ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

DESIGNING AN AUGMENTED REALITY BASED CITY BUILDING GAME USING CELLULAR AUTOMATA ALGORITHMS

M.Sc. THESIS

Şeref Atilla GÜRBÜZ

Department of Informatics

Architectural Design Computing Program

JULY 2020

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

DESIGNING AN AUGMENTED REALITY BASED CITY BUILDING GAME USING CELLULAR AUTOMATA

M.Sc. THESIS

Şeref Atilla GÜRBÜZ
523171012

Department of Informatics

Architectural Design Computing Program

Thesis Advisor: Assoc. Prof. Dr. Sema ALAÇAM

JULY 2020

ISTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

HÜCRESEL ÖZDEVİNİM İLE ARTIRILMIŞ GERÇEKLİK TABANLI ŞEHİR GELİŞTİRME OYUNU TASARIMI

YÜKSEK LİSANS TEZİ

Şeref Atilla GÜRBÜZ
523171012

Bilişim Anabilim Dalı

Mimari Tasarımda Bilişim Programı

Tez Danışmanı: Doç. Dr. Sema ALAÇAM

TEMMUZ 2020

Şeref Atilla GÜRBÜZ, a M.Sc student of ITU Graduate School of Science Engineering and Technology student ID 523171012, successfully defended the thesis entitled "DESIGNING AN AUGMENTED REALITY BASED CITY BUILDING GAME USING CELLULAR AUTOMATA", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Assoc. Prof. Dr. Sema Alaçam** İstanbul Technical University	.....
Jury Members :	**Prof. Dr. Gülen Çağdaş** Istanbul Technical University	.....
	Assoc. Prof. Dr. Güven Çatak Bahcesehir University	.....

Date of Submission : 15th June 2020
Date of Defense : 23th July 2020

To my family…

FOREWORD

I would like to express my gratitude to my advisor Assoc. Prof. Dr. Sema Alaçam who has always encouraged me with her positivity and guided me throughout this process despite the long distances caused by pandemic conditions. I also want to thank my family for their endless support and for the trust they hold in me, to my friends for always being there for me and making all this possible.

July 2020

Şeref Atilla GÜRBÜZ
Architect

TABLE OF CONTENTS

Page

ABBREVIATIONS

CA	**:** Cellular Automata
GAs	**:** Genetic Algorithms
LS	**:** L-systems
SI	**:** Swarm Intelligence
SG	**:** Shape Grammars
UI	**:** User Interface
GPU	**:** Graphics Processing Unit
CPU	**:** Central Processing Unit

LIST OF TABLES

Page

LIST OF FIGURES

Page

DESIGNING AN AUGMENTED REALITY BASED CITY BUILDING GAME USING CELLULAR AUTOMATA

SUMMARY

For the next fifty years, it's expected that the number of people living in urban areas will get higher and higher, and cities will expand in more unpredictable and complex ways. This rapid change and unpredictability of cities show that it's very important for both regional and local management to have better understandings of cities' elements and their relationships with each other. To address these issues and analyses, decision-makers have to strengthen their urban intervention reflexes to make the right choice at the right time.

The analysis of the current urban pattern and predictions of development and growth requires a long period, high financial support, and researchers from a wide range of branches. The slow feedback from urban analyses, continuous change of the cities, and fundamental information infrastructure necessitates the adoption of more innovative and rapid methods. In this context, serious games meet all these expectations and fill the gap between researchers, management, architects, designers, and beneficiaries. Towards this aim, previous studies, urban analyses, and different urban scenarios are evaluated in the context of games and serious games.

A large number of constraints and variables of the urban dynamics, difficulty in making accurate predictions, merged structures of the urban systems, and complex sub-systems which constantly interact with each other, direct researchers to apply computational design techniques more and more in their studies. Literature reviews that were made in this sense, shows that cellular automata is one of the most prominent computational design techniques in this issue. The flexibilities and computational capabilities of cellular automata are found suitable to create the foundation of urban growth algorithms in the serious game that is designed for this thesis.

Within the boundaries of the study, games and serious game concepts are mentioned and games with urban design concepts are studied by describing their notable aspects. These games are discussed on methods they use to simulate urban systems, the use in real-world cases, participatory environments they create, and aim for informative intentions. Game elements that form the game are examined in the perspective of the designed game by focusing on the relationship between the game and play notions, rules, and mechanics of the game, gameplay process, player interactions, and representations.

In this direction, the designed serious game is aimed to use cellular automata for in-game mechanics, create a participatory environment for players, and use augmented reality technologies along with board game structure to offer an innovative interaction-workspace that strengthens the urban intervention reflexes of the participants.

HÜCRESEL ÖZDEVİNİM İLE ARTIRILMIŞ GERÇEKLİK TABANLI ŞEHİR GELİŞTİRME OYUNU TASARIMI

ÖZET

Yapılan araştırmalardan yola çıkarak, gelecek 50 yılın perspektifinde oluşturulan senaryolarda, kentlerde yaşayan nufüsun daha da artması ve şehirlerin daha fazla, daha hızlı ve karmaşık şekilde büyümesi öngörüsü oluşmaktadır. Bu hızlı değişim şehirlerin analizinin ve doğru planlamanın ne kadar önemli olduğunu göstermektedir. Yapılan analizlerin yanı sıra kent içindeki gerek genel, gerekse yerel bütün karar mekanizmalarının bu analizleri iyi okuması ve kentin seri tepkilerine karşı yerinde reflekslerle müdahale etmeleri gerekmektedir. Kentsel tasarım süreçlerinde bazı şehirler daha yenilikçi yönetim biçimleri aramakta ve bu tasarım süreçleri kapsamında yapılacak olan çalışmalarda şehrin içindeki farklı tarafları bir araya getirmektedirler. Kent içinde daha küçük yerel yönetimleri katılımcı bir ortam çerçevesinde bir araya getiren ve çok sesli bir ortam oluşturabilen kentsel yönetim biçimleri git gide yaygınlaşmaktadır.

Kentsel gelişimin ve değişimin analizi, bir çok alanda uzmanlık sahibi kişilerin bir araya gelip, yüksek finansal kaynaklar kullanarak, uzun süren çalışmalar sonucu ortaya çıkardıkları uzun ve zorlu bir süreçtir. Bahsi geçen kentsel araştırmaların geri dönüş hızlarının yavaş olması, sürekli değişen ve karmaşıklaşan kentlerin daha hızlı ve erişilebilir yöntemlerin kullanılmasını zorunlu kılmaktadır. Bu noktada Ciddi Oyunlar dinamik yapıları ile bu ihtiyaçları karşılayabilmekte ve şehri yönetenler, mimarlar, tasarımcılar ve hak sahipleri ve mahallesakinleri arasındaki ilişkilerde köprü işlevini kurabilmektedir. Bu tez kapsamında, kentsel değişimin ve kentsel analiz yöntemlerinin ciddi oyun bağlamı altında değerlendirilmesinin potansiyelleri incelenmekte ve bir kentsel değişim senaryosu için ciddi oyunlar birer araç olarak kullanılmaktadır.

Günümüzde hesaplamalı tasarım yöntemleri mimarlık, müzik, sanat, bilişsel çalışmalar gibi bir çok farklı alanda kullanılmakta ve başlı başına tasarım süreçlerine ve araçlarına etki etmektedir. Hesaplamalı tasarım yöntemleri süreçlerin daha hedef odaklı olmasını sağlarken aynı zamanda daha üretken sonuçların ortaya çıkmasını da olanak vermektedir. Karmaşık tasarım problemlerinin çözümüne yönelik olan çalışmalarda, kısıtlar ve değişkenlerin yardımıyla hesaplamalı tasarım yöntemleri süreçlerdeki elle yapılması zor olan çözüm alternatiflerinin oluşturulmasında etkin rol oynamaktadır. Kentsel değişimin analizinde çok fazla kısıt ile değişkenin olması, tahmin edilebilirliğin düşük olması, kenti oluşturan faktörlerin çok katmanlı, dinamik ve birbirleri ile etkileşim halindeki yapısı, kente dair verinin büyük boyutlara ulaşmasına ve kentsel analizlerde bu dverileri işlemek için üretken sistemler gibi hesaplamalı tasarım yöntemlerinin daha çok tercih edilmesine önayak olmaktadır. Literatürde yer alan çalışmalar incelendiğinde, en yaygın teknik olan hücresel özdevinimin yanı sıra şekil gramerleri, genetik algoritmalar, etmen tabanlı sistemler ve sürü zekası gibi farklı üretken sistemlerin de tek tek veya birbirlerini destekleyici

şekilde bir arada kullanıldığı görülmektedir. Bu çalışma kapsamında tasarlanacak ciddi oyun için esneklik ve hesaplamalı tasarım avantajı sebebiyle üretken sistemlerden hücresel özdevinim algoritmalarının oyun içi kentsel değişim ve gelişim kurallarında kullanılması uygun görülmektedir.

Çalışmanın kapsamında öncelikle oyun ve ciddi oyun kavramları açıklanmakta ve daha sonra literatürde yer alan kentsel tasarım konseptindeki oyunlar ele alınmaktadır. Oyunların kentsel gelişme ve simülasyon kavramlarını nasıl yaklaştıkları, katılımcı bir ortam yaratıp yaratmadıkları, gerçek şehir senaryolarında kullanılmaları ve öğretici bir amaç gütmeleri gibi güçlü ve belirleyici noktalarına odaklanılmaktadır. Oyunu oluşturan elemanları; oyun ve oynama kavramlarının birbirleriyle ilişkileri, kuralların tanımlanması, oynanış süreci, oyuncu ilişkileri, tasarlanan oyun çerçevesinde incelenmektedir.

Bu tez kapsamında, kentsel dokuların değişimini hücresel özdevinim kuralları ile oyun mekaniklerine dönüştüren, oyuncuların diyaloğuna imkan veren, arttırılmış gerçeklik teknolojisi ile kutu oyununu bir araya getiren ciddi bir oyun tasarlanmıştır. Çalışmanın özgün olmasının birincil sebebi, hücresel özdevinim algoritmalarını ciddi bir oyun kapsamında kentsel gelişim tahmini yöntemi olarak kullanmış olmasıdır. Çalışmanın özgün katkısı, hücresel özdevinim algoritmalarının oyun kuralları bazında kentsel simülasyonda kullanılması, yarı dijital yarı fiziksel bir etkileşim arayüzü önermesi ve mimarlar, şehir planlamacıları, tasarımcılar ve hak sahipleri için kentsel müdahale reflekslerini güçlendiren katılımcı bir oyun ortamı oluşturması olarak belirtilebilir.

Çalışma kapsamında geliştirilen ciddi oyunun mekanikleri, kuralları, oyunculara sunduğu zorlukları ve amaçları, etkileşim dinamikleri ve temsil kapsamı tartışılmaktadır. Kullanıcılar hücresel özdevinim ile beslenen ve belirli düzeyde rastgele ve tahmin edilemez olan algoritmaya karşı oyunu bir arada strateji geliştirerek oynamaktadırlar. Kullanıcıların oyunu oynadıkları mobil cihaz ile ve oyun piyonları ile girdikleri etkileşimler oyunun oynanış süreci dahilinde tasarlanmaktadır. Oyunun sunduğu zorluk düzeyi oyuna başlarken seçilebildiği gibi kullanıcıların oyun oynadıkça kazanma ve kaybetme istatistiklerine göre güncellenen bir yapısı bulunmaktadır. Bu durum gerek daha fazla zorluk arayan gerekse de kendini daha basit bir şekilde test etmek isteyen oyunculara uygun bir yapı sağlamaktadır. Tasarlanan ciddi oyunda, kendini sürekli daha zor ile sınayan ve yeteneklerinin tecrübe ile gelişebileceğine inanan, gelişim öz-teorisi yapısındaki insanları daha zor oyun biçimleri ile sınayan; yeteneklerinin ve oyun içindeki başarısının tecrübe ve yeniden oynama ile gelişemeyeceğini düşünen varlık öz-teorisi yapısında insanları ise rahat edebilecekleri bir zorluk seviyesi ile karşılaştıran bir yapı mevcuttur. Ciddi oyunların birincil amaçlarından olan, oyuncularına belli bir alanda bilgi verme amacı, farklı yapıdaki oyunculara değişken zorluklar ve farklı oyun içi hedefleri sunması ile sağlanmaktadır. Tasarlanan oyundaki etkileşimler; oyuncuların kendi aralarındaki etkileşimi, oyuncuların fiziksel ögeler ile etkileşimi ve oyuncuların arttırılmış gerçekliği sağlayan cihaz ile etkileşimi olarak üç ana başlıkta incelenmiştir. Oyuncuların kendi aralarındaki etkileşimi ortak bir amaç doğrultusunda hareket etmeleri ve aynı katılımcı ortamda buluşmaları ile sağlanmaktadır. Oyuncuların oyun elemanları ile etkileşimi ise fiziksel ögeler ve dijital arayüzler üzerinden sağlanmakta olup oyuncuların geri bildirimi ile sonuç bölümünde değerlendirilmektedir. Tasarlanan ciddi oyunun elemanları, temsil konusunda, hem şehrin seçilmiş bir bölgesini temsil etmektedir hem de kentsel doku içindeki elemanların bir temsili görevini üstlenmektedir. Coğrafi bilgi sistemi üzerinden alınan veriler işlenerek şehrin dinamik noktalarından biri olan organize sanayi bölgesi oynanabilir bir senaryo olarak

sunulmaktadır. Var olan senaryonun yanı sıra, uygulama tarafından rastgele oluşturulmuş senaryolar üzerinden de oyun oynanabilmektedir. Oyunun fiziksel elemanları modellenerek hem mobil cihaz tarafından algılanabilecek hem de 3 boyutlu yazıcı ile çıktı alınabilecek şekilde tasarlanmaktadır.

Çalışmanın son bölümlerinde uygulamanın daha detaylı bir incelemesi yapılmaktadır. Bu inceleme kapsamında geliştirilmekte olan oyun içerisindeki fonksiyonlar, kurallar, olaylar ve değişkenlerin tahlili, bu fonksiyonların birbirleri ile olan ilişkileri, oyunun nasıl üretildiği, oyun içerisindeki aktörlerin arttırılmış gerçeklik teknolojisini kullanarak fiziksel oyun elemanlarının üzerlerine nasıl yansıtıldıkları ve hücresel özdevinim algoritmasının oyun yapay zekası tarafından kural tabanlı bir tahmin edilemezlik yaratıp oyun mekaniğini şekillendirdiği belirtilmiştir.Ek olarak, oyunun geliştirilmesi işinin de yapıldığı, uygulama iskeletini oluşturan Unreal Engine 4 oyun motoru ve arttırılmış gerçeklik uygulaması geliştirme donanımlarından GoogleAR hakkında bilgi verilerek arka planda yapılan çalışmaların kapsamları ve nitelikleri aydınlatılmaktadır.

Sonuç bölümünde, tasarlanan oyunun değerlendirilmesi oynanış testleri üzerinden yapılmaktadır. Yapılan oynanış testleri sonucunda oynanabilirlik, kullanılabilirlik, katılımcı ortam oluşturulması ve yeniden oynanabilirlik gibi kavramlar üzerinden değerlendirilmiştir. Oyunun fiziksel ve dijital ögeleri tasarlanıp prototipleri oluşturulmuş ve oyun iki test grubuna daha önceden belirtilmiş kurallar ile birlikte oynatılmıştır. Oyun süreci sonrasında her bir oyuncunun izlenimleri ve deneyimleri anket üzerinden elde edilmiştir. Değerlendirmenin nitel bölümü oynanış sürecinin kayıt altına alınıp, daha sonra analiz edilip incelenmesi sonucunda, nicel kısmı ise kullanıcılar tarafından doldurulan anket ve yapılan görüşmeler sonrasında elde edilmiştir. Geliştirilen uygulama, nihai sürüm öncesi bir deneme sürümü olarak ele alınmış olup değerlendirmeler bu ölçüte göre yapılmıştır.

1. INTRODUCTION

Cities, as we define, are complex systems with lots of elements and variables that they have. The constant change of these elements and subsystems, interactions of these elements with themselves, and also with each other makes it a very hard process to research them. For the last decade, the increasing and almost immeasurable amount of data we have and the technology helping us to use that data by the use of immense computational power in our favor, researchers focus on this topic more often and it helps us to have more tangible results for analyzing and predicting the urban dynamics of our cities. These results, however, require expert knowledge in various topics and take lots of time and effort to make and understand. While the cities are growing at a higher pace than ever, and some cities are adopting collaborative governance, this delay and the need of sharing expert knowledge can be crucial. In situations like this, games can take an important role in acting as a bridge between experts and non-experts. By their nature, games are great for negotiating, creating strategies, or using problem-solving skills. In the last decade, games became a more popular tool for decision-making and serious games are becoming more popular every day. Even though serious games are not suitable as tools for predicting and calculating urban dynamics, they help the players to give better reactions to the feedback in the future. Also, bringing the stakeholders together under an entertaining roof helps the share of the knowledge and learning process. Therefore, a serious game is designed for creating a participatory environment, which also uses augmented reality technologies along with cellular automata algorithms to offer an innovative workspace.

