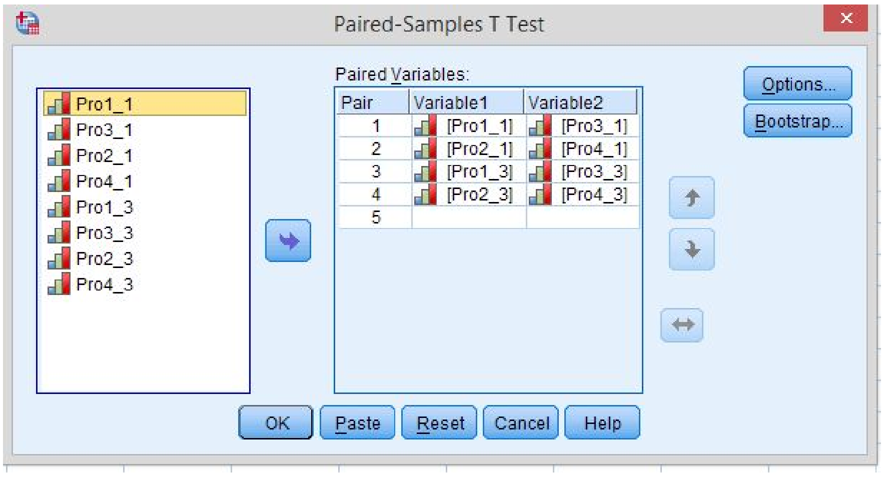


Figure 4.1: Pairs in T-Test

T-Test

[DataSet0] C:\Users\Burkay\Desktop\SPSS\Preference Answers_1.sav

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	Pro1_1	.5455	22	.59580	.12703
	Pro3_1	.7273	22	.93513	.19937
Pair 2	Pro2_1	.9545	22	1.21409	.25885
	Pro4_1	.4545	22	.73855	.15746
Pair 3	Pro1_3	.5000	22	.59761	.12741
	Pro3_3	.6818	22	.77989	.16627
Pair 4	Pro2_3	.9545	22	1.09010	.23241
	Pro4_3	.7727	22	1.06600	.22727

Paired Samples Correlations

		N	Correlation	Sig.
Pair 1	Pro1_1 & Pro3_1	22	.365	.095
Pair 2	Pro2_1 & Pro4_1	22	.661	.001
Pair 3	Pro1_3 & Pro3_3	22	.562	.006
Pair 4	Pro2_3 & Pro4_3	22	.687	.000

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
					95% Confidence Interval of the Difference				
		Mean	Std. Deviation	Std. Error Mean	Lower	Upper			
Pair 1	Pro1_1 - Pro3_1	-.18182	.90692	.19336	-.58393	.22029	-.940	21	.358
Pair 2	Pro2_1 - Pro4_1	.50000	.91287	.19462	.09526	.90474	2.569	21	.018
Pair 3	Pro1_3 - Pro3_3	-.18182	.66450	.14167	-.47644	.11280	-1.283	21	.213
Pair 4	Pro2_3 - Pro4_3	.18182	.85280	.18182	-.19629	.55993	1.000	21	.329

Figure 4.2: Results of Paired Samples T-Test

By looking at the results in the first question, which is ranking RPG game prototypes from Random to Close-To-Real, we see that paired T test between 1 and 3 does not have statistical significance which means users are not affected by tricky questions, in second question we see that the T-tests results are significant, which doesn't support our hypothesis, however this is also expected as we explained why dynamical generation in RPG game was not enough to correctly differentiate between reality and randomness. This also means that dynamical content generated for RPG games has to be explored in more detail.
By looking at the results in the second question, which is ranking sports game prototypes from Random to Close-To-Real, we see that both results in 1-3 and 2-4 are not significant, which supports our expectation on given info does not affect user's perception on closeness to reality. As we concluded above, users felt the difference a lot more apparent in sports genre.As a result we can't reject the null hypothesis and results supports our expectations.

Last two questions are answered after we told subjects the how Prototype-3 and Prototype-4 real generation algorithms work.
In **Trending Dungeon** 50% of the people said that "There's some room for improvement", which made us realize that only using themes and natural events are not apparent enough for our approach in the initial test run. It may worth to test this implementation on a game which has a game-play time of at least 1 month and see the subject's reaction. We can also say that for RPG kind of games, we should be exploring new ways and features to apply our approach. Following this choice 31.82% of people find the approach fascinating.

In **Dynamical Football Manager** 54.55% of the subjects find the idea of dynamical team & weather modification based on social networks fascinating. The main reason that we see better results in Football Manager is following matches and performance of football players is much more easier than following the natural events happening in the world. Which made us realize that if we generate events location based, it might have much more sense over users, however with that way, we might be limiting the amount events too much that only few events may happen in one game which is something we should avoid. Following this 36.36% of subjects thinks that there's some room for improvement, and we find only 5% negative comments on this test. Which we can conclude that subjects that played the game find the idea of context based dynamic generation meaningful and enjoyable.

CHAPTER 5

CONCLUSION AND FUTURE WORK

Throughout this research we have introduced a new dynamical content generation approach and developed a framework.
The main idea of our approach was using the social media data, applying some processes and use that data to generate or modify content and trigger new events.
To do that we have designed a core mechanic which mine tweets that includes keywords and synonyms of keywords that we include and find new keywords that has some kind of association and/or relation to the original keyword and finally extract frequently used keywords from the final keyword list. We have applied sentiment analysis on the tweets that has include frequent keywords and defined its positive, negative or neutral effect based on the aggregated sentiment value. To test usability of this approach, we have implemented this framework in two games one of which is RPG game called Trending Dungeon and the other is Dynamic Football Manager.
In Trending Dungeon, we have used our core generation mechanism to generate events using natural disasters happening in real world and change generated content according to the social data input. We have also used sentiment analysis to define the properties of these events, such as its damage or power based on its positive or negative value. We have implemented the game to be able to trigger new events when its tweeted on twitter. Players were able to see the events happening real time.
In Dynamic Football Manager game, we have used real weather data for the home location and altered the player's performance on top of our core generation mechanism based on the sentiment value gathered from twitter. While altering player's attributes we have changed morale of the player based on our heuristic and alter the motivation of player which resulted in affecting all skills of player at the same time. Weather data affected player's attributes based on their status of being home team or away team.
After we have implemented these games, we have tested usability and capability of our approach. We have developed Sandbox to administrate and watch the events as they happen in Trending Dungeon. With the help of Sandbox we were able to watch when events triggered by tweets in the game. We have used premade visuals for the events.
For dynamical football manager, we have assigned each team's home country and gathered weather data when making match. Weather is designed to affect football player's performance

according to their home location. We have designed an heuristic for that. Validation and enhancement of this heuristic can be done in a future work by scouting real football matches and running the simulation but its outside of our context for this research.
We have also tested usability of our approach for well known and moderately known players, extracted sentiment from twitter by using our core in different time intervals and shown that change rate as a guide that shows the data varies appropriately and can be used in commercial games.
After we shown the usability, we have applied two questionnaires to understand effectiveness and people's preference and also explore new ways to research and implement this new generation technique.
On first questionnaire, we have found out that most of the subjects who like to play games also like to share posts about games on social networks. We have also seen that most of the people who like sports and RPG games thinks integrating dynamical generation/up-to-date content to their beloved genre is very important.
None of them preferred random approach compared to context based approach. Again we found out that, if any subject doesn't like the game he give only neutral response to the question about importance of integrating dynamical approach. Which brings us to the point that, by only asking as question we can't make them prefer game genre that they don't like, however if they're interested in the genre but don't like current state, we're not sure if they'd prefer the new approach or not. This can only be seen by doing another specific questionnaire designed for dedicated subjects as a future work.
In the second questionnaire we have tested preference of 22 selected game experts over 4 prototypes of our two applications that uses both randomized and context based approaches.
In **Trending Dungeon**, we have seen that least preference is the randomized version. On 2 of prototypes that we haven't said their correct generation method, we have seen that subjects nearly mixed between which one is close-to-real which one is more random, but most of them still preferred our approach without knowing. This means that our version is more preferable by gaming experts even if its not marketed with its new features. The reason they're a little confused about Trending Dungeon's reality is that we think only generating events by extracting natural events tweets happening in social networks are not enough to realize what's going on and where. One possible approach can be filtering events location by location but that should have tested because it might filter too much events for just one location. Another possible future work might be changing themes based on locations, that would possibly increase rate of realization in the game.
We also think that there's lots of ways to improve and use context based generation in RPG games. We believe that first target should be Quests and Story, especially since side quests and stories quickly bore players. Lots of research is being done in quest generation by using STRIPS or other similar technologies. Implementing our approach to generate quests & storyline should be rather simple and very good path to explore new ways of using this approach.
In **Dynamical Football Manager** we have seen that most of the subjects find the idea of altering team performance and weather data according to social networks and real data fascinating. Additional to the team/player performance and weather data, they adding dynamical generation & fresh content to visuals really important. Which is another area that we really need to

explore.
By looking at **Questionnaire 1**, we can see that subjects saw MMO as the most important game genre that needs dynamic generation and fresh content, using our approach to explore different MMO genres, especially MMORPG can help us to identify potential issues and possibly create truly refreshing MMO games in the future.
Subjects also thinks that visuals are really important category for dynamic generation. We believe our approach can also be applied on **Image generation**. Using large database of images in Google or any other search engine, we believe anyone can easily find assign images with appropriate keywords. Using image search and genetic algorithms such as Interactive Evaluation Algorithm and post processing filters, it might be possible to create images that's stylized and appropriate to any game genre in the future.
To summarize, we have introduced a new approach to procedural generation context and we named context based dynamical content generation. We have applied our approach and tested its core mechanics and its usability. We have applied questionnaires to find new possible ways to use our approach and understand preference of game experts over both generation models. We have targeted new game genre and possible features to apply our approach. We have also found out that our approach is preferred when compared to our generation models.