1.1 Games and Serious Games

The history of games dates back to ancient times and they are one of the most common social interaction methods between humans. The definition of the games can be simplified as a willing act of creating and overcoming unnecessary obstacles (Suits & Hurka, 2005). Throughout history, humans mostly played games for pure entertainment but there are some examples where they had different focuses in the

foreground. Some of the pre-modern games had religious or moral intentions for the players such as Moksha Patam, in modern world Chutes and Ladders, from Ancient India is a game to teach virtue to children and the gameplay is based on pure luck (Arneson, 2019). The use of games for different purposes than entertainment is nothing new but the number of cases where games are used this way is rising and it is becoming common to use games in different subjects these days.

Serious Games are games that have a more prior objective in areas such as education, healthcare, advertisement, defense other than just entertainment and they use the nature of the games as a tool to fulfill an objective while entertaining the players. More specifically, serious games enhance problem-solving, learning, and deciding processes. Serious games are often used as a reinforcement to the learning process. Games and serious games are nothing new to architectural design issues. There are lots of games that simulate real-life design problems such as the difficulty of managing a city. Architects and designers can learn much from these games while entertaining themselves. Additionally, games are more unlikely to have lots of design constraints comparing traditional architectural design techniques. Game environments with urban context may not even consider the static strength of the buildings they have or the socio-cultural backgrounds of the city. This allows designers to have more freedom and try different perspectives than they're used to in their fields as well as games inspiring them and stimulating their creative mind.

1.2 Computational Approaches in Urban Growth Analysis

Cities are dynamic and complex systems with lots of other subsystems. They are hard to analyze and it takes lots of time and effort in various areas to do so. While understanding urban systems and urban growth can be quite challenging, it is also very important to take quick and on the spot decisions that are based on accurate analyses. Nowadays, most of the urban areas are growing and sprawling rapidly and researchers have to catch up with this speed. The number of humans in the world is increasing and starting in 2007 more people are living in urban areas than rural. More than 54% of the population lives in the cities and in 2030 this number is expected to be around 60% where at least 2 out of 3 people will be living in urban areas (Ritchie & Roser, 2018). Thereupon, countries are trying to overcome the problems that this rapid growth brings. To deliver sustainable land management and urban planning, countries should

reduce the complexity in understanding, analyzing, and managing urban systems and their sub-systems. One of the key methods to reduce the complexity of spatial and temporal urban growth to a minimum is trying to guess it. For more than 20 years, researchers have created models that provide useful information and a better understanding of our cities.

Urban Growth Prediction Models (UGPM) are processes of analyzing numerous complex relationships of an urban area in time and space axes as well as decision-making. Increasing usage of Geographic Information Systems (GIS) provides a great framework for analyzing, managing, and gathering data and led to great advancements in monitoring urban features but it is not enough by itself. Along with GIS, techniques and theories should be accordingly developed to use it as a supporting element of the decision-making process. (Jianquan, 2003). Understanding the sub-systems of the urban structure and its variations requires a multi-disciplinary study. Urban growth prediction models use a wide variety of algorithms to reflect numerous relations and interactions between socioeconomic subset, significant natural environment impacts, and the built environment. The algorithms vary from cellular automata, agent-based modeling, artificial neural networks, fractal modeling to many more. In some of the research, there is more than one algorithm that has been implemented to a single model for better accuracy.

1.3 Games with Urban Design Concept

One of the formal elements of the games is players. In every game, the players are voluntary and engaging participants in the gameplay process (Nacke, 2014). The players should have an interest in playing the game and be in the right mental state. The players should be in curious states to reach the end of the game but also accept the rules as they are while participating. The mindset of players to start engaging in a game is called Lusory Attitude (Suits & Hurka, 2005). The players in that attitude try to achieve the goals by following the determined rules. The rules should be introduced to the players clearly to encourage attraction to the game so very complex rules and algorithms such as urban analysis or urban growth systems should be simplified and abstracted.

Games are one of the best ways to make a process more entertaining. At the same time, games can teach players to think in more innovative ways. The players can be

diversified into two groups in terms of gaming mindsets. Some players have fixed mindsets where they believe that their skillsets can't be improved therefore, they tend to seek challenges that they can surpass with no problem and avoid harder challenges that can make them fail. The rest of the players have growth mindsets where they think training and consistency let them overcome harder and harder goals so they push themselves more for improving their skills (Lee et al., 2012). Adjustable difficulty and goals can satisfy players with different gaming mindsets and maximize what they get from a serious game.

It's no surprise that serious and epistemic games are often used in architectural design and urban planning. Whether these games aim for pure entertainment or they are serious games that focus mostly on design and education aspects, they let the players get feedback from their actions and strengthen their decision-making process. While playing games, the players can start over and reconsider the decisions they made for improving the direction it's going or just to try different perspectives. The players try to accomplish certain objectives while playing and a well-balanced game gives enough satisfaction for the players to devote themselves to give serious attention and overcome in-game problems.

Cities: Skylines is a good example of a detailed urban simulation that the players have the responsibility of managing a city. Cities: Skylines is a city-building and management game that digs in so many aspects of a city like traffic-flow, zoning, managing, public services, taxation, and more. In this game, the players are faced with lots of thrill and trouble while managing the city and the game even includes random natural disasters, detailed public services including but not limited to education, health-care, waste management, noise regulations, etc (Figure 1.1). However, the socio-economic aspects of the game seem to be weak as residents have no social simulation, historical background, political thoughts, or urban memory. The success of Cities: Skylines in urban management and design with its highly detailed sub-systems led it to be used as a tool in real-life urban management. Besides the millions of players that have played the game, the publisher of the game, Paradox Interactive, worked with architects, urban planners, and public officials in a professional intersection. The game mechanics were used in designing the transportation line to test how infrastructure could change traffic and energy systems in Stockholm, Sweden in 2016 (Nutt, 2016).

Figure 1.1: Cities: Skylines as a tool for Urban Development in Stockholm, Sweden [Url-1, 2016].

Even though games can benefit from complex algorithms that a computer can do on the background in real-time, not all games are digital. Inspired by a medieval castle, Carcassonne is an award-winning tile-based multiplayer board game that was first created in 2000 by Klaus-Jürgen Wrede (Figure 1.2). In Carcassonne, the players create a medieval-themed landscape by placing their terrain cards in the game area when their turn comes. The 2000 version of the game contains 72 terrain cards, 40 wood pieces called followers in-game, and a score-table. Carcassonne is a great example of competitive strategic thinking games and the players compete with each other by creating the landscape. There are also multiple renewed board game versions and digital versions of the game after reaching great success. Every year, a tournament in Germany takes place where lots of gamers around the world come together and play Carcassonne to be the best player in the world.

Figure 1.2: Modular board game elements of Carcassonne [Url-2, 2016].

Regarding the freedom and creativity in video games, Minecraft is one of the first games that come into mind being on the top in the best-selling games of all time with more than 180 million copies sold (Hart, 2019). Minecraft is a sandbox type, 3D, first-person perspective game where the players have no goal but to create and design (Figure 1.3). The game is made up of blocks and the world is randomly generated each time it starts. The blocks have a wide range of variety and the players can shape up the surrounding environment by mining, digging, and building. The players can interact with each other in Minecraft by using local area networks or servers on the internet. Throughout the years, Minecraft has been used in other fields such as education and experimental research (Nebel et al., 2016).

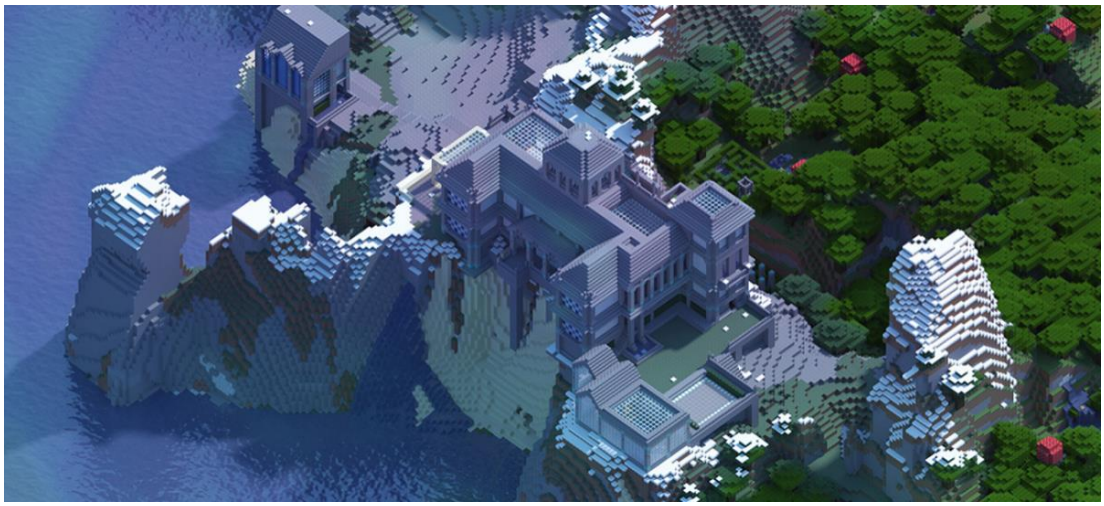


Figure 1.3: An exhibition area for digital artists created in Minecraft Realm [Url-3, 2020].

Minecraft is also used in some real-life focused projects (Figure 1.4). Cities at Play is a project that brings young people living in the deprived areas of Copenhagen to create together and define the problems and advantages of their neighborhoods (Magnussen & Elming, 2017). Cities at Play also aims to create a starting point for the urban planning department (Magnussen & Elming, 2017).

Another example of projects that use Minecraft as a design platform is Block by Block. Block by Block is an international non-profit organization supported by the UN-Habitat, Microsoft, and Mojang that aims to use Minecraft as a tool in participatory urban design activities founded in 2012 (McDaniel, 2018). The participants to Block by Block workshops work together to create public spaces. The process starts with modeling the site in Minecraft. After the model is finished, a workshop is organized with a wide range of representations from the community including women, elderly, youth, and people with disabilities. Then, the participants are introduced to the area by visiting, observing, and being trained by experts. Following this process, the participants divide into groups to discuss and create new ideas and share them with others. The last step is building the design that is created as a common outcome from the participants.

Figure 1.4: Recreation of a Sports Field in Minecraft (Westerberg & von Heland, 2015)

Other projects that are developed using Minecraft are GeoCraft NL (Scholten et. al., 2017), EcoCraft, Vibcraft, Denmark in Minecraft, and Britain in Minecraft.

Play the City is an international organization that creates participatory urban design games in areas such as affordable housing, migration, urban transformation, sustainable tourism, and participatory design. It aims to bring stakeholders from various backgrounds and let them play together in a creative and design-oriented process. Play the City, creates a common ground with different parts of the society including local managements, residents, beneficiaries, and more. It remains an important example of how games are used as a tool in participatory design and decision-making processes. Play the City contains many sub-projects in different cities with different scenarios (Figure 1.5). Scenarios were described according to the city in which it's been played in. Before the gameplay sessions, participants meet and define the main goal of the play further by exchanging ideas.

Figure 1.5: Play Noord, a subproject of Play the City, creates a participatory environment [Url-4, 2016].

Play Noord is a subproject of Play the City that takes place in Amsterdam Noord district. It aims to create a high-density mixed-use area by addressing the disuse of research laboratories in the neighborhood. Play Noord, brings people from various backgrounds together such as environmental activists, stakeholders, entrepreneurs, investors, and more. The attention it gathered provided an opportunity to create a digital gathering area with social media. Later, this social media participation led formations of polls and a platform for sharing and discussing ideas.

2. GENERATIVE DESIGN TECHNIQUES IN URBAN GROWTH MODELS

Generative design techniques are usually adopted by architects where conventional design techniques are not capable enough to explore huge numbers of variables and constraints. While analyzing a city pattern and its growth, there are endless factors that researchers should consider. Therefore, researchers approach the problem with generative design techniques (Table 2.1).

Table 2.1 : Previous examples of urban growth models using generative design techniques.

Author	Objective	Scale	Method
Al-Ahmadi et al. (2009)	Predictive	Urban	GA, CA
Al-Khedar et al. (2007)	Descriptive, predictive	Urban	GA, CA
Arai & Akiyama (2004)	Predictive	Urban	CA
Barredo et al. (2003)	Predictive	Urban	CA
Batty et al. (1999)	Predictive	Urban	CA
Beirão & Duarte (2005)	Prescriptive	Urban	SG
Blecic 1et al. (2013)	Predictive	Urban	CA
Brown & Johnson (1985)	Descriptive, predictive	Architectural	SG
Cao et al. (2012)	Prescriptive	Urban	GA
Clarke & Gaydos (1998)	Predictive	Urban	CA
Chan & Chiu (2000)	Predictive	Urban	LS
Deadman et al. (1993)	Descriptive	Urban	CA
Duarte & Rocha (2001)	Descriptive	Architectural	SG
Duarte. et al. (2006)	Descriptive	Architectural	SG
Feng & Lin (1999)	Prescriptive	Urban	GA
Feng et al. (2011)	Predictive	Urban	SI, CA
Foroutan & Reza (2012)	Descriptive, predictive	Urban	GA, CA

Author	Objective	Scale	Method
Jan et al. (2008).	Prescriptive	Urban	SG
Jenerette & Wu (2001)	Predictive	Urban	GA, CA
Khalilna et al. (2013)	Predictive	Urban	SI, CA
Kocabas & Dragicevic (2006)	Descriptive	Urban	CA
Lau & Kam (2005)	Predictive	Urban	CA
Liu et al. (2015)	Prescriptive	Urban	GA
Ménard & Marceau (2005)	Descriptive	Urban	CA
Naghibi et al. (2016)	Descriptive	Urban	SI, CA
Shafizadeh Moghadam & Helbich (2013)	Predictive	Urban	CA
Stewart et al. (2004)	Prescriptive	Urban	GA
Stewart & Janssen (2014)	Prescriptive	Urban	GA
Tan et al. (2015)	Descriptive, predictive	Urban	CA

There are lots of computational design techniques researchers use to analyze urban growth. One of the most common techniques used in urban growth prediction is cellular automata (CA). On the basis, cellular automata use clusters of aligned cells on grids where it evaluates the states of cells and changes their states in the subsequent generation to create new design possibilities in every iteration according to the current states of surrounding cells (Singh & Gu, 2012). Cellular automata use grids of cells and each cell is defined by a state. The layout of the cells can be three dimensional, two dimensional, or on a single axis. On two-dimensional cellular automata, cells are used to generate complex patterns or to replicate behaviors (Packard & Wolfram, 1985).

The basics of two-dimensional cellular automata system can be listed as:

- Every system consists of numerous cells that rest on a grid.

- Every cell has a state. The number of states affects the complexity of the CA systems. While there is no limit for the maximum number of states, the more it's used the more computational power is required for each iteration. The most basic CA has two states of 0 and 1.
- Each cell has neighbor cells. Neighbors of the cell can also be classified differently (Figure 2.1). Some models only classify adjacent cells as neighbors and called 5-neighbor square, where others make the calculations for both adjacent and diagonal cells called 9-neighbor square or von Neumann and Moore neighbors. On some CA systems neighbors can also have triangular or hexagonal relations.

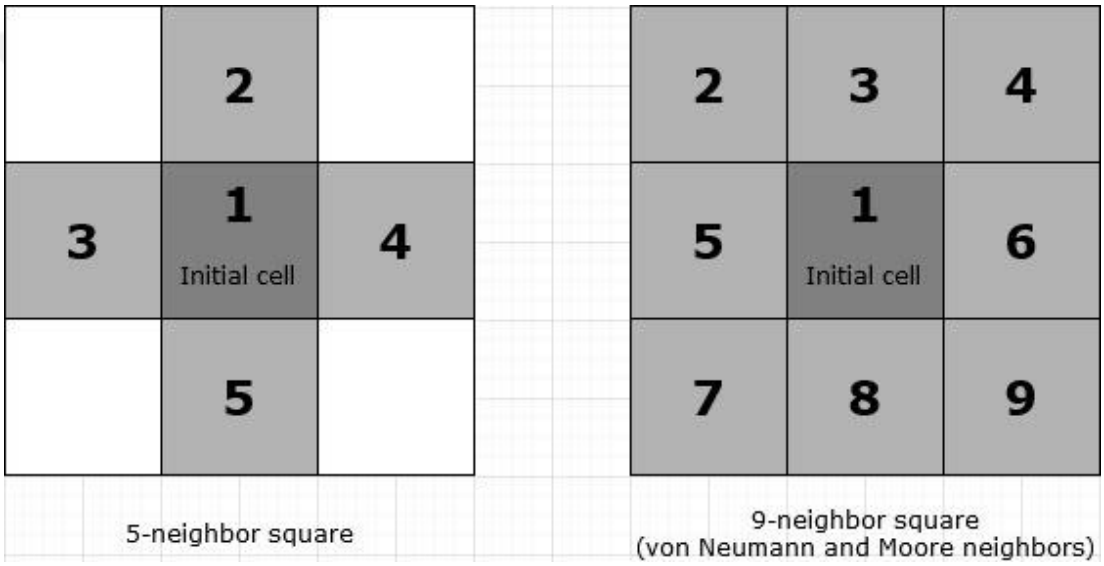


Figure 2.1: 5-neighbor and 9-neighbor squares around the initial cell.

Introduced by John Horton Conway in 1970, Conway's Game of Life is one of the most common examples of the cellular automaton. In the Game of Life, Conway introduces his own set of rules to imitate the basic mechanics of life with only two states; alive and dead. Therefore, there are three main mechanics called Birth, Death, and Stasis (Figure 2.2). Unlike previous examples, Conway's rules and states are simpler and the game takes place on an assumed infinite grid (Gardner, 1970):

- Death: Every alive cell that has 4 or more alive neighbors will be a dead cell in the subsequent generation due to overpopulation. Every alive cell that has 1 or less alive neighbor will be a dead cell in the subsequent generation due to loneliness.

- Birth: Every cell that has exactly 3 alive neighbors will be alive in the subsequent generation.
- Stasis condition: Cells remain in the same states in two ways, remaining dead and remaining alive. Every dead cell that has a number of living neighbors other than 3, will remain dead. Every alive cell that has 2 or 3 alive neighbors, will remain alive.

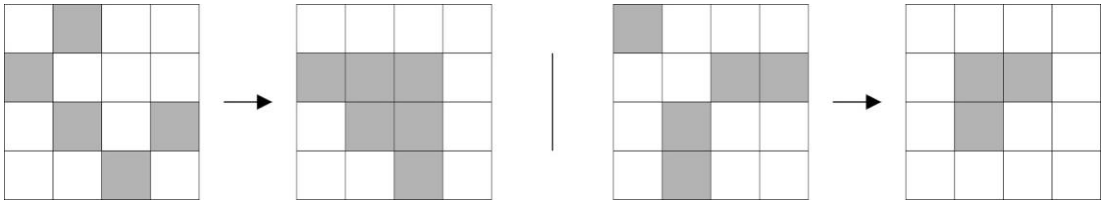


Figure 2.2: Cellular automata rule applying to an initial shape and alternating.

Conway's Game of Life remains on an infinite grid without any limits and both deaths and births happen at the same time on each iteration (Gardner, 1970). After the initial pattern of cells is defined, the Game of Life plays itself so it can be classified under zero-player games. Some patterns of the Game of Life results in oscillating shapes that change back and forth. Patterns that return to their initial shape in every n^{th} generation are oscillating (Figure 2.3).

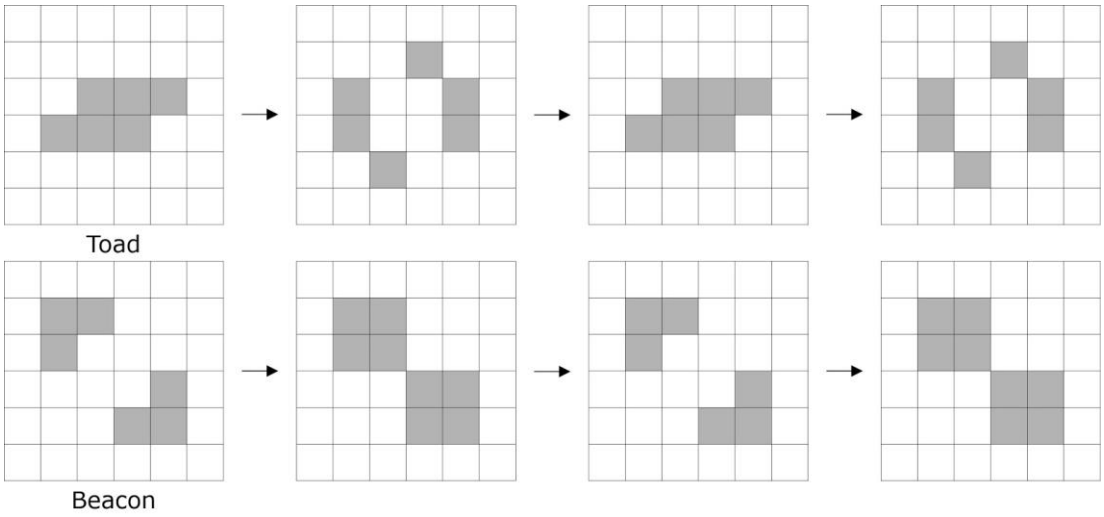


Figure 2.3: Examples of some oscillating initial patterns.

Some of the patterns stay as they are and become static with no change (Figure 2.4). These patterns can only change again if there is any intervention in the system. There are also migrating patterns that create the perception of movement around the grid to

infinity. These patterns tend to go further away from their starting point and oscillates between shape in every n^{th} generation (Figure 2.4).

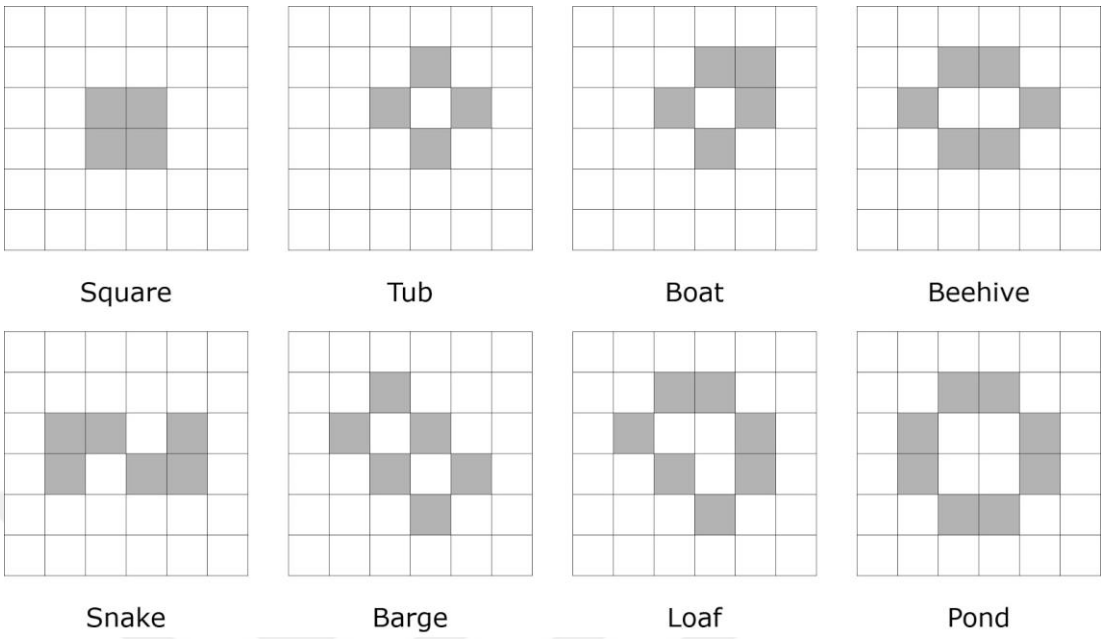


Figure 2.4: Example of static initial patterns that do not change from generation to generation.

The idea of creating an urban system using CA dates back to the 1960s. First studies were including a grid-based residential development model and tests were made for residential growth in metropolitan areas (Chapin & Weiss, 1968). These studies were using cell-based algorithms but they were not exactly using cellular automata for predicting urban growth. Cell-based algorithms were considered useful for spatial analysis but their 'background conventions' were preventing them to be useful in real-world geographic cases (Couclelis, 1985). In the following decades, the use of strict CA with geographical systems started to take place through regional simulations (Tobler, 1970) and a similar model to Conway's "Game of Life" with geographical consideration started to be used in these simulations (Couclelis, 1985).

Some of the first applications of urban CA to the simulation of real-world cases were carried out by Batty and Xie (1994) in Amherst, New York followed by the research to predict urban growth in San Francisco and Washington / Baltimore (Clarke & Gaydos, 1998). Starting from the 2000s, the number of studies that are using cellular automata in urban modeling increased rapidly. (Wahyudi & Liu, 2016). Along with the increase in the number of studies, additional computer power also let researchers add more factors, states, and rules to their algorithm that helped them to make more

detailed analyses (Wahyudi & Liu, 2016). Following these advancements, researchers achieved accurate predictions and descriptions about cities (Figure 2.5). These places including but not limited to Dublin, Ireland (Barredo et al., 2003); Lagos, Nigeria (Barredo, et al., 2004); Tokyo, Japan (Arai & Akiyama, 2004); Quebec, Canada (Ménard & Marceau, 2005); Riyadh, Saudi Arabia (Al-Ahmadi et al., 2009); Mumbai, India (Shafizadeh Moghadam & Helbich, 2013) and Wuhan, China (Tan et al., 2015).

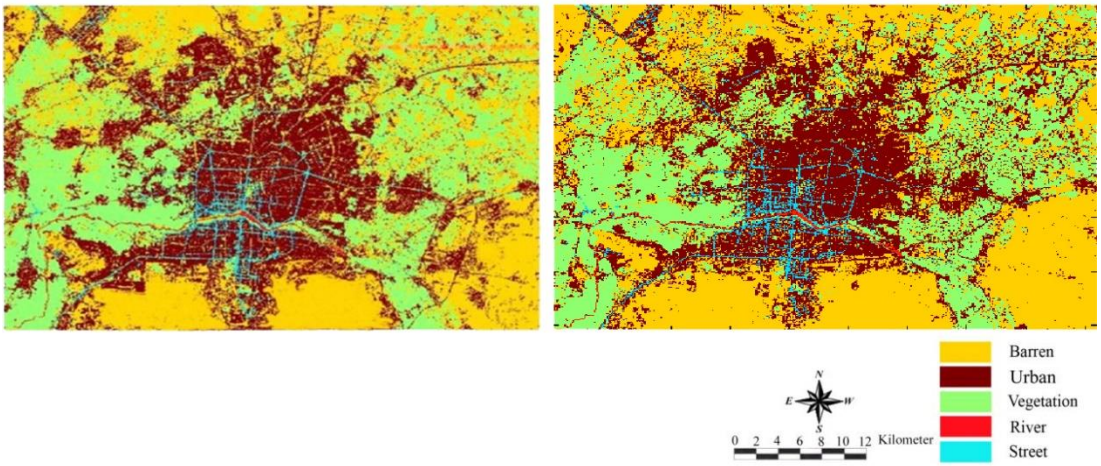


Figure 2.5: Analysis of Isfahan metropolitan area, Iran, according to land-use *(left)* and simulated land use map with 3x3 cellular automata calibrated with genetic algorithms *(right)* (Foroutan & Reza, 2012, p. 6).

Aside from cellular automata, another generative design technique, genetic algorithms, is used frequently in this subject. Genetic algorithms were first introduced by Holland (1975) and it's used for solving both constraint and non-constraint problems that are using a system, based on natural selection and evolutionary process. Genetic algorithms provide solutions which are modified from generation to generation. Genetic Algorithms are theoretically and empirically proven to provide robust search in complex spaces (Goldberg, 1989) and suitable for providing predictions or solutions to complex systems such as urban growth. Genetic Algorithms are based on the theories of Charles Darwin, so they benefit from biological behaviors and it uses genetic operators such as crossover, mutation, or natural selection. Genetic algorithm process can be distributed into five different phases:

- Initialization: Every algorithm has a set of individuals called phenotypes. Phenotypes are candidate solutions to the problem with various variables known as Genes. Genes come together to create a string of solutions to the problem and form Chromosomes (Figure 2.6).

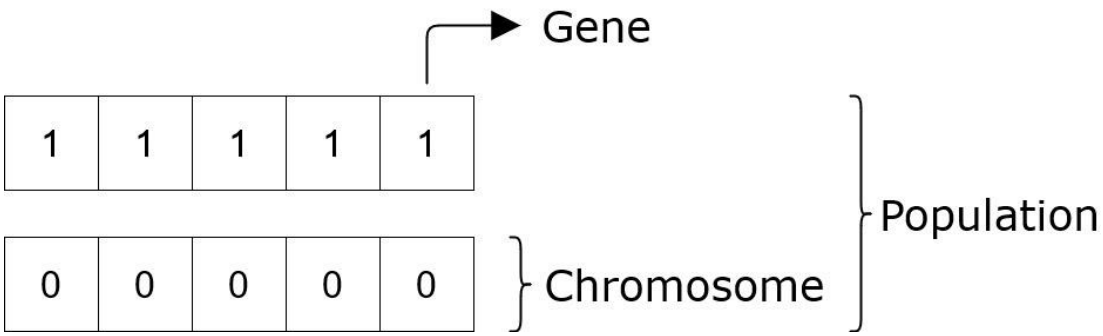


Figure 2.6: Chromosome, genes, and population

- Fitness Function: Fitness Functions are functions to derive better solutions to subsequent generations. It determines how well a solution competes with others in terms of being the fittest one.
- Selection: After every generation, a set within the solutions are selected according to Fitness Function. Selected ones then are passed to the subsequent generation.
- Genetic Operators are crossover, mutation, and termination. In the crossover phase, two parents come together and create offsprings by exchanging genes until a certain point on the chromosome (Figure 2.7). After this operation, newly created offsprings are added to the solution set. This operation helps to create better solutions and varies the solution set regularly.

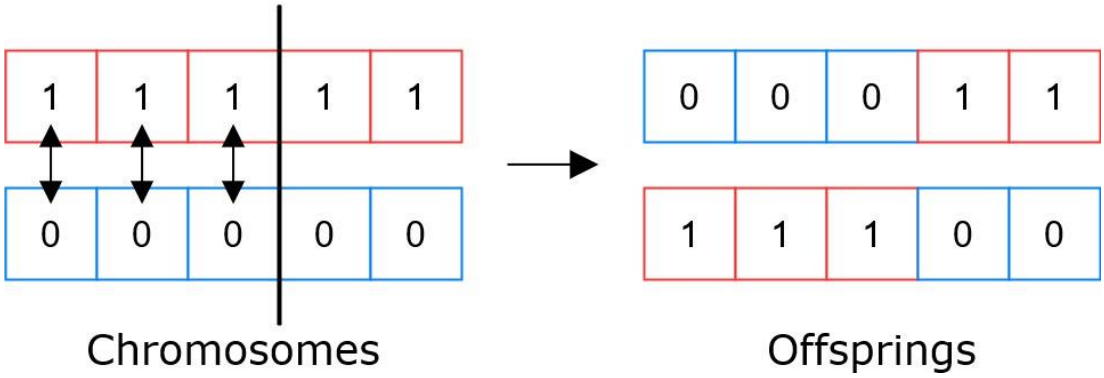


Figure 2.7: Crossover Genetic operation creating new offsprings

In some of the crossover processes, some genes change with a very low rate of probability which is called a mutation (Figure 2.8). Mutation creates diversity in the solution set to have better solutions in the end. However, the rate of probability of mutations should be set correctly. A low probability of mutations can lead to premature convergence that results in worse solutions, and a high

probability of mutations can lead to loss of some valuable solutions and harm to the overall process.

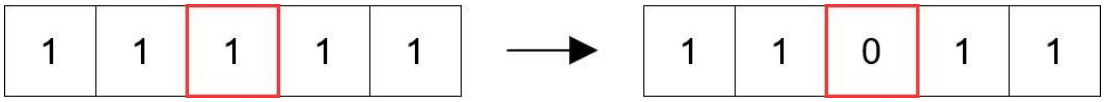


Figure 2.8: Mutation of a gene on chromosome

Termination is the process of stopping producing new results. Termination can be decided in various situations such as new offsprings becoming very similar to previous generations, meeting minimum criteria satisfy the problem, the lack of computational or managing power, or the decision of termination after a fixed number of generations reached.

Using genetic algorithms, urban analysis and growth prediction studies have been made for the cities, Provo, Utah, USA (Balling et al., 1999); Taipei, Taiwan (Feng & Lin, 1999); Phoenix, Arizona, USA (Jenerette & Wu, 2001); Jisperveld, The Netherlands (Stewart et al., 2004); Tongzhou, China (Cao et al., 2012); Bodegraven, South Holland, The Netherlands (Stewart & Janssen, 2014); Gaoqiao, Zhejiang, China (Liu et al., 2015). Some studies integrate genetic algorithms to define or select the precise solution sets from the outcomes of cellular automata models (Al-Ahmadi et al., 2009; Al-Kheder et al., 2007; Foroutan & Reza., 2012; Jenerette & Wu, 2001).