REFERENCES

[1] Agoston E Eiben and James E Smith. *Introduction to Evolutionary Computing*, pages 3–20. Springer, 2003.

[2] Michael Nitsche, Calvin Ashmore, Will Hankinson, Rob Fitzpatrick, John Kelly, and Kurt Margenau. Designing procedural game spaces: A case study. *Proceedings of FuturePlay*, pages 5–7, 2006.

[3] Julian Togelius, Georgios N Yannakakis, Kenneth O Stanley, and Cameron Browne. Search-based procedural content generation: A taxonomy and survey. *Computational Intelligence and AI in Games, IEEE Transactions on*, 3(3):172–186, 2011.

[4] Erin J Hastings, Ratan K Guha, and Kenneth O Stanley. Automatic content generation in the galactic arms race video game. *Computational Intelligence and AI in Games, IEEE Transactions on*, 1(4):245–263, 2009.

[5] Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 9(1):1, 2013.

[6] Michael W Martin and Yuzhong Shen. The effects of game design on learning outcomes. *Computers in the Schools*, 31(1-2):23–42, 2014.

[7] Rilla Khaled, Mark J Nelson, and Pippin Barr. Design metaphors for procedural content generation in games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1509–1518. ACM, 2013.

[8] Michael Nitsche. Video game spaces. *Image, play, and structure in 3D worlds. Massachusetts: MIT Press*, pages 6–31, 2008.

[9] Michael Edwards. Algorithmic composition: computational thinking in music. *Communications of the ACM*, 54(7):58–67, 2011.

[10] Simon Colton. Automated puzzle generation. In *Proceedings of the AISB'02 Symposium on AI and Creativity in the Arts and Science*, pages 2–7, 2002.

[11] Ian Schreiber. *Challenges for game designers*, pages 16–21. Cengage Learning, 2009.

[12] Julian Togelius, Noor Shaker, and Mark J. Nelson. Introduction. In Noor Shaker, Julian Togelius, and Mark J. Nelson, editors, *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*, pages 3–7. Springer, 2014.

[13] Julian Togelius, Noor Shaker, and Mark J. Nelson. Constructive generation methods for dungeons and levels. In Noor Shaker, Julian Togelius, and Mark J. Nelson, editors, *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*, pages 31–53. Springer, 2014.

[14] Georgios N Yannakakis and Julian Togelius. Experience-driven procedural content generation. *Affective Computing, IEEE Transactions on*, 2(3):147–161, 2011.

[15] Mike Treanor, Bryan Blackford, Michael Mateas, and Ian Bogost. Game-o-matic: Generating videogames that represent ideas. In *Proceedings of the The third workshop on Procedural Content Generation in Games*, page 11. ACM, 2012.

[16] Gillian Smith, Jim Whitehead, and Michael Mateas. Tanagra: A mixed-initiative level design tool. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, pages 209–216. ACM, 2010.

[17] Anne Sullivan, Michael Mateas, and Noah Wardrip-Fruin. Rules of engagement: moving beyond combat-based quests. In *Proceedings of the Intelligent Narrative Technologies III Workshop*, page 11. ACM, 2010.

[18] Julian Togelius and Jürgen Schmidhuber. An experiment in automatic game design. In *Computational Intelligence and Games, 2008. CIG'08. IEEE Symposium On*, pages 111–118. IEEE, 2008.

[19] Julian Togelius, Renzo De Nardi, and Simon M Lucas. Towards automatic personalised content creation for racing games. In *Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on*, pages 252–259. IEEE, 2007.

[20] Sayed Kazmi and Ian J Palmer. Action recognition for support of adaptive gameplay: A case study of a first person shooter. *International Journal of Computer Games Technology*, 2010:1–2, 2010.

[21] Gillian Smith. Understanding procedural content generation: a design-centric analysis of the role of pcg in games. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 917–926. ACM, 2014.

[22] Ruth Aylett, Joao Dias, and Ana Paiva. An affectively driven planner for synthetic characters. In *Icaps*, pages 2–10, 2006.

[23] Dajana Dimovska, Patrick Jarnfelt, Sebbe Selvig, and Georgios N Yannakakis. Towards procedural level generation for rehabilitation. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, page 7. ACM, 2010.

[24] Yun-Gyung Cheong and R Michael Young. A computational model of narrative generation for suspense. In *AAAI*, pages 1906–1907, 2006.

[25] Anders Drachen, Lennart E Nacke, Georgios Yannakakis, and Anja Lee Pedersen. Correlation between heart rate, electrodermal activity and player experience in first-person shooter games. In *Proceedings of the 5th ACM SIGGRAPH Symposium on Video Games*, pages 49–54. ACM, 2010.

[26] Niklas Ravaja, Timo Saari, Marko Turpeinen, Jari Laarni, Mikko Salminen, and Matias Kivikangas. Spatial presence and emotions during video game playing: Does it matter with whom you play? *Presence: Teleoperators and Virtual Environments*, 15(4):381–392, 2006.

[27] Nathan Sorenson and Philippe Pasquier. Towards a generic framework for automated video game level creation. In *Applications of Evolutionary Computation*, pages 131–140. Springer, 2010.

[28] Anders Drachen, Alessandro Canossa, and Georgios N Yannakakis. Player modeling using self-organization in tomb raider: Underworld. In *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*, pages 1–8. IEEE, 2009.

[29] Olana Missura and Thomas Gärtner. Player modeling for intelligent difficulty adjustment. In *Discovery Science*, pages 197–211. Springer, 2009.

[30] Anders Drachen and Alessandro Canossa. Towards gameplay analysis via gameplay metrics. In *Proceedings of the 13th International MindTrek Conference: Everyday Life in the Ubiquitous Era*, pages 202–209. ACM, 2009.

[31] Georgios N Yannakakis, Héctor P Martínez, and Arnav Jhala. Towards affective camera control in games. *User Modeling and User-Adapted Interaction*, 20(4):313–340, 2010.

[32] Simone Tognetti, Maurizio Garbarino, Andrea Bonarini, and Matteo Matteucci. Modeling enjoyment preference from physiological responses in a car racing game. In *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*, pages 321–328. IEEE, 2010.

[33] Regan L Mandryk and M Stella Atkins. A fuzzy physiological approach for continuously modeling emotion during interaction with play technologies. *International Journal of Human-Computer Studies*, 65(4):329–347, 2007.

[34] Chris Pedersen, Julian Togelius, and Georgios N Yannakakis. Modeling player experience in super mario bros. In *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*, pages 132–139. IEEE, 2009.

[35] Christopher Pedersen, Julian Togelius, and Georgios N Yannakakis. Modeling player experience for content creation. *Computational Intelligence and AI in Games, IEEE Transactions on*, 2(1):54–67, 2010.

[36] Scott W McQuiggan, Sunyoung Lee, and James C Lester. Predicting user physiological response for interactive environments: An inductive approach. *AIIDE*, pages 60–65, 2006.

[37] Héctor Pérez Martínez and Georgios N Yannakakis. Genetic search feature selection for affective modeling: a case study on reported preferences. In *Proceedings of the 3rd international workshop on Affective interaction in natural environments*, pages 15–20. ACM, 2010.

[38] Lisa A Feldman. Valence focus and arousal focus: Individual differences in the structure of affective experience. *Journal of personality and social psychology*, 69(1):153, 1995.

[39] Vincent Hom and Joe Marks. Automatic design of balanced board games. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, pages 25–30, 2007.

[40] Julian Togelius, Mike Preuss, and Georgios N Yannakakis. Towards multiobjective procedural map generation. In *Proceedings of the 2010 workshop on procedural content generation in games*, page 3. ACM, 2010.

[41] Julian Togelius and Jürgen Schmidhuber. An experiment in automatic game design. In *Computational Intelligence and Games, 2008. CIG'08. IEEE Symposium On*, pages 111–118. IEEE, 2008.

[42] Cameron Browne. *Automatic generation and evaluation of recombination games*. PhD thesis, Queensland University of Technology, 2008.

[43] Franz Aurenhammer. Voronoi diagrams-a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405, 1991.

[44] Bastien Chopard and Michel Droz. *Cellular automata modeling of physical systems*, volume 24, pages 407–433. Cambridge University Press Cambridge, 1998.

[45] Guoning Chen, Gregory Esch, Peter Wonka, Pascal Müller, and Eugene Zhang. Interactive procedural street modeling. *ACM Transactions on Graphics (TOG)*, 27(3):103, 2008.

[46] Paul Davidsson. *Multi agent based simulation: beyond social simulation*, pages 97–107. Springer, 2001.

[47] David E Goldberg and John H Holland. Genetic algorithms and machine learning. *Machine learning*, 3(2):95–99, 1988.

[48] Simon Haykin and Neural Network. A comprehensive foundation. *Neural Networks*, 2:484–545, 2004.

[49] Stuart Russell, Peter Norvig, and Artificial Intelligence. A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25:337–364, 1995.

[50] Gavin SP Miller. The definition and rendering of terrain maps. In *ACM SIGGRAPH Computer Graphics*, volume 20, pages 39–48. ACM, 1986.

[51] J Mallett and M Lefler. Zillions of games. *Available online at: www. zillions-of-games. com*, 1998.

[52] David Oranchak. Evolutionary algorithm for generation of entertaining shinro logic puzzles. In *Applications of Evolutionary Computation*, pages 181–190. Springer, 2010.

[53] Daniel Ashlock. Automatic generation of game elements via evolution. In *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*, pages 289–296. IEEE, 2010.

[54] Julian Togelius, Renzo De Nardi, and Simon M Lucas. Making racing fun through player modeling and track evolution. pages 5–8, 2006.

[55] Miguel Frade, Francisco Fernandez de Vega, and Carlos Cotta. Evolution of artificial terrains for video games based on accessibility. In *Applications of evolutionary computation*, pages 90–99. Springer, 2010.

[56] Julian Togelius, Mike Preuss, and Georgios N Yannakakis. Towards multiobjective procedural map generation. In *Proceedings of the 2010 workshop on procedural content generation in games*, page 3. ACM, 2010.

[57] Andrew Martin, Andrew Lim, Simon Colton, and Cameron Browne. Evolving 3d buildings for the prototype video game subversion. In *Applications of Evolutionary Computation*, pages 111–120. Springer, 2010.

[58] Paolo Burelli and Georgios N Yannakakis. Combining local and global optimisation for virtual camera control. In *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*, pages 403–410. IEEE, 2010.

[59] Paolo Burelli and Georgios N Yannakakis. Global search for occlusion minimisation in virtual camera control. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE, 2010.

[60] Jerry O Talton, Daniel Gibson, Lingfeng Yang, Pat Hanrahan, and Vladlen Koltun. Exploratory modeling with collaborative design spaces. *ACM Transactions on Graphics-TOG*, 28(5):167, 2009.

[61] Kenneth O Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8(2):131–162, 2007.

[62] Kenneth A De Jong. *Evolutionary computation: a unified approach*, pages 139–140. MIT press, 2006.

[63] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics, 2004.

[64] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

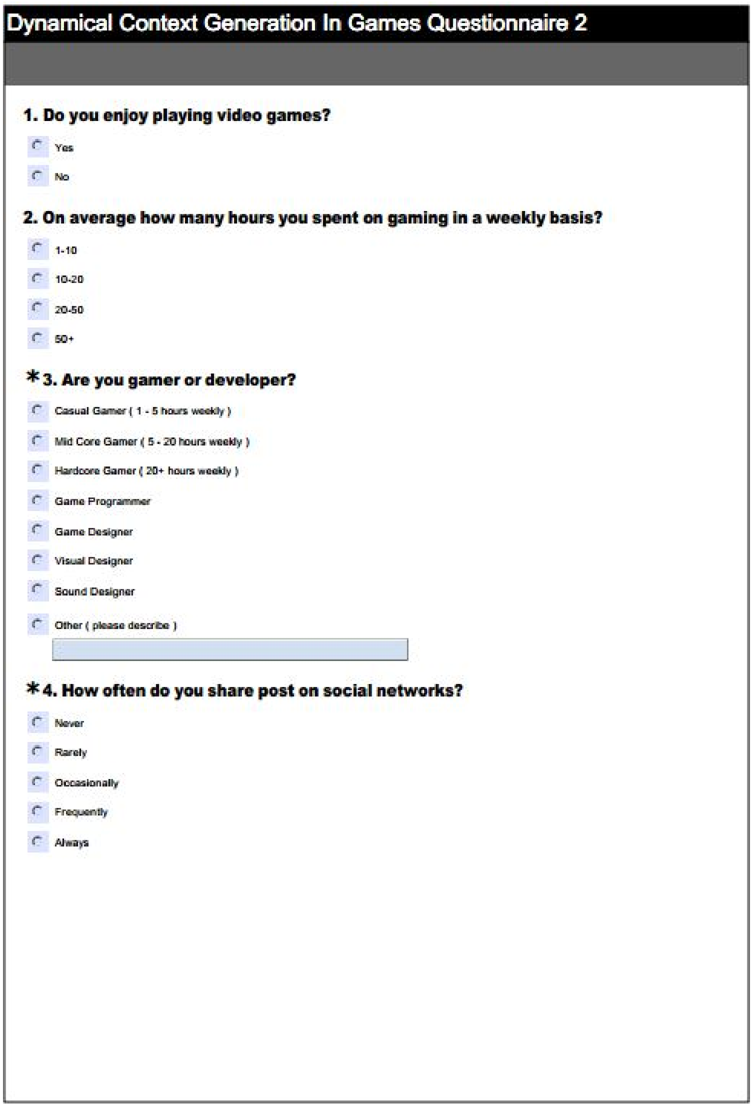
[65] Janyce M Wiebe, Rebecca F Bruce, and Thomas P O'Hara. Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 246–253. Association for Computational Linguistics, 1999.

[66] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558, 2010.

[67] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 252–259. Association for Computational Linguistics, 2003.

APPENDIX A

Appendix A



Dynamical Context Generation In Games Questionnaire 2

1. Do you enjoy playing video games?

- Yes
- No

2. On average how many hours you spent on gaming in a weekly basis?

- 1-10
- 10-20
- 20-50
- 50+

***3. Are you gamer or developer?**

- Casual Gamer (1 - 5 hours weekly)
- Mid Core Gamer (5 - 20 hours weekly)
- Hardcore Gamer (20+ hours weekly)
- Game Programmer
- Game Designer
- Visual Designer
- Sound Designer
- Other (please describe)

***4. How often do you share post on social networks?**

- Never
- Rarely
- Occasionally
- Frequently
- Always

Dynamical Context Generation In Games Questionnaire 2

***5. What type of content do you follow on social networks?**

- [ ] Festivals
- [ ] Movies
- [ ] Theaters
- [ ] Politics
- [ ] Natural Events
- [ ] Science
- [ ] Technology
- [ ] Magazine
- [ ] World Economy
- [ ] Historical Events
- [ ] Other (please describe)

***6. What type of content would you like to share on social networks?**

- [ ] Sports
- [ ] News
- [ ] Politics
- [ ] Games
- [ ] Tech
- [ ] Science
- [ ] Family
- [ ] Business
- [ ] Other (please describe)

Dynamical Context Generation In Games Questionnaire 2

7. What genre do you love playing most?

- ○ Action
- ○ Arcade
- ○ Hack & Slash
- ○ Moba
- ○ Puzzle
- ○ Shooter
- ○ Sports
- ○ Adventure
- ○ Role Playing Games
- ○ Simulation
- ○ Strategy
- ○ Massive Multiplayer Games
- ○ Other (please describe)

The term "Dynamic Content", used in question below, means, creating the game content dynamically or through algorithmic means. "Game content" includes all aspects that affects gameplay

8. Looking from game content perspective, what is your opinion using on dynamical generated content for your most loved game genre?