One of the other generative design techniques that are used to computationally model an urban pattern change and prediction is shape grammar. Shape grammar is a set of rules to generate new geometric shapes, forms, and patterns from an initial shape (Stiny, 1980). It consists of four major elements; shapes, symbols, shape rules, and initial shape. Basically, a set of rules is applied to the initial shape to generate a new shape and the set of rules repeat on the last generated shape to create new generations of shape (Figure 2.9). Shape grammars are used in subjects such as architectural design problems, simulating Turing machines, music scores, linguistics, chemistry, painting, and sculpture (Stiny & Gips, 1971). There are also some examples of shape grammar algorithms in urban analysis. Some of the urban analysis using shape grammars takes place in Medina of Marrakesh, Morocco (Duarte et al., 2006); Alentejo, Portugal (Beirão & Duarte, 2005); Pittsburgh, Pennsylvania, USA (Flemming, 1987); Punggol, Singapore (Jan et al., 2008).

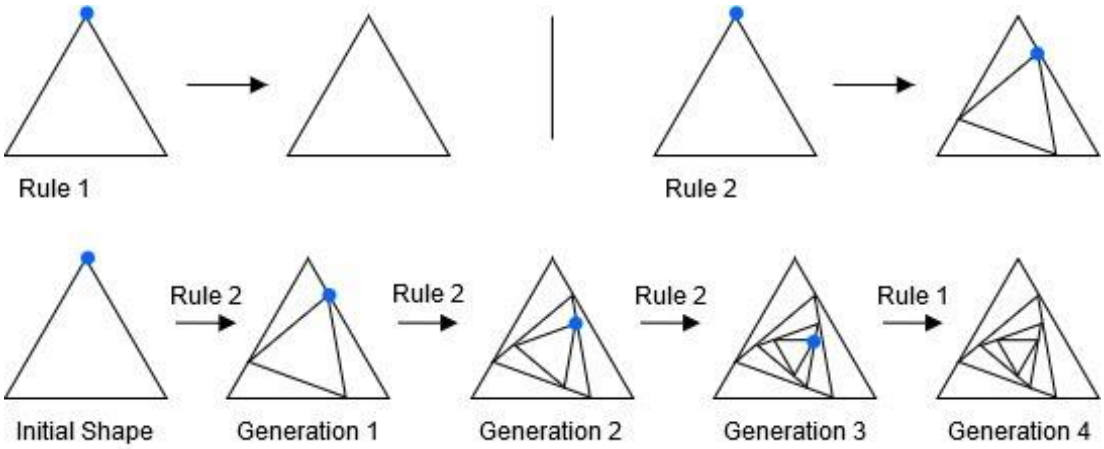


Figure 2.9: An example of shape generation using shape grammars.

Swarm intelligence imitates a common mind that is formed by the collective behaviors of individual agents (Singh & Gu, 2012). Most of the time, swarm intelligence algorithms have been used along with cellular automata models to evaluate the results and improve the overall accuracy of the models (Naghibi et al., 2016; Feng et al., 2011; Khalilnia et al., 2013).

3. GAME ELEMENTS

The developed game, called On the Grid, is an augmented reality city planning board game. The players prepare the game board and use their devices to define the game area to reflect augmented game objects and they can follow the gameplay using both their device's screen and physical objects. They participate further by using tangible game elements by correlating physical and digital dimensions. Game development processes consist of different elements starting from an idea or concept to a unique end product. These elements of the games not only stay as they are but also interact with each other. That's why it's not inaccurate to define games as systems. In a game system, game elements interact, interlace, and act as a whole to create a complex system. There are different approaches to analyzing game elements. In this study, game elements are classified under 5 different categories according to previous studies: Play and Game (Avedon & Sutton-Smith, 1971; Salen & Zimmerman, 2004; Huizinga, 2014), Challenges and Goals (Constikyan, 2002; Crawford, 2003; Fullerton, et al., 2004), Rules (Brathwaite & Schreiber, 2009), Interactions (Constikyan, 2002; Adams, 2010; Fullerton, et al., 2004) and Representations (Sillaots, 2015).

If an element of the game interacts with an external factor that system is described as an open system, if they only interact with each other then it is called a closed system. The designed game named On The Grid can be classified as an augmented reality board game, and while its rules and goals can only interact with each other like a closed system, the gameplay, and look of the game correlates with exterior elements such as players or the context of where the gameplay happens. Therefore, the gameplay has a rich and immersive structure where the players themselves and the environment surrounding them are both included in the gameplay experience, physically and digitally.

3.1 Play and Game

Every game is made for someone to play. As a concept, play and game complement each other. Play happens inside games but also it can happen on its own, without

games. Play as behavior is older than the culture and it is not something we learn or teach to ourselves. Even animals play and they do it without the need of human beings to teach them (Huizinga, 2014). Huizinga (2014) examines play under three main characteristics:

- Play is a voluntary activity.
- Play is not real life or ordinary.
- Play is limited and secluded.

The act of play is voluntary. The activity is not forced and a forced playful activity is not considered as a play but an imitation of play. At the same time, participants know that they are playing a game and they are pretending to have fun. There are distinctions from real-life events. The play has another reality apart from the real world with its own rules and scenarios. The players get into a playful mindful set when they're playing. The reality of play end when the player stops so play activities have certain starting and ending times and are limited between them (Huizinga, 2014).

3.2 Challenges and Goals

Games, on the other hand, are just a small subset of the play activity but also contains play in itself (Salen & Zimmerman, 2004). Just like play, games have various characteristics. One of the main characteristics of the games is challenges and goals (Fullerton et al., 2014; Crawford, 2003; Costikyan, 2002). Every game offers some kind of challenge to its players. Challenges are tasks that players are supposed to fulfill. The designed game, On the Grid, offers a challenge of time. The players have to achieve their objective before a certain amount of turns. The players try to reach a winning cap by increasing the total number of one of the 4 city units; residential, commercial, recreational, or industrial. The winning cap is a bounded variable to the size of the gameplay area (Table 3.1). The difficulty of a game session can be changed before starting the gameplay. The challenge of each gameplay session is shown to the players before they start players.

Table 3.1: Different goals of On the Grid according to grid size.

Gameplay Area (Grid Size)	Winning Cap	The percentage for target cells
10 x 10	40	40.00 %
12 x 12	50	41.66 %
16 x 16	100	39.06 %

A gameplay process requires managing resources to overcome the conflicts by calculating risks. Since there is a goal to be reached before a certain time, the first resource is the time itself. Every action that is made by a player counts as turns and is limited. On the other hand, after each player ends their turns, the artificial intelligence (AI) makes its move. The AI uses cellular automata algorithms which adds randomness and unpredictability to the game and creates the conflicts. After AI's turn, the players can inspect desired game cells and create a strategy according to risks.

3.3 Mechanics

Every game consists of mechanics that designate how game elements work and how players participate in gameplay (Figure 3.1). These game mechanics, which are also called rules, allow players to take action, have their limits, and win or lose at the end. Rules also provide transitions between game objects and value and determine how artificial intelligence works and reacts. Game mechanics can be studied in 5 major topics: setup, victory conditions, sequencing, player actions, and game views (Brathwaite & Schreiber, 2009).

Figure 3.1: Flowchart showing the playing process of On the Grid.

Setup rules are the initial rules of a game where they determine the beginning of the gameplay event. On the Grid allows players to choose between setup rules. Before starting the gameplay, the players should set-up the game board and augmented reality supported device (Figure 3.1). After physical and digital setup is completed, the number of players, urban scenario, winning conditions, gridded gameplay area size, and time limit can be determined from the game menu and game rules are set accordingly (Figure 3.2).

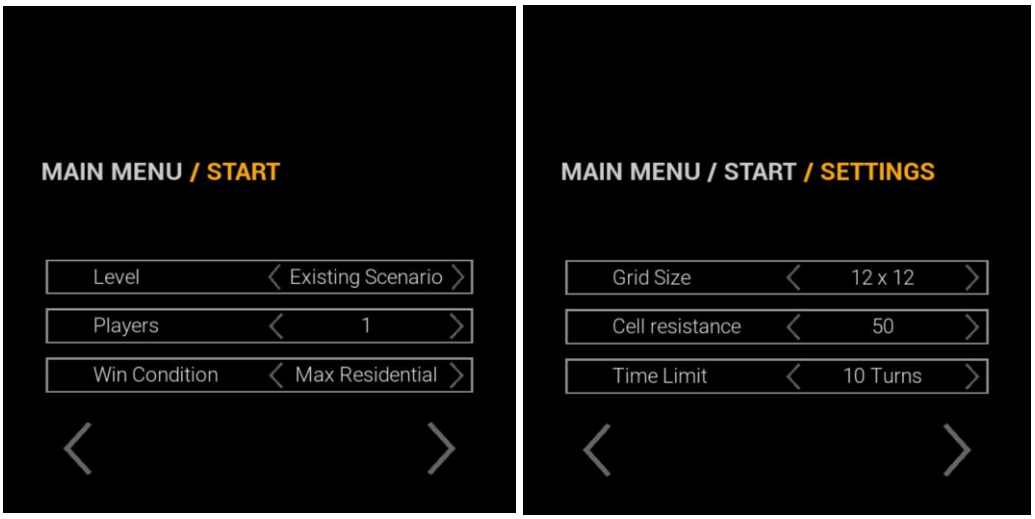


Figure 3.2: Setup rules of On the Grid before starting the game.

Flexible setup rules allow players to have more options and increase the range of the player profile that the game is targeting. On the Grid is a type of serious game, the desired difficulty, type of the challenge, or the necessity of cooperation can be different for some players. While the players with a growth mindset can always increase the amount of challenge, others with fixed mindsets would stick with easier scenarios. It's also important to determine what should the player achieve to win the game. A clearly defined win condition and feedback are important for preparing the players for the upcoming challenge and also help the players to devote themselves by making the true moves for the true outcome (Salen & Zimmerman, 2004). At the end of each game, the players win or lose according to the final number of units that are targeted. If the players are on a winning or losing streak the difficulty is adapted on the next game to keep the players motivated.

Just like initial rules, the gameplay process also needs its own set of rules. Sequencing is the set of rules that determines what happens, when, and with which action. Sequencing directly affects the flow of the game. On the Grid is a turn-based game, where the player and AI actions happen consecutively (Figure 3.3). Gameplay activity happens accordingly:

- Step 1: An information screen demonstrates the goal, time limit, and targeted unit type to the players on the device screen.
- Step 2: All players create strategies. Information popups can be opened by clicking on the individual cells on the device screen.

- Step 3: The player puts the cube token on the desired grid position and the device registers the player's move.
- Step 4: The cube token is removed from the game board area and the player puts the turn token to the same place where the cube was. Turn token contains the player tag and turn number, and unit type.
- Step 5: If there are more players that haven't played this turn, then they also make their moves as stated in steps 3 and 4.
- Step 6: Artificial Intelligence makes its move.
- Step 7: Step 2-6 is repeated until the turn limit is reached.
- Step 8: Players win or lose on the condition of meeting the challenge.
- Step 9: If there's a win or lose streak, the difficulty of the game is adapted.

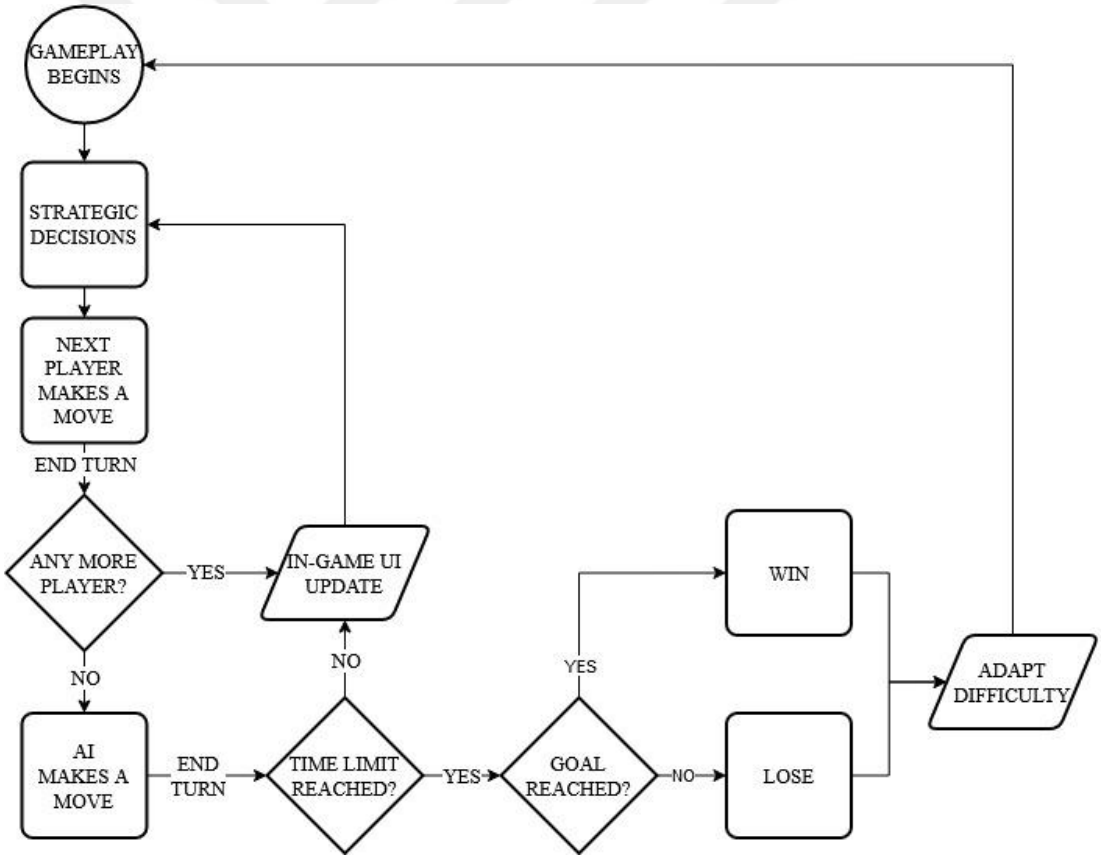


Figure 3.3: Gameplay sequence of On the Grid.

Rules allow players to make their actions. It manages when and how a player can take action. Adams (2010) describes player actions as events that happen firsthand in a game environment, following by user inputs. To make an action, players have to give

proper inputs at the proper time. Players of On the Grid make their gameplay-related actions on their turns by using the cube token and change a specific cell. Changing a cell is the action of the players but the simulation of cellular automata-based rules in artificial intelligence's turn is the consequence of their actions. Being an augmented reality board game, users can make their actions both digitally and physically. Players use user interface elements on the screen to navigate through menus, changing options and settings, ending their turns, and using information popups of the cells. On the other hand, cube token, game board, and gameplay area are the physical elements that players make their actions with. The amount of information and interactions available for the players is also bounded by rules. There are some information rules where players can discover by in time or by doing specific actions, and some of the data is never shown to the players and run in the background to maintain the flow of gameplay.

3.4 Interactions

Players create interactions with the game elements by giving input into the game and these inputs form actions that correlate events in the game world (Adams, 2010, Figure 3.4). Every game is interactive and being interactive what makes a game different than a puzzle (Costikyan, 2002). A puzzle is static and a game is interactive even though games include small puzzles in them. A player input alters the game state and games give feedback to the player. This process is reciprocated and actions of the players cause events so the consequences of the actions take place in games.

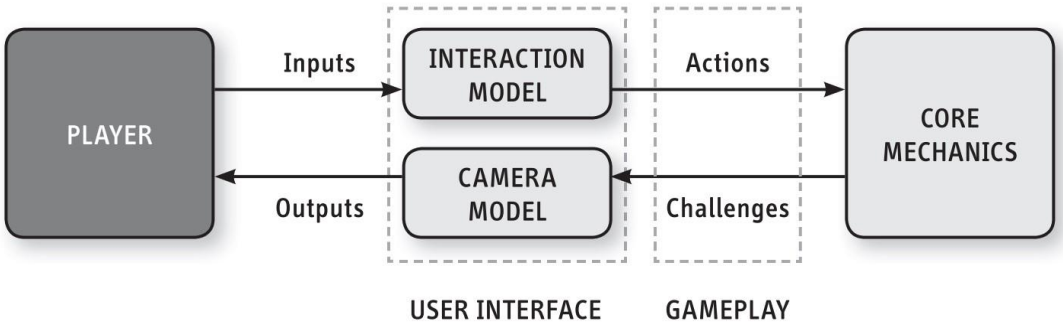


Figure 3.4: Player interaction with the gameplay. (Adams, 2010, p. 38).