***9. Please rank game genres you think it needs more dynamic/up-to-date/actual content**

	Least Important				Most Important
Action	○	○	○	○	○
Arcade	○	○	○	○	○
Hack & Slash	○	○	○	○	○
Moba	○	○	○	○	○
Puzzle	○	○	○	○	○
Shooter	○	○	○	○	○
Sports	○	○	○	○	○
Adventure	○	○	○	○	○
Role Playing Game	○	○	○	○	○
Simulation	○	○	○	○	○
Strategy	○	○	○	○	○
Massive Multiplayer Games	○	○	○	○	○

Dynamical Context Generation In Games Questionnaire 2

***10. How do you find current status of Sports games today?**

- I always discover new things
- I play long time without getting bored
- I play on my spare time
- It starts to get monotonous after some time
- find it really boring

***11. In any sports game, please rank the options that you think should be changing based on real life from most to least**

	Least important				Most Important
Team Performance/Player Performance					
Weather Conditions					
Environment					
Events					
Visuals					

***12. What is your opinion on dynamically modifying team performance and weather real time using real data from social networks?**

- I find it fascinating
- There is some room for improvement
- Neutral
- I don't believe it will make any difference
- I don't think it's a good idea

Other (please specify)

***13. How do you find current content of Role Playing games today?**

- I always discover new things
- I play long time without getting bored
- I play on my spare time
- It starts to get monotonous after some time
- find it really boring

Dynamical Context Generation In Games Questionnaire 2

***14. In any Role Playing Game game, please rank the options that you think should be changing based on real life consequences from most to least**

	Least important				Most Important
Story	○	○	○	○	○
Environment	○	○	○	○	○
Enemy	○	○	○	○	○
Events	○	○	○	○	○
Mechanics	○	○	○	○	○

Other (please specify)

Term " Events and Triggers" means all of the incidents that happens in the game. Such as raining, snowing, arthquake, sudden death, ambush etc.

***15. What is your opinion on dynamically generating events and triggers in game depending on social networks (When earthquake happens in real world, earthquake also happens in game)**

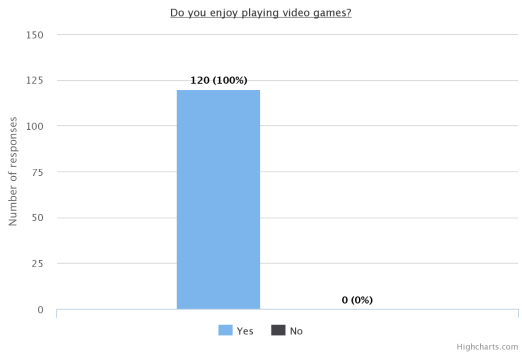
- ○ I find it fascinating
- ○ There is some room for improvement
- ○ Neutral
- ○ I don't believe it will make any difference
- ○ I don't think it's a good idea

Other (please specify)

APPENDIX B

Appendix B

B.1 Questionnaire 1: Answers



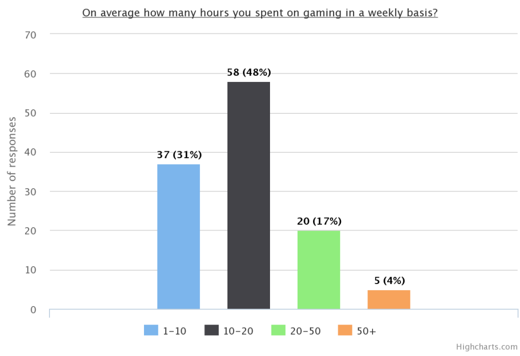


Figure B.1: Questionnaire 1 Answers

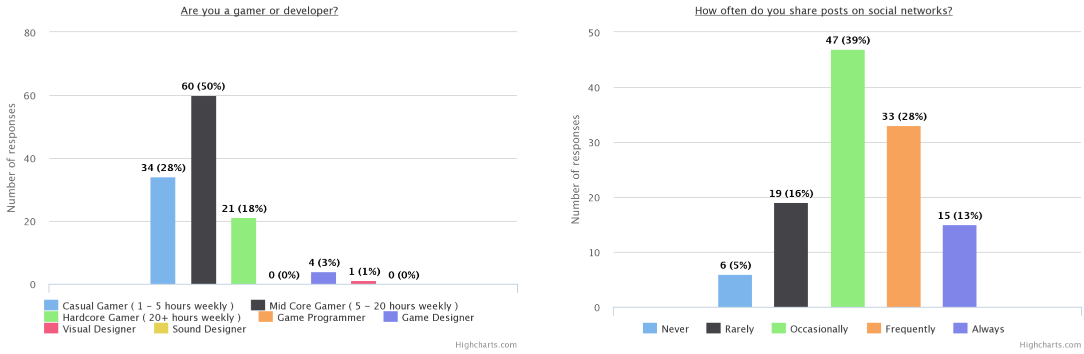


Figure B.2: Questionnaire 1 Answers

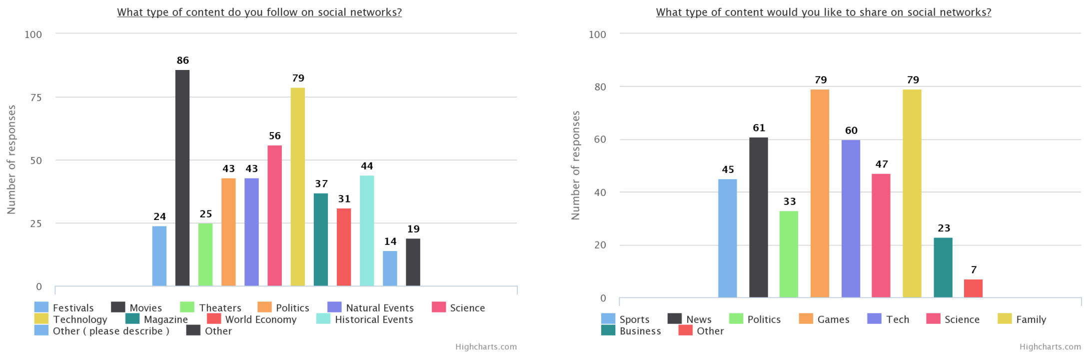


Figure B.3: Questionnaire 1 Answers cont.

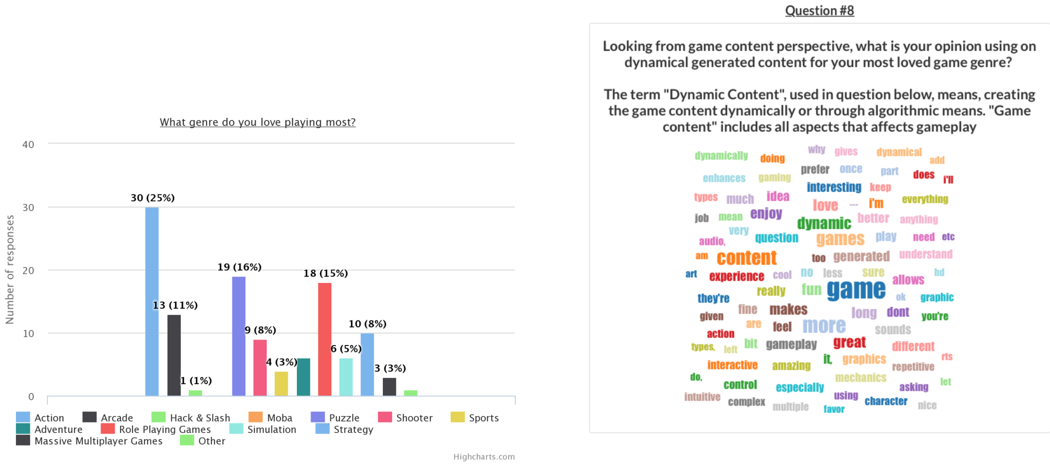


Figure B.4: Questionnaire 1 Answers cont.

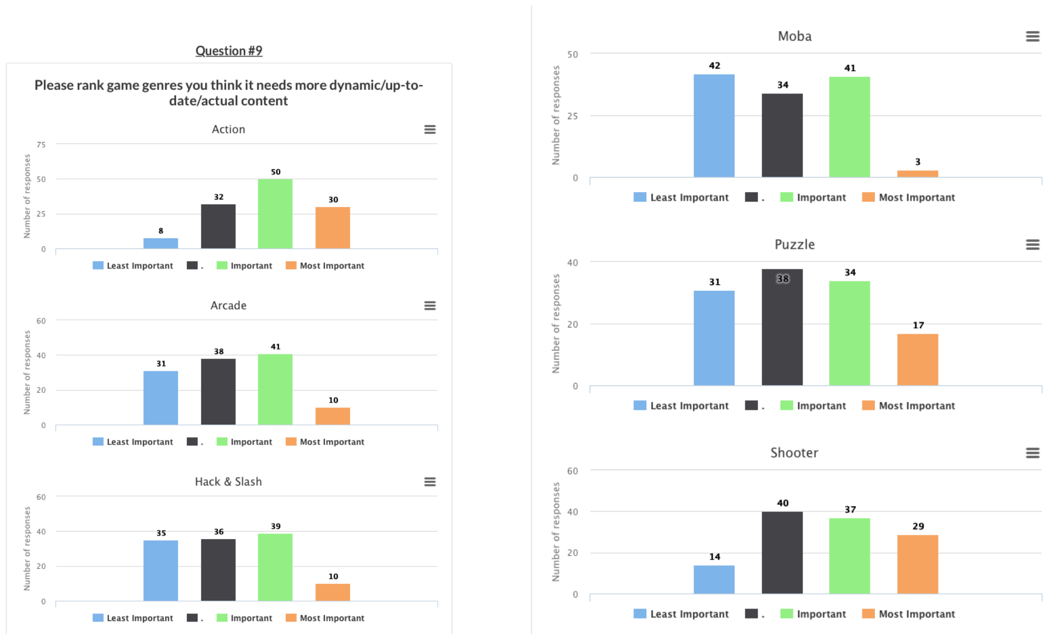


Figure B.5: Questionnaire 1 Answers cont.

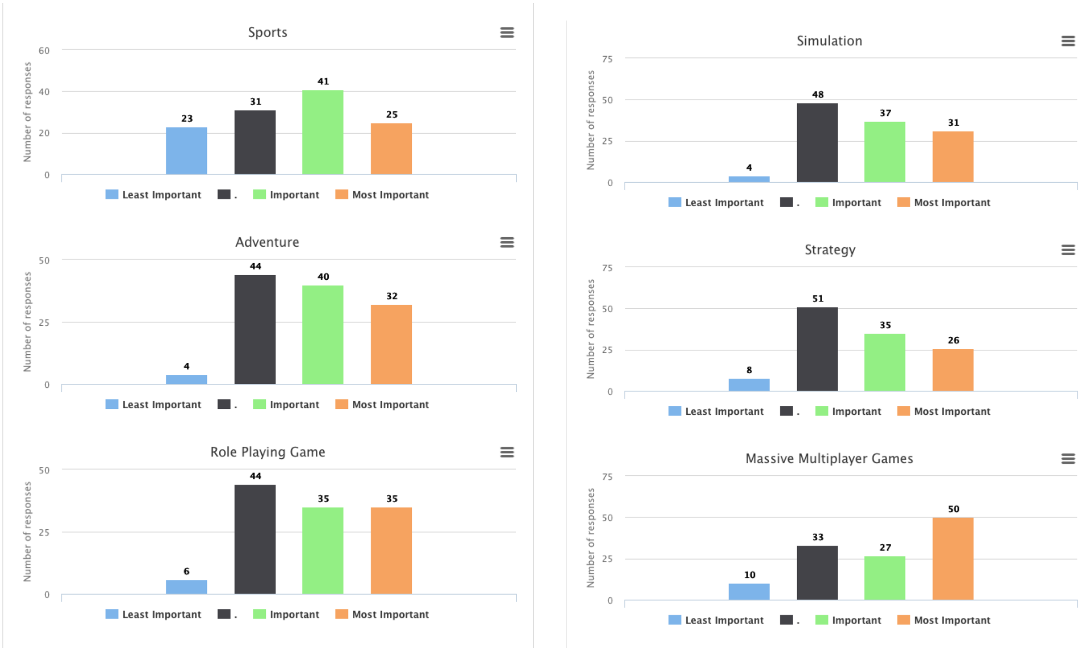


Figure B.6: Questionnaire 1 Answers cont.

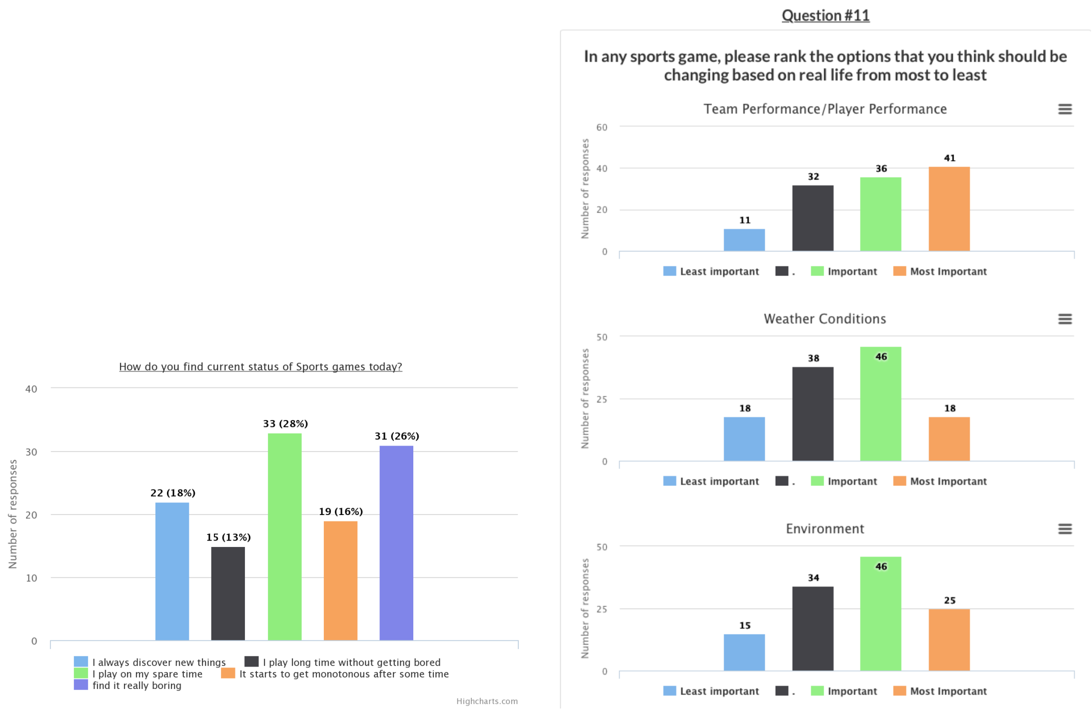


Figure B.7: Questionnaire 1 Answers cont.

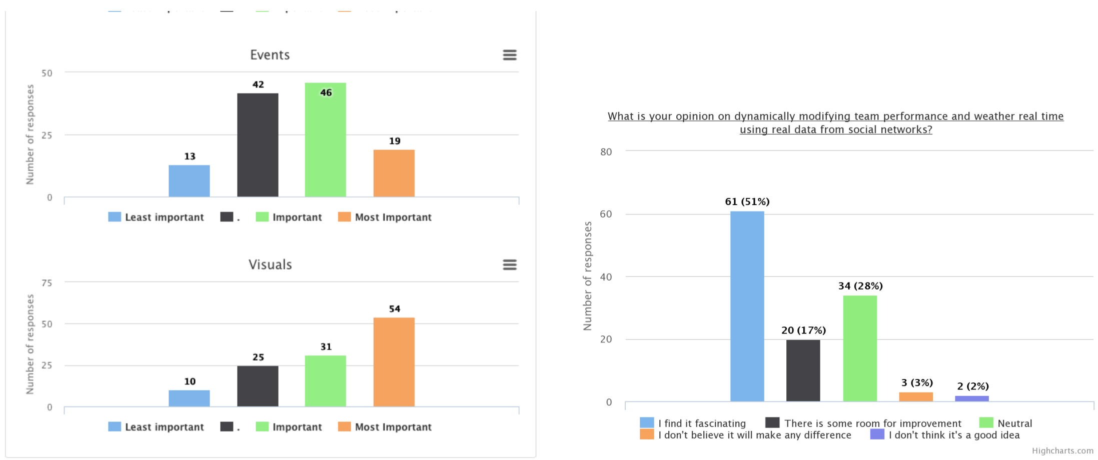


Figure B.8: Questionnaire 1 Answers cont.