Player interactions happen between players and gameplay, game world, user interface, or with other players (Adams, 2010). Gameplay related interactions are made through challenges and actions; game world related interactions are provided by interaction models and feedback system through camera models; user interface interactions create a bridge between the core mechanics and the players; lastly, players can interact with each other according to competition mode (Sillaots, 2015, figure 3.5).

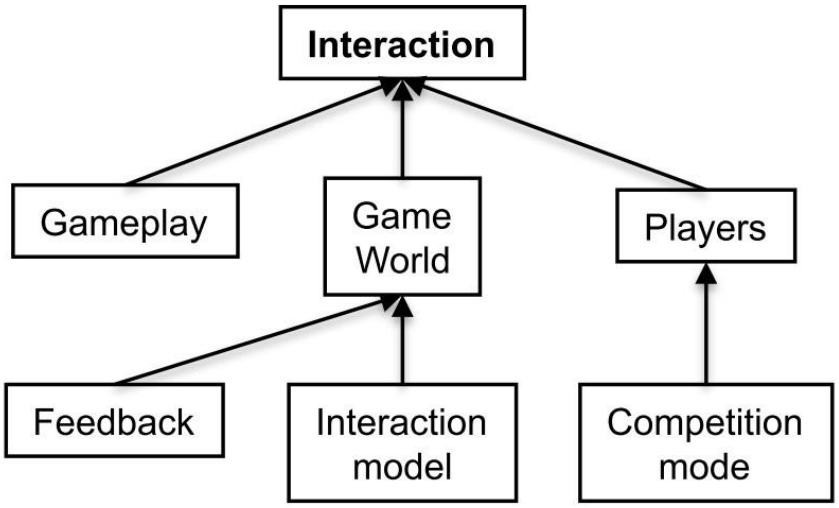


Figure 3.5: Interaction patterns with gameplay, game world, and other players (Sillaots, 2015, p.5).

Interaction models are used by the user interface to fill the gap between player input and gameplay mechanics. Adams (2010) defines five different types of interaction models in games: avatar-based, omnipresent, party-based, contestant, and desktop interaction model. Avatar-based interaction models reflect players' actions to a character in a game world. This character can be a human, humanoid, or anything that players can control such as a vehicle. These players are only in one part of the gameplay area and move accordingly. On the other hand, the omnipresent interaction model allows players to take action in different parts of the game world (Adams, 2010). The proposed interaction in On the Grid can be considered in Adams' omnipresent interaction model as it gives the opportunity to make moves in any area of the game board when it's player's turn. No area is hidden from the player and the player can use the host device to have control over the camera. Party-based interactions usually take place in role-playing games where the player form parties and work together to overcome the challenges. Contestant interaction models involve choice-based

gameplay scenarios with no movement at all. Lastly, desktop interaction models imitate real-world desktops environments (Adams, 2010).

There are different kinds of interactions that happen between players during gameplay (Figure 3.6). The patterns of interaction between a player, the game, and other players can be listed as player vs game, player vs player, multilateral competition, unilateral competition, team vs team, multiplayer co-operative competition, multiple individuals vs game (Fullerton et al., 2004).

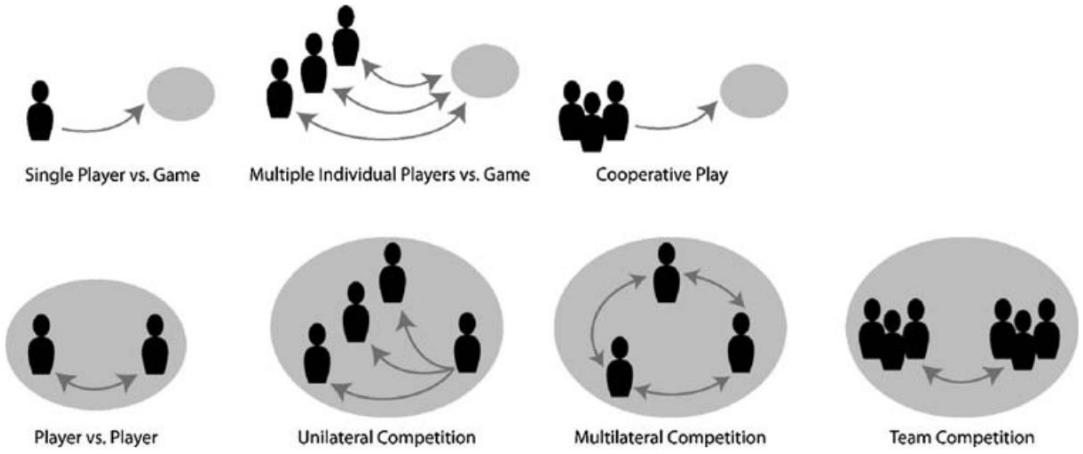


Figure 3.6: Player interaction patterns (Bekker et al., 2010, p. 392).

Player vs game interaction patterns or single-player games takes place when there's no interaction with other players at all. This model has no social interaction capabilities and players only react to game elements with their allowed actions to overcome the challenges. On the other hand, players can team up against the game and create strategies together to overcome challenges in cooperative games. On the Grid allows both single-player and cooperative play possibilities. Players from different backgrounds can team up and play against the unpredictable AI. A wide range of backgrounds from the players create consensuses or conflicts on the strategy-shaping processes and enrich the gameplay experience. It also helps players who struggle with achieving goals by decreasing the difficulty. Other types of player interaction patterns include players playing against each other such as chess. If more than one player competes against another player it's called unilateral competition and more than two players competing with each other called multilateral competition. Team competition structures form groups and players in the teams compete with the other teams.

3.5 Representations

Games create their reality using the reflections from real-life situations. Salen & Zimmerman (2004), discusses the connection between games and representation in two ways where games are defined as representations, and on the other side, games are representing the external world in their fictional worlds.

On the Grid has various representations. First of all, it is a representation of chosen real-world urban patterns, defined parts of a city and it offers pre-made scenarios to players. For the scenario, organized industrial site in Dudullu District, Istanbul, Turkey is selected. The existing pattern of the area is analyzed through GIS data and it is transferred into the game after following processes below:

- The selected site is focused on the map and different zones are marked in different colors using the GIS data. Residential areas are marked yellow, commercial areas are marked blue, recreational areas are marked green, and industrial areas are marked red. Areas with no information are marked white and they are considered as habitable places. The main transportation axes and water bodies are marked dark gray and considered as inhabitable places so players are not allowed to change the site types of those areas.
- Colorized and rasterized GIS data is transferred into texture creation software, Substance Designer (Figure 3.7).

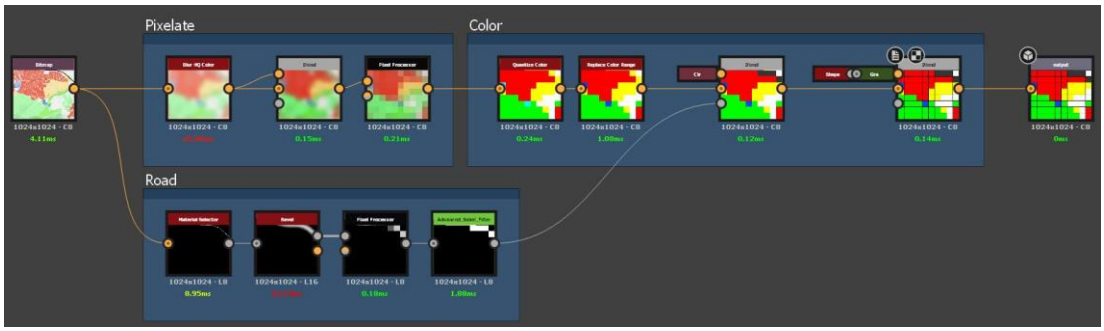


Figure 3.7: The graph showing the pixelation process of rasterized GIS data.

- Rasterized GIS data is blurred and pixelated with a resolution variable. The resolution variable is set to 10 by default so the blurred image is divided into 10 by 10 grid. The resolution is set to 10, 12, or 16 according to the selected initial rules of the game (Figure 3.8).

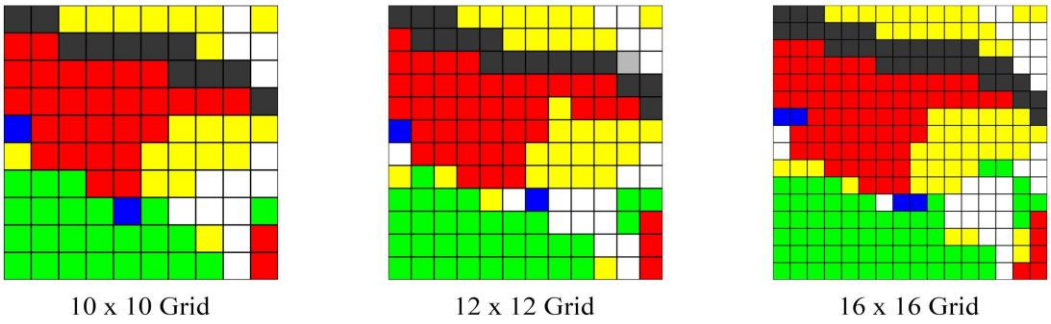


Figure 3.8: Different grid sizes for the pixelation process.

- Colors are corrected and gridlines added blended on the pixelated image (Figure 3.9). The pixelated end product later imported into the game to create the scenario.

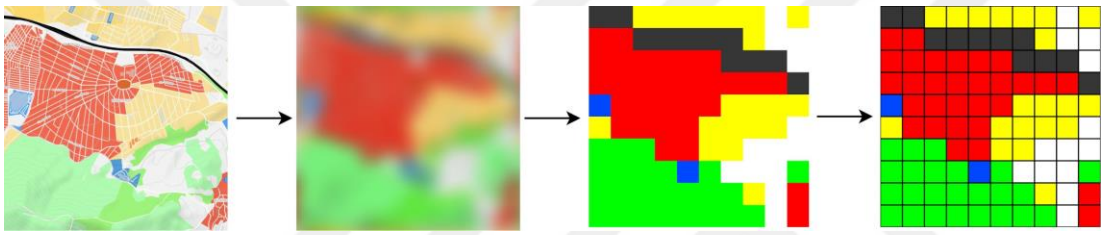


Figure 3.9: Pixelation process of the rasterized image created using GIS data.

To maintain the artistic and functional consistency, the color palette carried over through the game. It serves an informational value to the units and game assets are colorized accordingly. The 4 different unit types that use real-world concepts are residential, commercial, recreational, and industrial units (Figure 3.10).

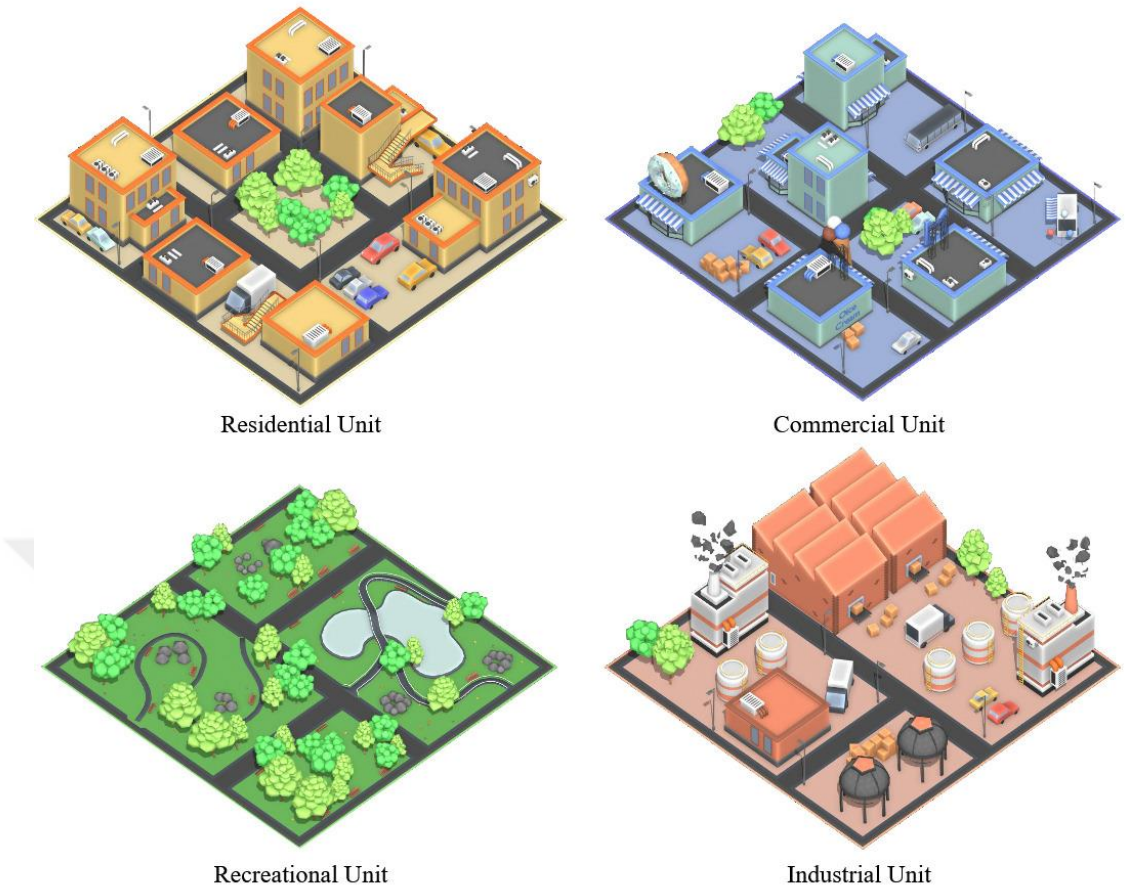


Figure 3.10: Orthographic views of color-coded game assets for different cell types.

4. CASE STUDY: DESIGNING THE CITY BUILDING GAME

This thesis is focused on creating a serious game for architects, designers, and local management by designing an augmented reality-based city managing board game. The game aims to create a participatory environment for people from various backgrounds while strengthening the urban intervention reflexes of the players. The secondary aim of this project is combining augmented reality technology with board game systems and creating an innovative interaction-workspace for the players. The main computational approach for the urban simulation model in the game is selected cellular automata, as it's one of the most common techniques used for many different urban prediction models according to the literature review that has been made. As a result of this, an algorithm that defines urban transformation in time is created using cellular automata. Previous chapters mentioned serious games, games with urban design concepts, computational techniques used in urban growth models, types of players, and elements that form a game so these aspects are taken into consideration in the following chapters (Figure 4.1).

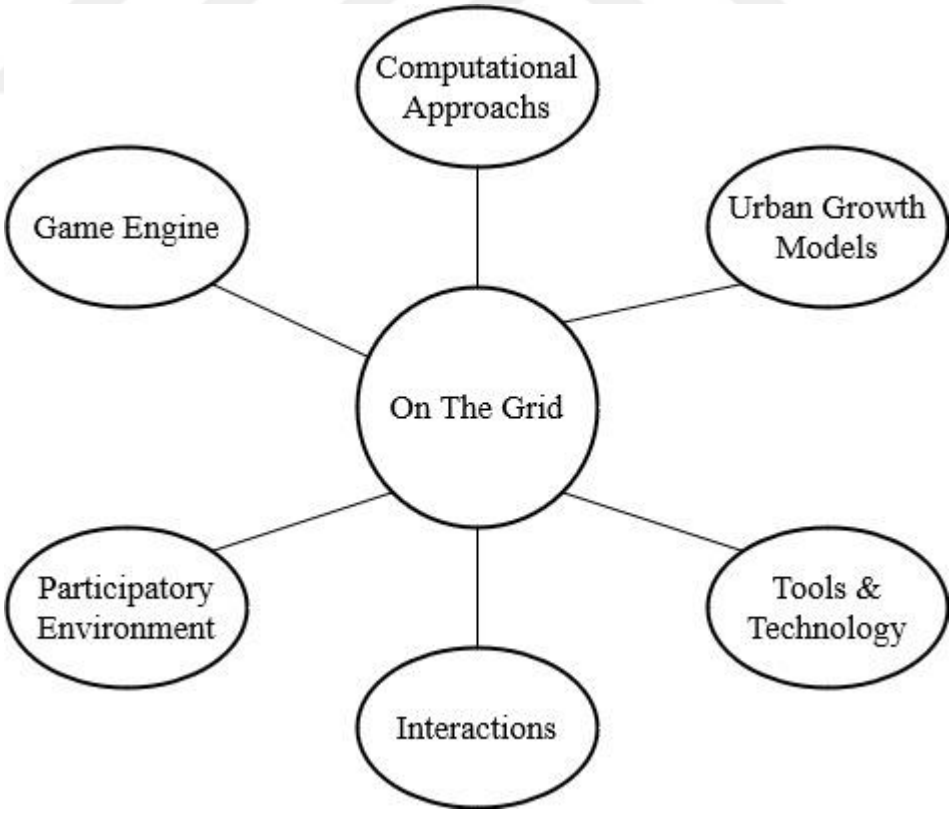


Figure 4.1: Relation diagram of On the Grid

The games and projects that are inspected on the previous chapter under games with urban design concepts discuss the aims and possibilities of these projects. One of the common concerns of these projects is them using only digital or only physical aspects and not both at the same time. The concept of using generative design techniques and combining digital and physical in a tabletop environment is the key feature for creating a participatory environment with enriched interactions.