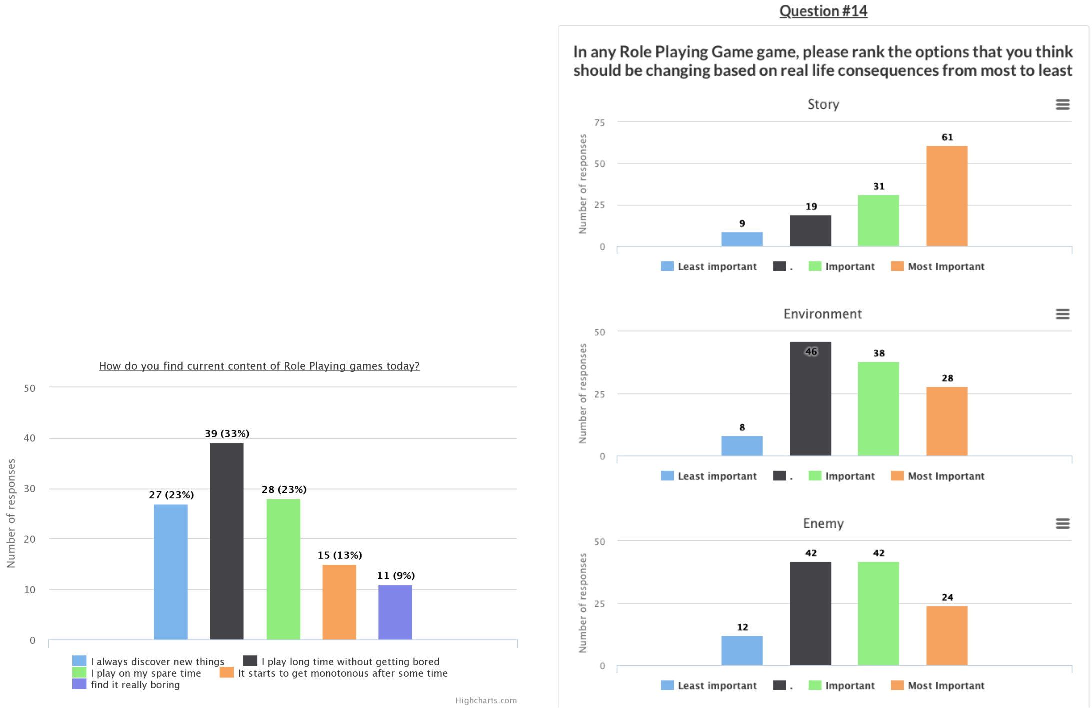


Figure B.9: Questionnaire 1 Answers cont.

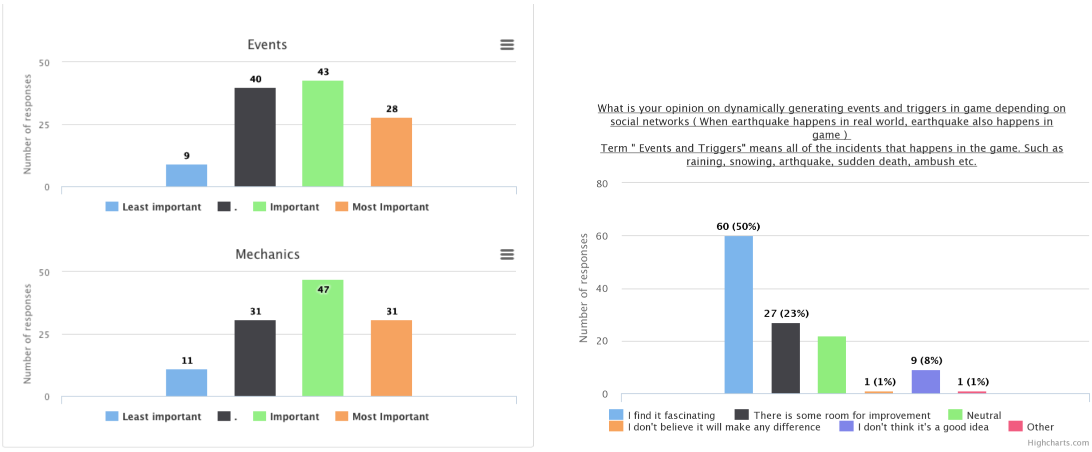
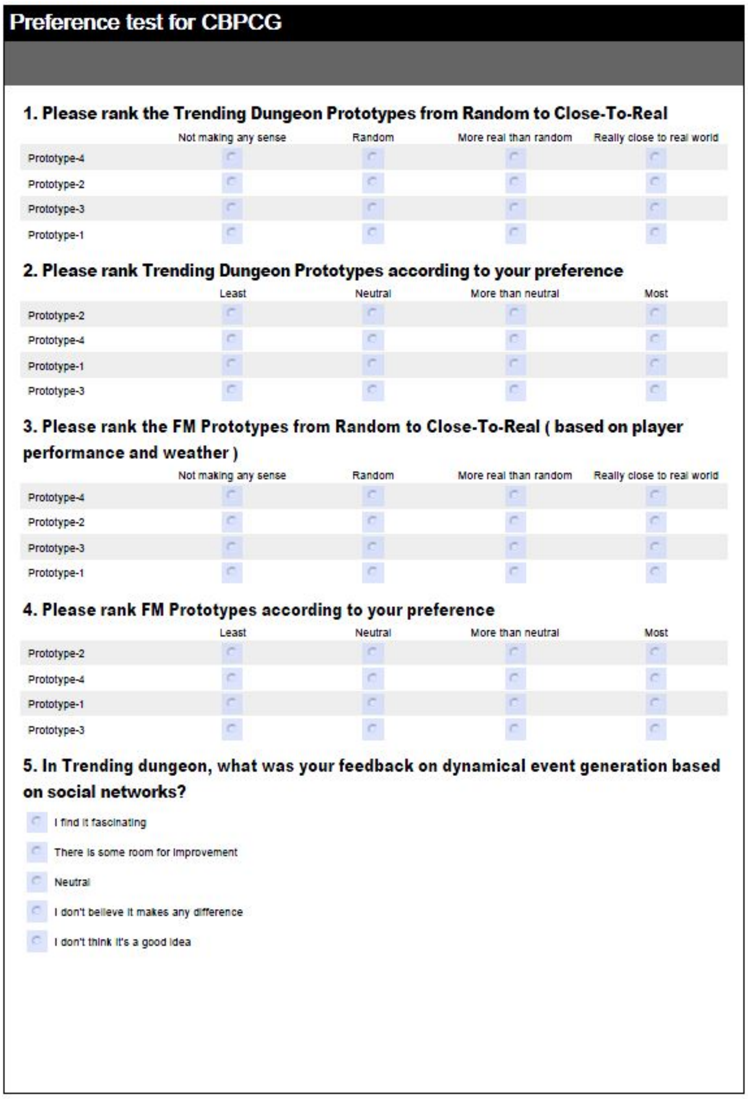


Figure B.10: Questionnaire 1 Answers cont.

APPENDIX C

Appendix C



Preference test for CBPCG

1. Please rank the Trending Dungeon Prototypes from Random to Close-To-Real

	Not making any sense	Random	More real than random	Really close to real world
Prototype-4				
Prototype-2				
Prototype-3				
Prototype-1				

2. Please rank Trending Dungeon Prototypes according to your preference

	Least	Neutral	More than neutral	Most
Prototype-2				
Prototype-4				
Prototype-1				
Prototype-3				

3. Please rank the FM Prototypes from Random to Close-To-Real (based on player performance and weather)

	Not making any sense	Random	More real than random	Really close to real world
Prototype-4				
Prototype-2				
Prototype-3				
Prototype-1				

4. Please rank FM Prototypes according to your preference

	Least	Neutral	More than neutral	Most
Prototype-2				
Prototype-4				
Prototype-1				
Prototype-3				

5. In Trending dungeon, what was your feedback on dynamical event generation based on social networks?

- I find it fascinating
- There is some room for improvement
- Neutral
- I don't believe it makes any difference
- I don't think it's a good idea

Preference test for CBPCG

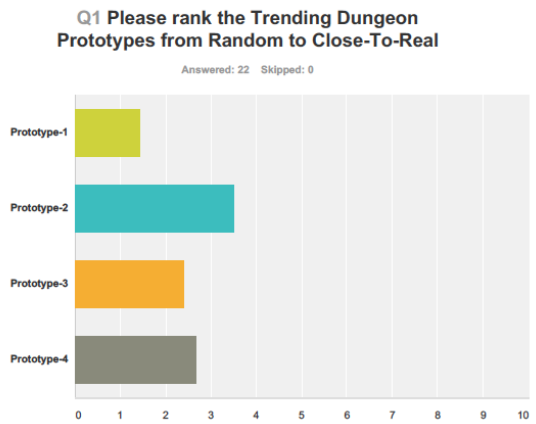
6. In Dynamical Football Manager, what was your feedback on dynamical team & weather modification based on social networks?