4.1 Tools and Technology

On the Grid is uses augmented reality technology along with modular physical assets. The game consists of a downloadable application on handheld devices. Developing an application for various AR supporting handheld devices requires a game engine and the application packages are built and ported on the game engine for different platforms.

4.1.1 Game Engine for On the Grid

The core functionalities and algorithms of games are executed in real-time. A game engine is a development workspace and provides software frameworks to create games. Game engines offer various tools in a combined workspace. This tool consists of codes, functions, 3D and 2D models, textures, sounds, user-interface, animations, and many more. Unreal Engine is one of the industry-leading game engines in the market. It is first developed for the game named Unreal in 1998 by Epic Games. The game engine later launched for developers with a royalty model where developers share 5% of game's revenues with Epic Games. The source code of Unreal Engine is available for the developers and it's fully editable for the final products. Unreal Engine is not limited to the game development industry and it's also used by media companies, individual game developers, film producers, architects, and hobbyists (Figure 4.2).

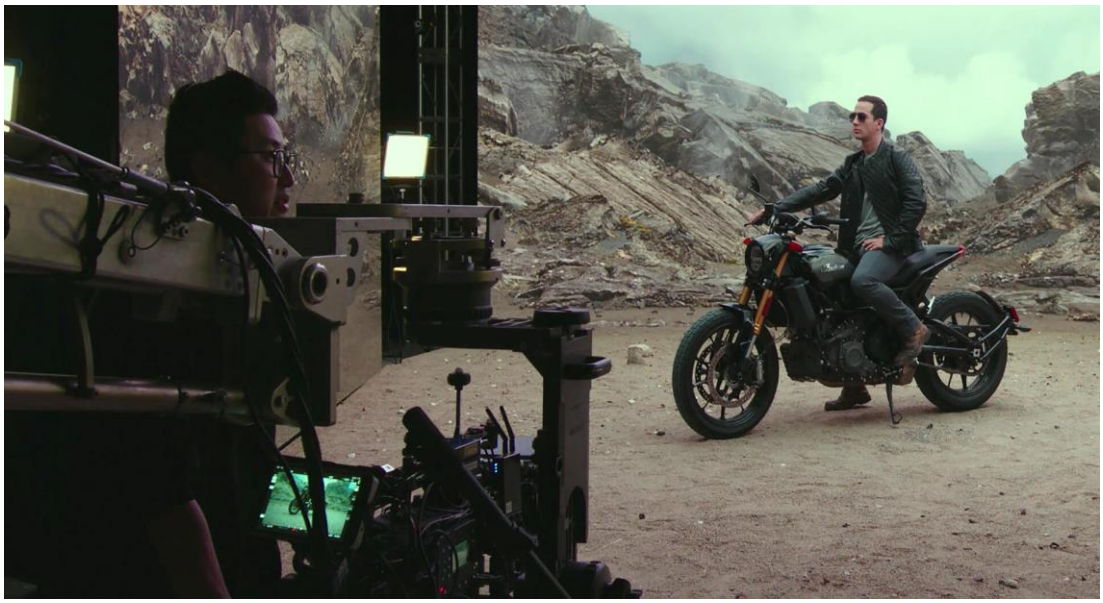


Figure 4.2: Dynamic environments in the film industry with Unreal Engine [Url-4, 2019].

Unreal Engine provides a visual scripting system in itself called Blueprint. The blueprint visual coding system allows developers to create fully-functional games and environments without coding (Figure 4.3). The node-based structures are very functional to those who are not very familiar with coding. Blueprints system, removes the necessity of syntax knowledge of C++ coding language. The Blueprints visual scripting system uses an object-oriented programming foundation and creates an easy interaction interface with designers. It can be implemented in projects with C++ coding language that Unreal Engine uses.

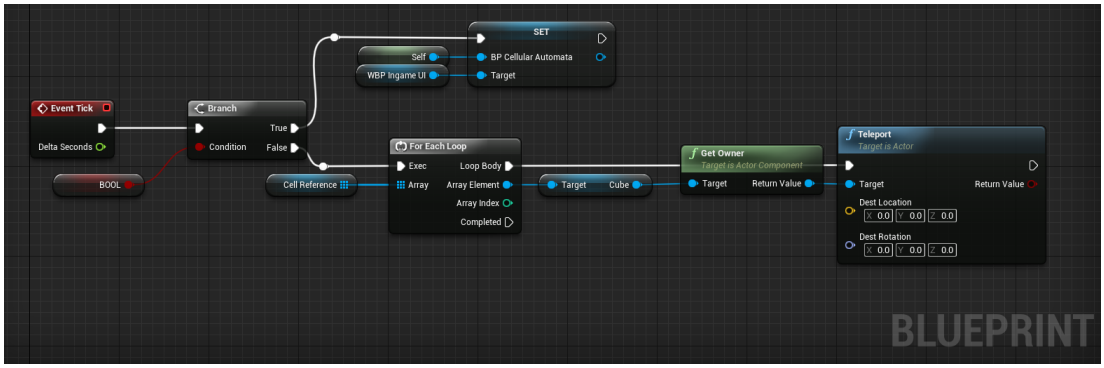


Figure 4.3: Example of the Blueprints visual coding system.

The ease of use with visual scripting, augmented reality support, free-to-use royalty model, and flexibility in building and porting options made it suitable to use for the designed game.

4.1.2 Augmented Reality integration in On the Grid.

Augmented reality is the superposition of rendered images and the real-world surrounding environment. The augmented reality on handheld devices is mostly made through device camera. There are different Augmented Reality software development kits for different platforms. For development purposes, Google's ARCore is selected as it can run on various brands of handheld devices that use the Android operating system. Google ARCore is also suitable for Unreal Engine integration and comes as a plugin to the engine. The ARCore plugin brings required functions, events, variables, and settings to the engine. The augmentation process happens through images. The device that the software running on scans the target area and compares results with designated candidate images. The board of the game has high contrast symbols on the edge so that it can be detected easily by the ARCore algorithms. Image detection happens on each frame and software makes the augmentation accordingly. The position and rotation information of the augmented image in real-world space are gathered to be used in computations.

4.1.3 The game board and tangible game elements

On the Grid contains a board, a cube token, player tokens, turn tokens, and unit tokens. The board is required to start the game and defines the gameplay area. Players use the cube token to make their moves. The cube token has 4 different target images. Each image augments one of four different unit types, residential, commercial, recreational, or industrial (Figure 4.4). After each turn, players combine the turn, player, and unit tokens together to mark the move they've made. Turn, player, and unit tokens are modeled and printed using a 3d printer (Figure 4.5).

Figure 4.4: Different images on cube token to augment unit types.

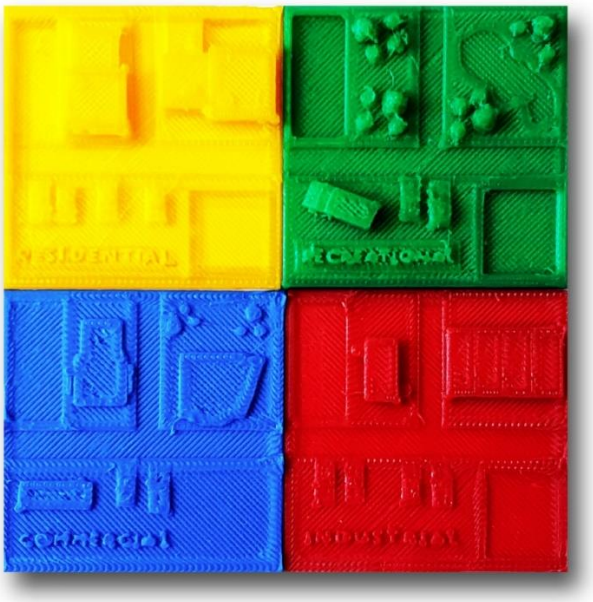


Figure 4.5: 3D Printed game tokens fabricated for On the Grid.

4.2 Game process breakdown

The gameplay process is accompanied by user interfaces and menus (Figure 4.6). After the initial setup of the board and distribution of the tokens to the players, the application is started.

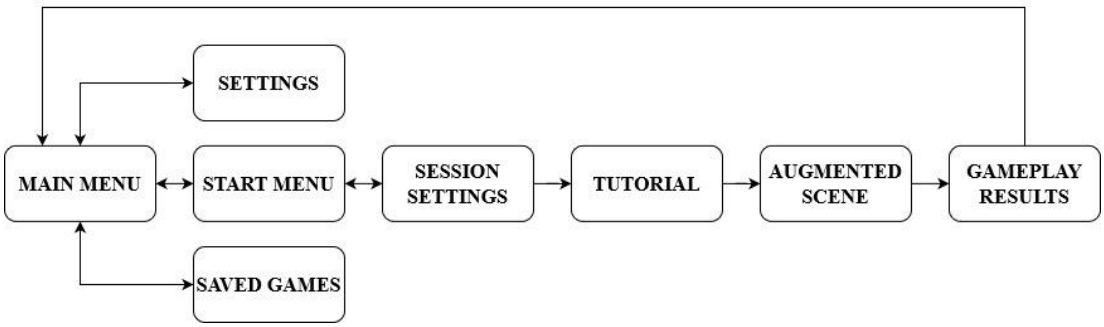


Figure 4.6: Menu structure map of On the Grid.

The game starts with a splash screen and the main menu appears on the player's screen. Users can reach general options for the game from settings menus such as toggling sound or tutorials. The other menu that can be reached from the main menu is the saved games menu. In this menu, players can see stats and results of previous games. The Start Menu is the first step for starting a new game session. In this menu, the number of players, game scenario, and winning condition options take place. Game scenarios

consist of existing scenarios from real-world urban patterns or create procedurally generated environments (Figure 4.7).

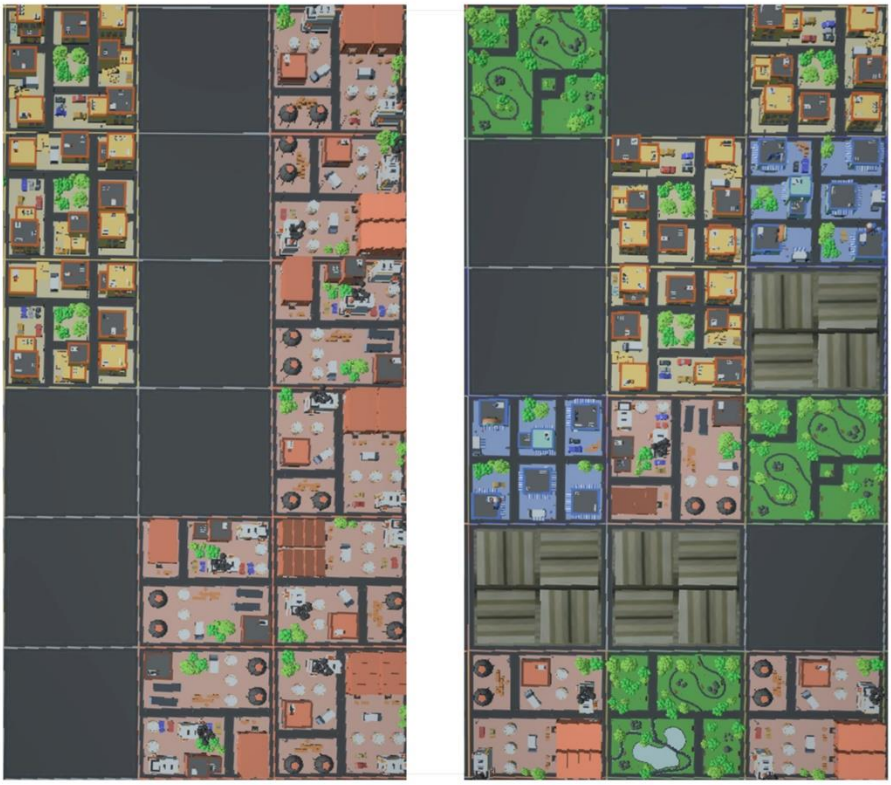


Figure 4.7: Two snapshots from different scenarios: existing scenario (*left*) and generated scenario (*right*).

The gameplay begins after tutorial and there're informative user interface elements on top of the augmented screen (Figure 4.8). Players can save their current game state, reach information about unit numbers of each type. The end turn button is activated when the player makes a move. Until then, it stays deactivated and reminds players to make their moves.

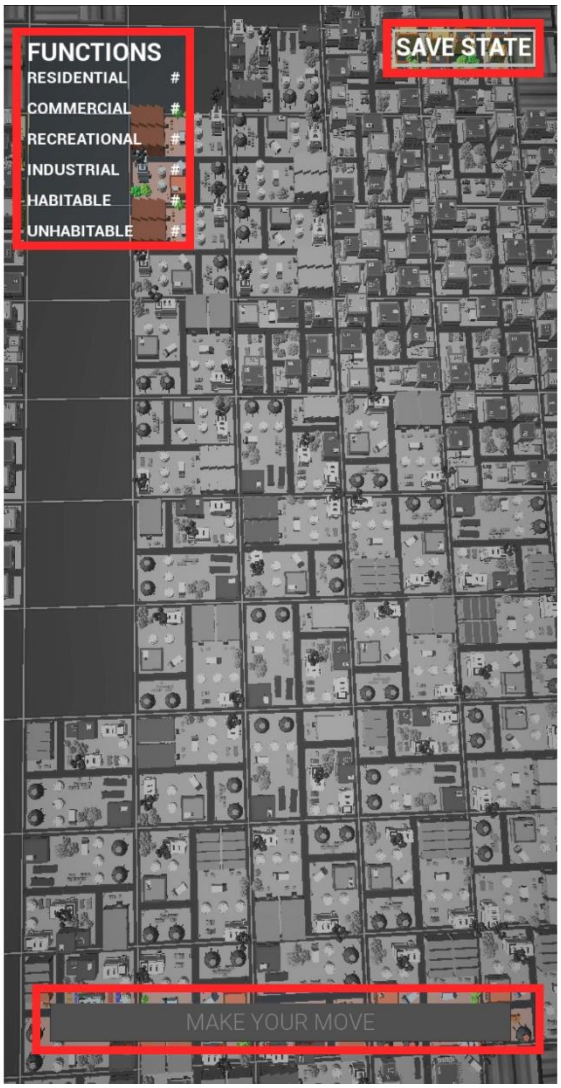


Figure 4.8: User Interface Elements on the augmented scene screen.

4.3 Creating the algorithm of the game

The main menu and settings are determinants of variables and game flow. The options on these menus change the variables for the gameplay. Grid size, winning condition, number of players, and more are used for changing variables in the game so when a player makes a change in these settings, global variables are affected. The gameplay contains lots of different functions and events. One of the first functions fired is the “Start Session” event. It gets all of the information from the game menus and uses them to create the correct scenario for the rules (Figure 4.9).

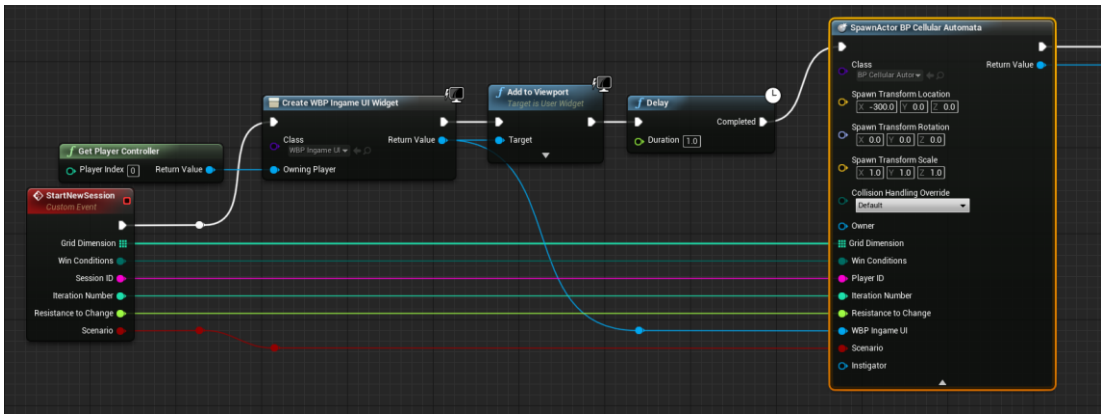


Figure 4.9: Using start menu values while spawning Cellular Automata actor on starting a new session.

When the camera is launched and menu user interfaces are removed from the screen, the AR session starts. In Unreal Engine, players are represented in the world as an actor called pawn. In AR games, these pawns are specialized as AR pawns with AR and handheld device camera functions. The AR pawn of On the Grid is named as BP_ARPawn for persistence in naming conventions. Events happen at regular intervals, usually, one time on every frame is called ticking events. On BP_ARPawn actor, event tick is executed on each frame (Appendix A). At first, the handheld device's camera starts to scan the environment. The handheld application's understanding of the surrounding environment is specialized as it tries to detect geometries in the world space. The function "Get all AR geometries" retrieves all of the real-world geometries detected by the AR system. After getting all of the real-world geometries as an array, Google ARCore's Augmented Image object reference is filtered by trying to cast every item of the array. If the detected object is Google ARCore's Augmented Image object, then it's saved as a variable named AugmentedImage. AugmentedImage's location and rotation in the world are also saved as variables. There are different reference images in Google Augmented Image data asset for detecting game board, cube token, and to be used in functions. Each image is listed with a distinctive name in data asset so a branch follows the nodes on event tick and check if the name the detected image is the board. If the detected image is the game board, the algorithm repositions the cellular automata actor on to the board. The reposition is made through a function.

Functions in Unreal Engine can have inputs and outputs. To align the actor on the board the normal orientation vector, position vector, and rotation are fed as inputs to

the reposition function in the cellular automata actor. In the reposition function, the rotation input vector is broken into the roll, pitch, and yaw values (*x*,*y*,*z*), and the cellular automata actor's rotation and location are interpolated between current rotation and location. The speed of interpolation is set to low values to give a smooth transition between transforms and prevent jittering of the augmented actor on the board. After reposition is complete on BP_ARPawn's event tick, a "do once" node spawns the information UI on the augmented scene screen and closes the gate, meaning that if this "do once" node is triggered again, the nodes after won't be triggered again until "do once" node is reset.

On construction, a series of events are triggered on cellular automata actors. (Figure 4.10). Cellular Automata actor starts generating cells by spawning another actor named BP_Cell which represents individual urban units. Two "for loops" with the repetition of X and Y grid sizes are used one above another to create a two-dimensional layout (Figure 4.11). Positions of the units are calculated by multiplying their coordinates (*x*,*y*) with unit dimensions (Figure 4.12).

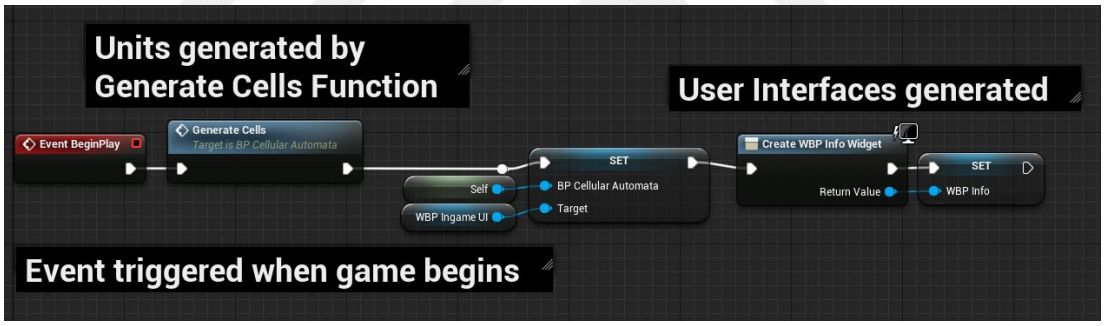


Figure 4.10: Events triggered when users scan the game board with the device camera and gameplay begins.

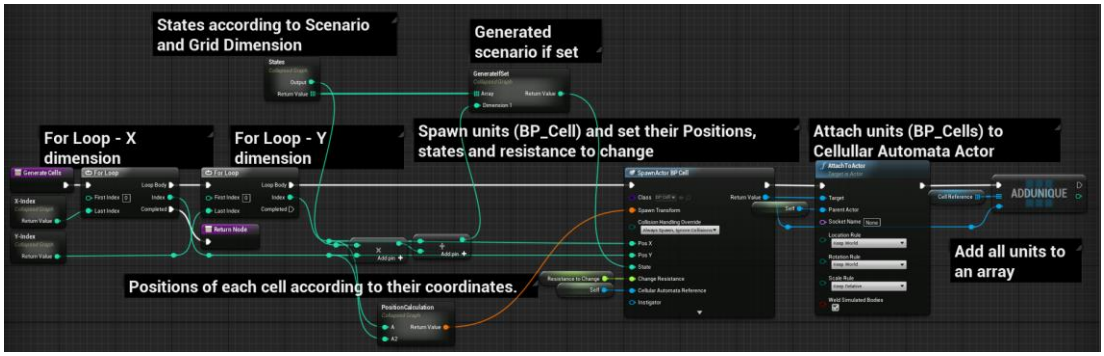


Figure 4.11: Generate cells function spawning every unit on true state and position.

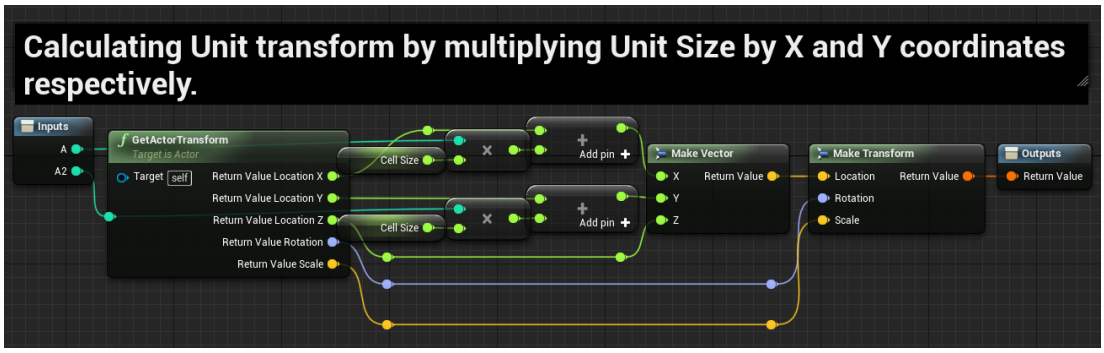


Figure 4.12: Position calculation algorithm for every unit according to their coordinates on the grid.

The units' preliminary states are set according to a scenario and grid size. Every spawned unit is attached to BP_CellularAutomata actor and they are stored inside an array to call them later easily. The Check Overlaps function and setting units' states happen inside an event named CheckTrendingState and it runs for each cell when artificial intelligence makes its move (Figure 4.13).

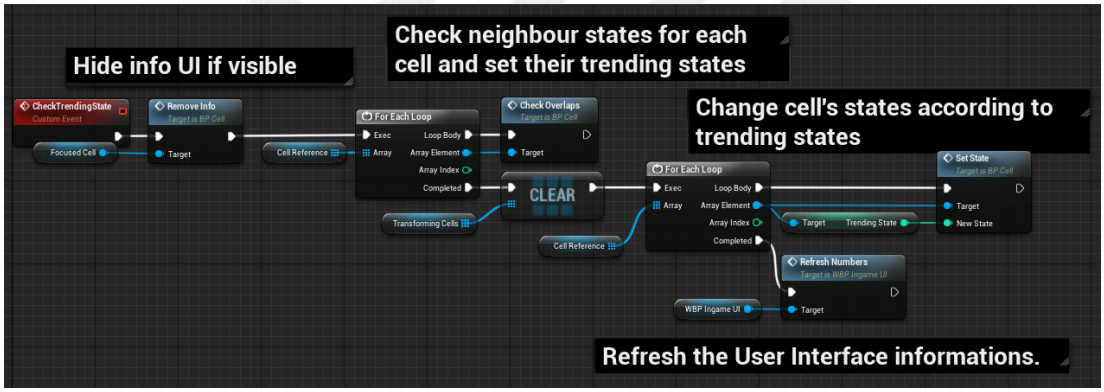


Figure 4.13: Check trending state runs for each unit and change their states concerning trending state.

BP_Cell actors have construction script that determines how they will look. On the construction script, the materials for the static mesh is set and static mesh is also changed according to the state of the individual cells (Figure 4.14).

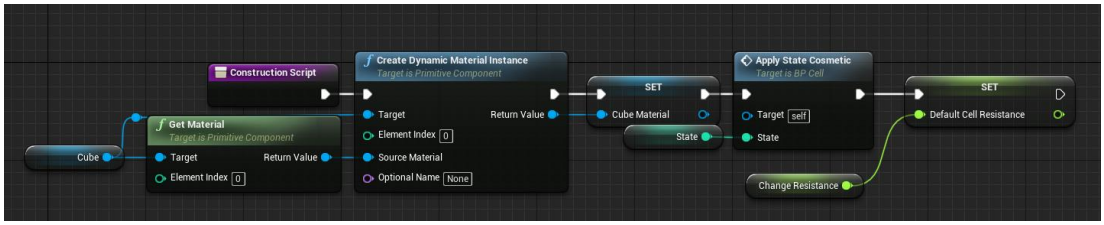


Figure 4.14: Functions executed on the construction script of BP_Cell actor.

Different static meshes are selected for four main unit types (Figure 4.15).

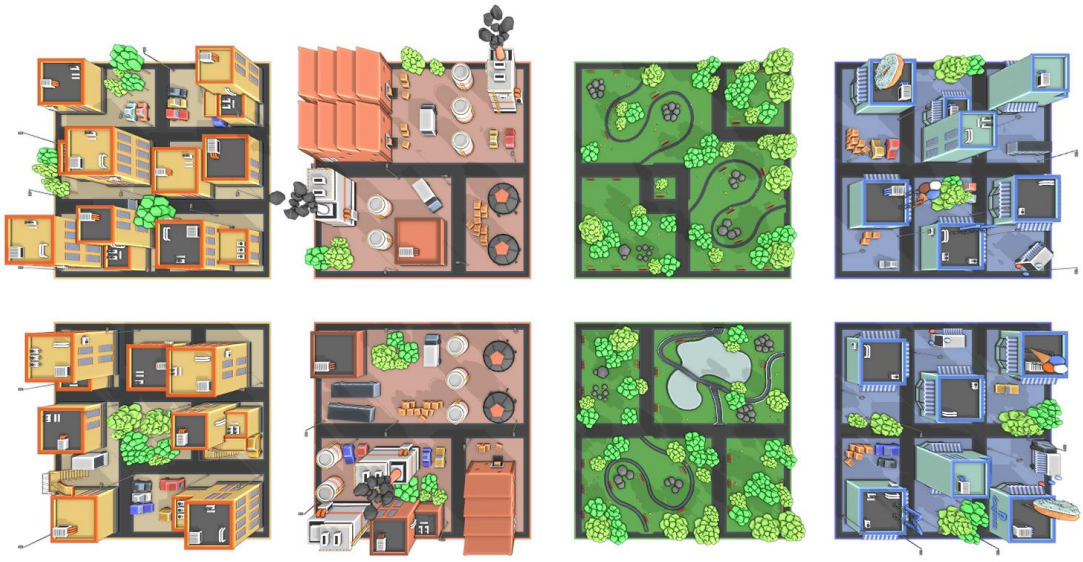


Figure 4.15: Two different alternatives for each unit type.

To create more variations of the cellular automata grid, one of the two variations of a single unit is aligned with the randomly assigned rotation of 0, 90, 180, or 270 (Figure 4.16).

Figure 4.16: Different looks are created for the grid by giving random rotation and static mesh.

Different static mesh alternatives are described in an array and specific array elements set the static mesh of units. The cell actors change the static mesh in reference to their state. This is executed by a function named Apply State Cosmetic. Apply State Cosmetic function changes how cells look in the gameplay area (Figure 4.17). There's an array in cell actor that contains different static meshes. One of the two mesh

alternatives for a unit type is selected randomly. Then the materials assigned and relative rotation of each cell is changed.

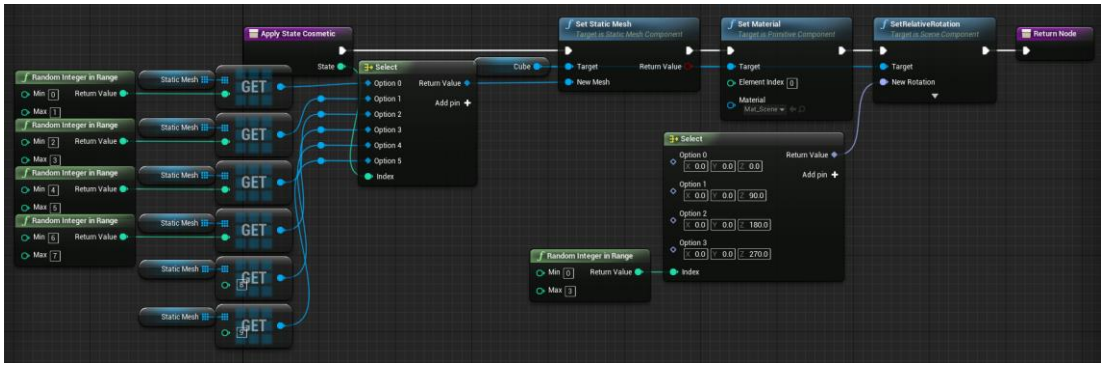


Figure 4.17: Apply State Cosmetics function changing the in-game look of the unit.

Every actor with texture creates a draw call on the graphics processing unit (GPU) of the handheld device. To render an actor, a call is made from the central processing unit (CPU) to GPU, asking it to be rendered. The cumulative If the total amount of draw calls is too high then the GPU of the handheld will be too busy and delay behind CPU. This will cause GPU bottleneck for the game and result in lower framerates. Maintaining high framerates is especially important for handheld devices using a camera to have the user experience with a good rating. Every object sharing the same texture will only cause a single draw call on the graphics processing unit. On the artistic perspective, On the Grid uses nothing but plain colors to texture its assets. Therefore, joining all of the plain colors used in all of the units inside a single texture atlas in Targa format reduces the total number of draw calls so it improves the overall performance (Figure 4.18).

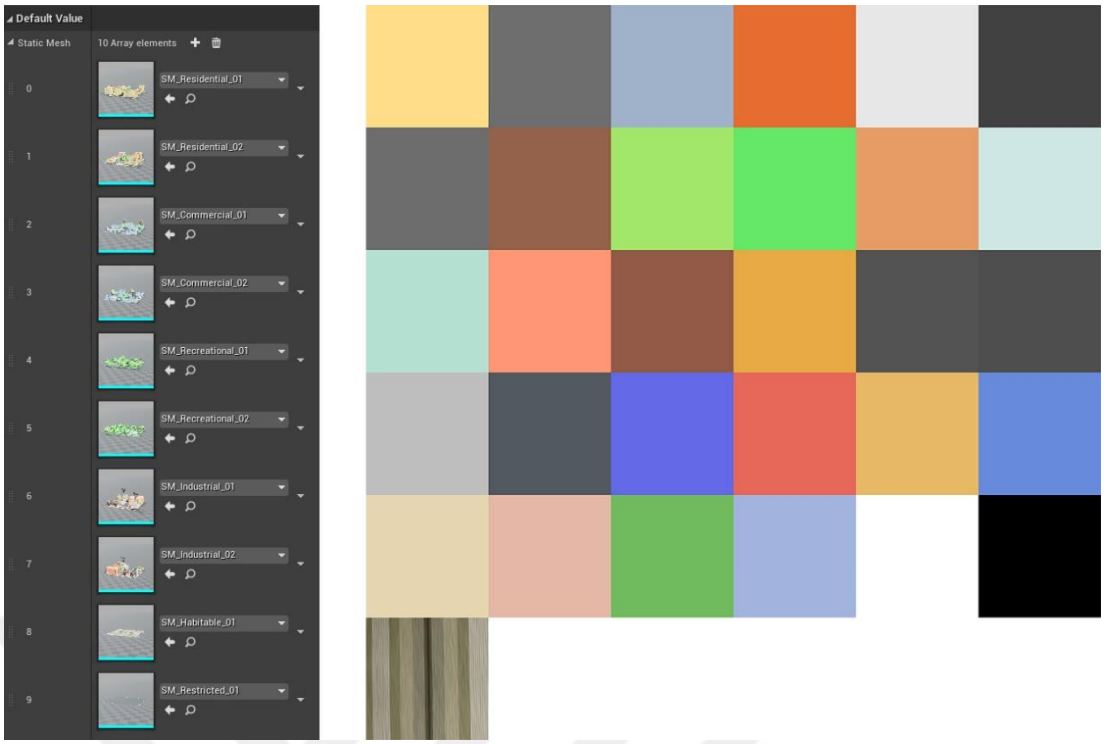


Figure 4.18: Texture atlas and static mesh variables for the unit assets of On the Grid.