- ○ I find it fascinating
- ○ There is some room for improvement
- ○ Neutral
- ○ I don't believe it makes any difference
- ○ I don't think it's a good idea

APPENDIX D

Appendix D

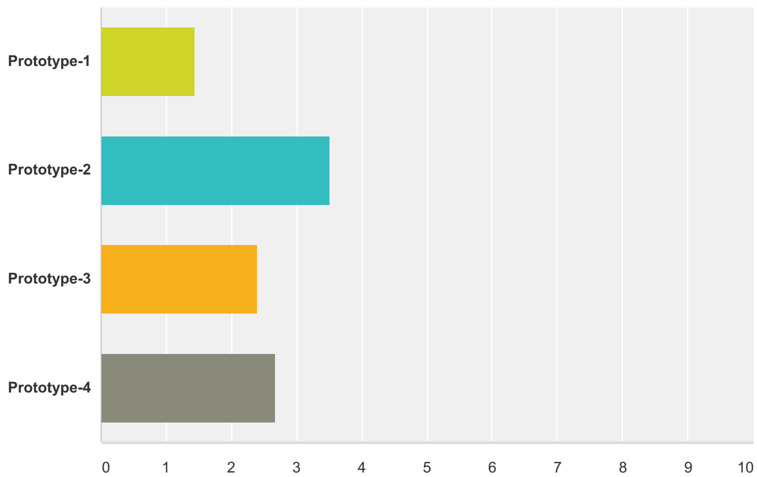
D.1 Questionnaire 2: Answers



	Not making any sense	Random	More real than random	Really close to real world	Total
Prototype-1	54.55% 12	45.45% 10	0.00% 0	0.00% 0	22
Prototype-2	0.00% 0	4.55% 1	40.91% 9	54.55% 12	22
Prototype-3	13.64% 3	40.91% 9	36.36% 8	9.09% 2	22
Prototype-4	9.09% 2	31.82% 7	40.91% 9	18.18% 4	22

Q1 Please rank the Trending Dungeon Prototypes from Random to Close-To-Real

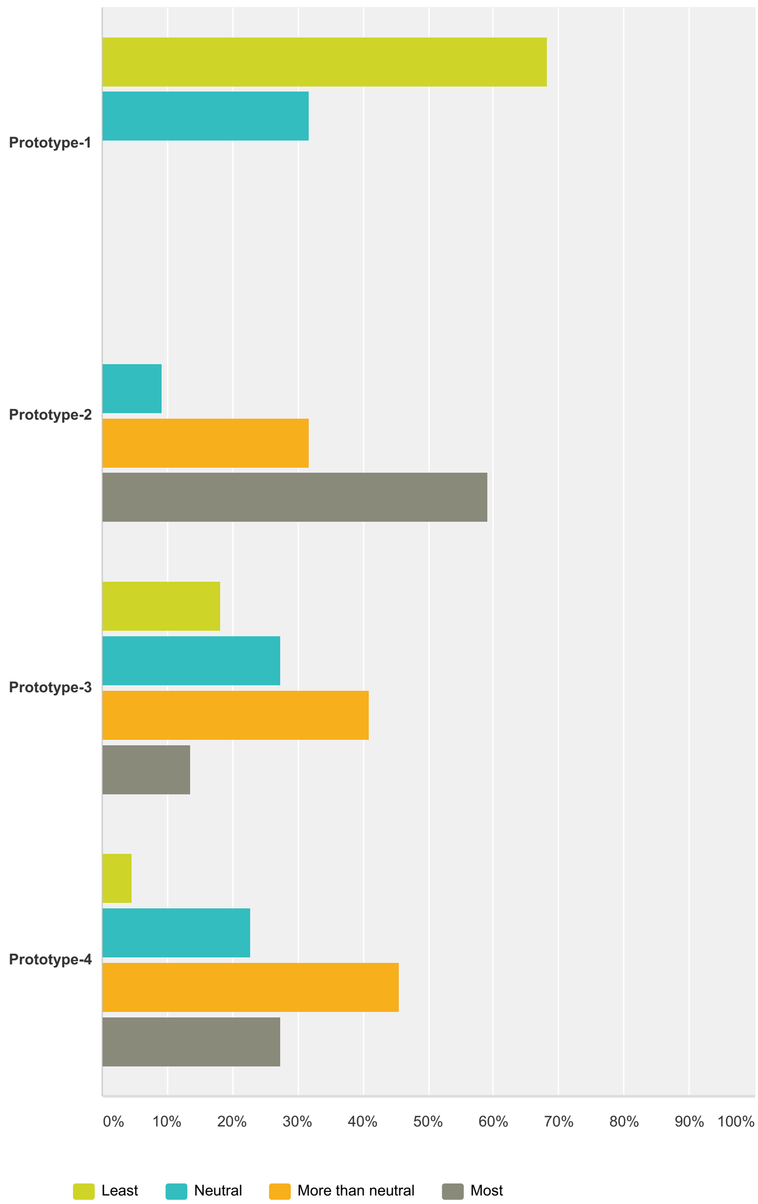
Answered: 22 Skipped: 0



	Not making any sense	Random	More real than random	Really close to real world	Total
Prototype-1	**54.55%** 12	**45.45%** 10	**0.00%** 0	**0.00%** 0	22
Prototype-2	**0.00%** 0	**4.55%** 1	**40.91%** 9	**54.55%** 12	22
Prototype-3	**13.64%** 3	**40.91%** 9	**36.36%** 8	**9.09%** 2	22
Prototype-4	**9.09%** 2	**31.82%** 7	**40.91%** 9	**18.18%** 4	22

Q2 Please rank Trending Dungeon Prototypes according to your preference

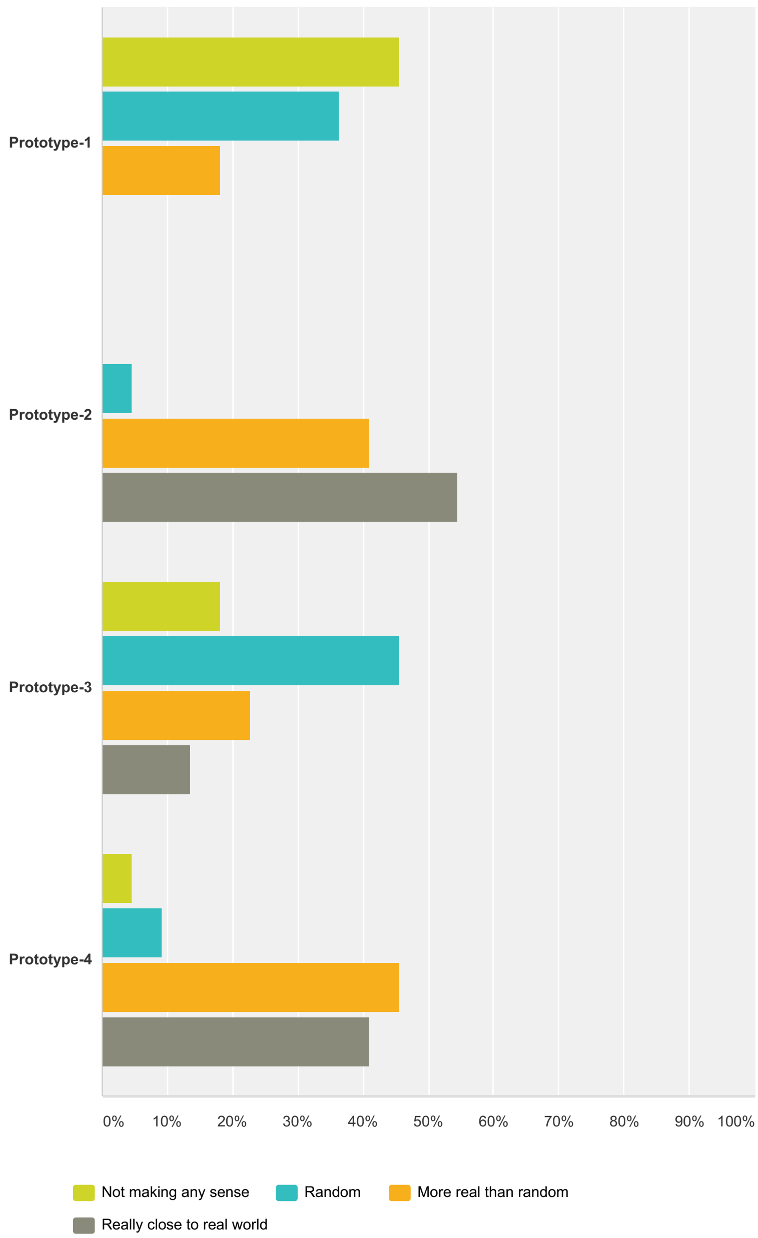
Answered: 22 Skipped: 0



	Least	Neutral	More than neutral	Most	Total
Prototype-1	**68.18%** 15	**31.82%** 7	**0.00%** 0	**0.00%** 0	22
Prototype-2	**0.00%** 0	**9.09%** 2	**31.82%** 7	**59.09%** 13	22
Prototype-3	**18.18%** 4	**27.27%** 6	**40.91%** 9	**13.64%** 3	22
Prototype-4	**4.55%** 1	**22.73%** 5	**45.45%** 10	**27.27%** 6	22

Q3 Please rank the FM Prototypes from Random to Close-To-Real

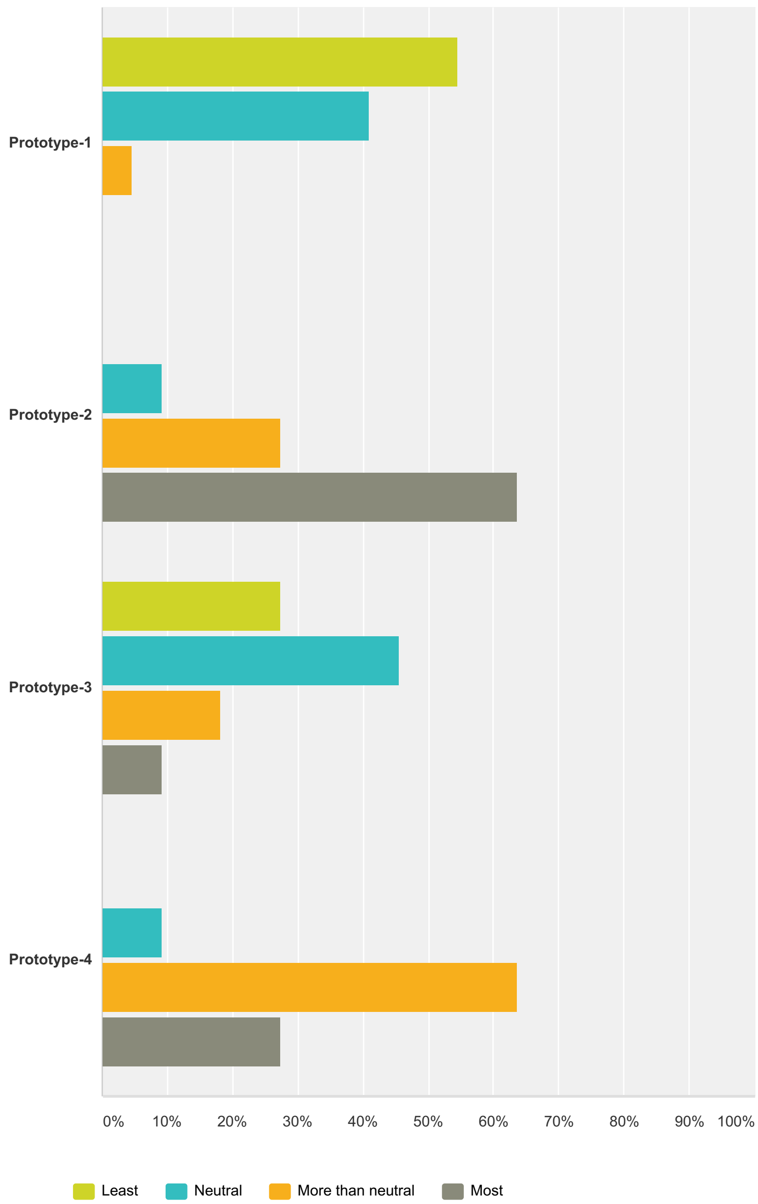
Answered: 22 Skipped: 0



	Not making any sense	Random	More real than random	Really close to real world	Total
Prototype-1	**45.45%** 10	**36.36%** 8	**18.18%** 4	**0.00%** 0	22
Prototype-2	**0.00%** 0	**4.55%** 1	**40.91%** 9	**54.55%** 12	22
Prototype-3	**18.18%** 4	**45.45%** 10	**22.73%** 5	**13.64%** 3	22
Prototype-4	**4.55%** 1	**9.09%** 2	**45.45%** 10	**40.91%** 9	22

Q4 Please rank FM Prototypes according to your preference

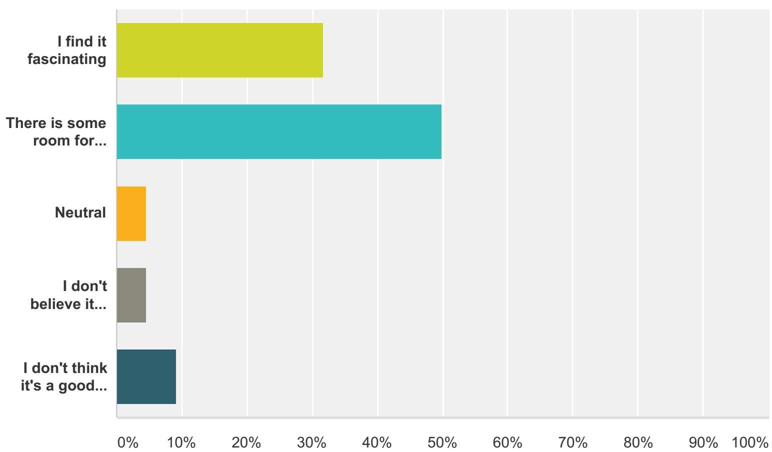
Answered: 22 Skipped: 0



	Least	Neutral	More than neutral	Most	Total
Prototype-1	**54.55%** 12	**40.91%** 9	**4.55%** 1	**0.00%** 0	22
Prototype-2	**0.00%** 0	**9.09%** 2	**27.27%** 6	**63.64%** 14	22
Prototype-3	**27.27%** 6	**45.45%** 10	**18.18%** 4	**9.09%** 2	22
Prototype-4	**0.00%** 0	**9.09%** 2	**63.64%** 14	**27.27%** 6	22

Q5 In Trending dungeon, what was your feedback on dynamical event generation based on social networks?

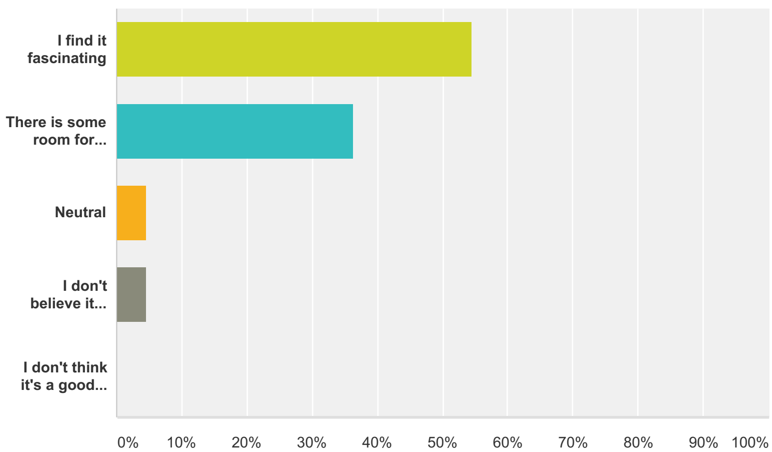
Answered: 22 Skipped: 0



Answer Choices	Responses	
I find it fascinating	31.82%	7
There is some room for improvement	50.00%	11
Neutral	4.55%	1
I don't believe it makes any difference	4.55%	1
I don't think it's a good idea	9.09%	2
Total		22

Q6 In Dynamical Football Manager, what was your feedback on dynamical team & weather modification based on social networks?

Answered: 22 Skipped: 0



Answer Choices	Responses	
I find it fascinating	54.55%	12
There is some room for improvement	36.36%	8
Neutral	4.55%	1
I don't believe it makes any difference	4.55%	1
I don't think it's a good idea	0.00%	0
Total		22

TEZ FOTOKOPİ İZİN FORMU

ENSTİTÜ

Fen Bilimleri Enstitüsü ☐

Sosyal Bilimler Enstitüsü ☐

Uygulamalı Matematik Enstitüsü ☐

Enformatik Enstitüsü ☐

Deniz Bilimleri Enstitüsü ☐

YAZARIN

Soyadı : ....
Adı : ....
Bölümü : ....

TEZİN ADI (İngilizce) : ....
....
....
....
....

TEZİN TÜRÜ : Yüksek Lisans ☐ Doktora ☐

1. Tezimin tamamı dünya çapında erişime açılsın ve kaynak gösterilmek şartıyla tezimin bir kısmı veya tamamının fotokopisi alınsın. ☐

2. Tezimin tamamı yalnızca Orta Doğu Teknik Üniversitesi kullancılarının erişimine açılsın. (Bu seçenekle tezinizin fotokopisi ya da elektronik kopyası Kütüphane aracılığı ile ODTÜ dışına dağıtılmayacaktır.) ☐

3. Tezim bir (1) yıl süreyle erişime kapalı olsun. (Bu seçenekle tezinizin fotokopisi ya da elektronik kopyası Kütüphane aracılığı ile ODTÜ dışına dağıtılmayacaktır.) ☐

Yazarın imzası .... Tarih ....