The players use a cube token to make their move. The cube token has a total of 6 faces where 4 of them augment different unit types (Figure 4.19).

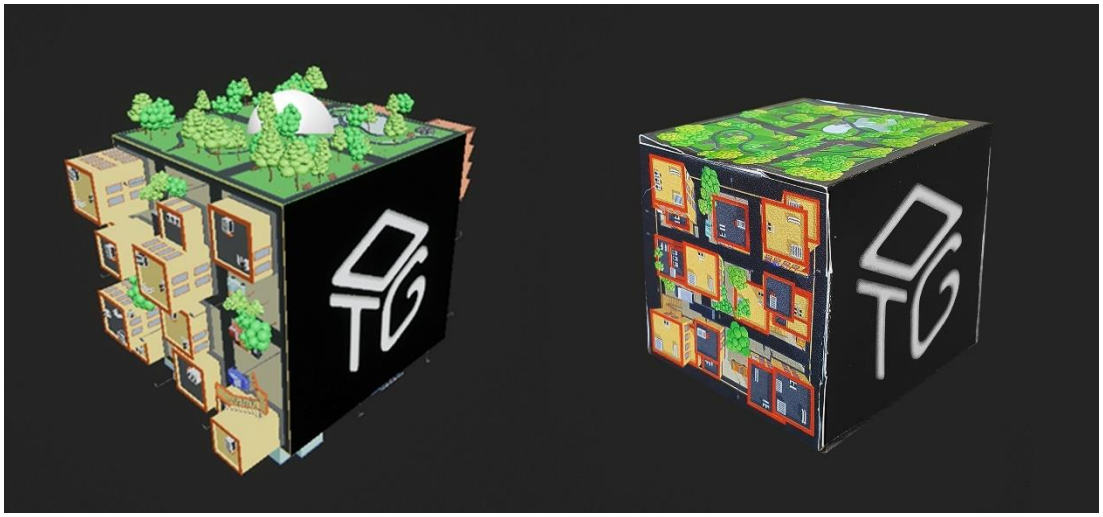


Figure 4.19: Cube Token augmenting different units on different sides.

The player puts the cube on the board and keep it in the camera angle so that the event tick on BP_ARPawn detects the cube as a target image. On the boolean check of the target image for getting board image target, these cube tokens return false because they are not matching with board image target reference variable so another set of nodes is

executed from the false branch of the boolean. There different kinds of variables in the Unreal Engine. Variables hold values like booleans, integers, floats, actor references, object references. The names of the augmented images are also saved as variables and the new execution path has a specific node called “Switch” which makes different actions according to the input variable. The Switch node determines which side of the cube is traced from the camera (Figure 4.20).

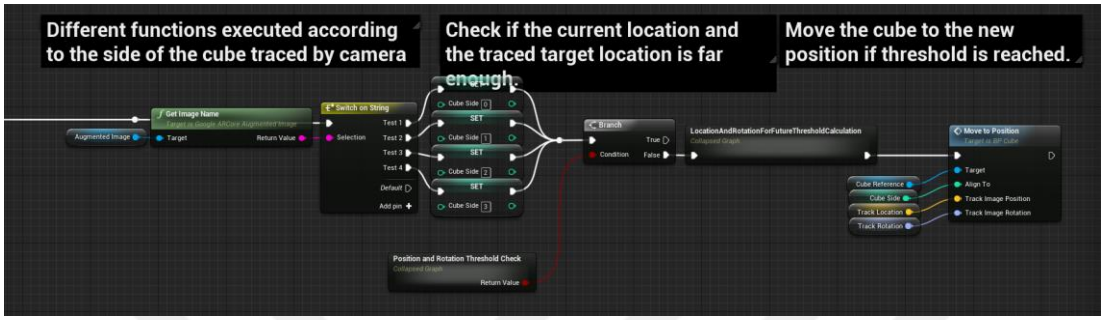


Figure 4.20: Transforming cube token by using the augmented images and calculating the distance from the previous location.

The augmented cube reference is represented as an actor in the game engine, named BP_Cube. How the player holds the cube on the camera angle changes the augmented cube's rotation and position. The side of the cube on top is set to the variable “CubeSide”. The correct rotation and location are get from the arrays of cube’s locations and rotation ith CubeSide variable. To prevent jittering and also to remove unnecessary actions on event tick that reduces framerate, a threshold between the new transformation of the cube and old transform of the cube is set before making any change. If the distance or rotation between the tracked and current targets is high enough, then the reposition function for the cube is executed with interpolation between transforms.

Artificial intelligence in the game uses cellular automata like rules for iterating the urban development pattern. Neighboring cells evaluates and determines the states for each cell. Every unit gathers points from its neighbors and trending types of units are determined according to gathered points. On every move of AI, a function is executed where it gathers arrays of points from every neighbor cell. Every cell has a point table for its trending state (Table 4.1). The highest sum of the points on the array defines the new trending state.

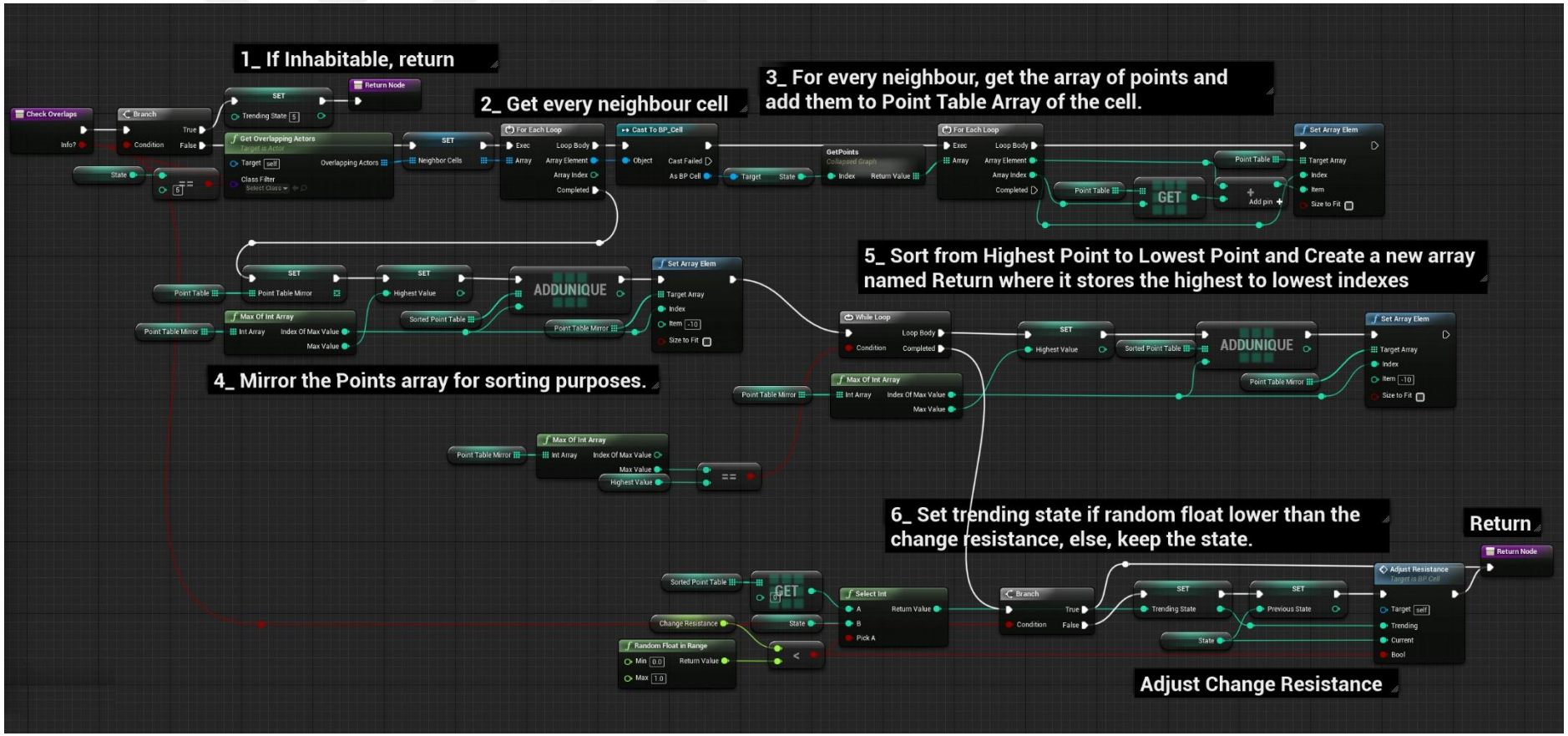


Figure 4.21: Check Overlaps Function

Table 4.1: The number of points provided by each neighbor unit types to the center cell.

Neighbor State / Points	Residential Points	Commercial Points	Recreational Points	Industrial Points
Residential	2-3	3	1-3	-3
Commercial	1-3	2-3	-3	3
Recreational	3	-3	2-3	1-3
Industrial	-3	1-3	3	2-3
Habitable	0	0	0	0
Inhabitable	0	0	0	0

The function that checks trending states is called Check Overlaps (Figure 4.21). This function runs on individual BP_Cell actors. First, it checks if the state of the cell is Inhabitable. Inhabitable cells have no chance to change states so this branch helps preventing unnecessary calculations and keep the application stable. The number of calculations and draw calls increases exponentially with the grid size. The number of calculations in each iteration is predefined as there are 3 presets for grid sizes. If the target cell's state is something other than inhabitable, then the neighbor cells are called by "get overlapping actors" node. This node returns actors of a specific class that overlaps with the target actor as an array. In this case, the target cell is set as self and overlapping actor's class is BP_Cells so this node returns every other cell actor that overlaps with the cell that is calling this function, in other words, its neighbors in an array.

After getting and storing all the neighbor cells in an array variable, for loop executes one by one for each neighbor cells and gets their states and adds points accordingly to the target cell's points table array (Figure 4.22).

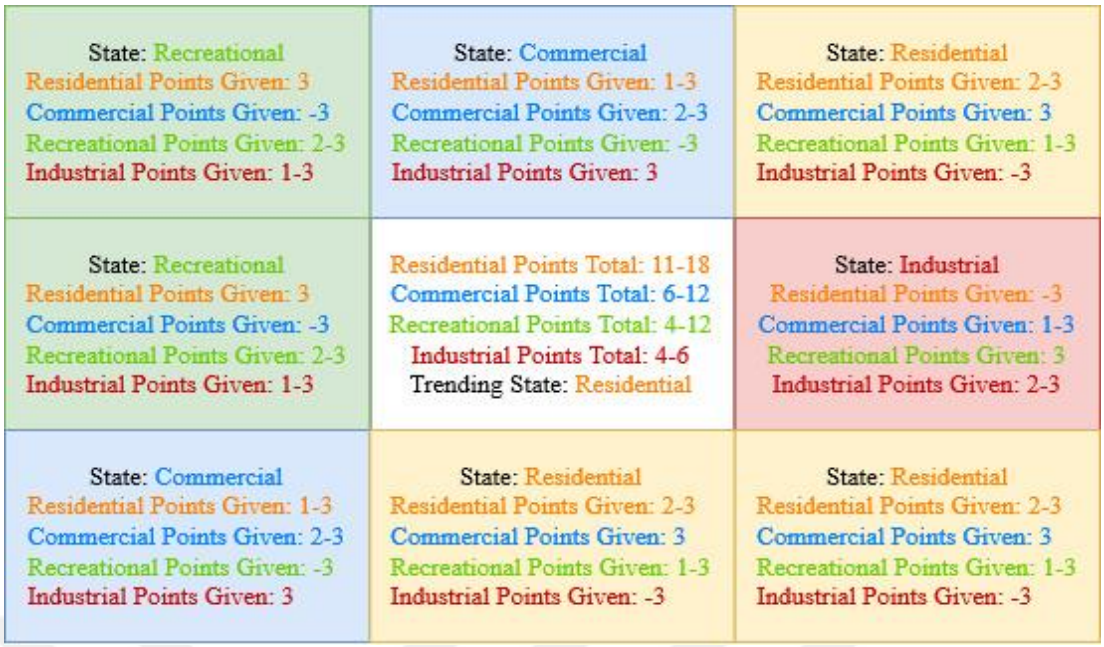


Figure 4.22: An example of a total point calculation for the center cell. Every neighbor cell gives the points array according to their states.

The trending state has the most points in the array. Before getting and comparing the points in the table array, it's required to sort the array from the highest point to lowest. A new array variable is used for sorting, along with a copy mirror of the original array. The highest value of the Point Table Mirror array added into Sorted Point Table array and removed from Point Table Mirror array. This is repeated until there is no more element in Point Table Mirror array so the array is sorted from high to low in Sorted Point Table. Up until this point, only calculations are made in Check Overlaps function and no change in states of cells has been made. An if condition checks if this function is called to change the cell. On true value, it returns only calculating the trending state but on false, it continues to change the cell type according to the change resistance of the actor. A random float in 0-1 range is generated by the engine and a "select" node is used to determine the new state. This node returns one of the two inputs according to the boolean. This is where change resistance becomes an important factor by providing a chance to keep the current state. If the change resistance is lower than the random float then the trending state becomes the new state, else, the cell preserves its state. After changing the cell's state then an adjustment in cell's change resistance is made. Just like the real world, if a section of the neighborhood is older, it takes more effort to change it. That's why cells gain additional change resistance for each the turn they don't convert. If the state of the cell is changed then the change resistance is reset to the initial value (Figure 4.23).

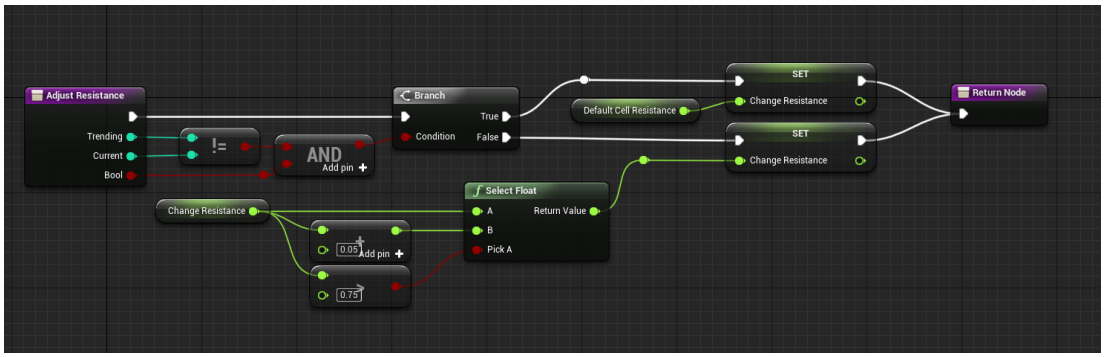


Figure 4.23: Adjusting resistance by increasing it if the state of the cell stays the same.

User Interface elements in Unreal Engine are collected under Unreal Motion Graphics UI Designer. Visual user interface elements such as menus, in-game Heads-up displays, and graphic-based user interface elements are designed in this UI Designer. The widgets are special types of blueprints to create interfaces. It contains two sections: designer and graph. The designer part of the widgets is focused on what the user sees on their screen when it's added in the viewport. The graph part handles all the functions, events, macros, timelines, variables, and more with classic blueprint structures containing special functions. The interaction with players in On the Grid is very dependent on user interface and tangible elements like tokens. Therefore, the game uses several widgets to provide functionality and interactivity. During gameplay, players can touch on individual units to get information about them. An info widget spawns on the world tracking the rotation and location of the specified unit (Figure 4.24).

Figure 4.24: Information widget spawned on the highlighted unit during gameplay.

The widget provides information about the unit type, position on the grid, neighbor cells' states, trending state, and resistance to change. The other widget in the gameplay screen is In-game widget contains Save State button and unit types panel. Save State button executes a function on clicked and this saves date, time, amount of unit of each type, and also takes a screenshot by executing the console command "HighResShot 1". Between each turn, a widget named WBP_Tips shows the turn number and which player's turn it is. On the final turn, WBP_Tips spawns the End Game widget. End Game widget gets the number of cells with the type that the goal defines and checks if the player meets the requirements to win the game.

4.4 Playtesting

The case study of this study contains 4 sections: the game, playtesting, survey, and findings. Previous sections explained the game in detail with its algorithms and structure. For the playtesting, two different sessions are organized. In each session, three players participated in the gameplay. The players are from different backgrounds of architecture, landscape architecture, graphic design, cognitive science. Every player holds a master's degree in their fields or candidates to hold one.

Each gameplay session is limited to thirty minutes and brief information about the game is given to the players. These pieces of information are basic rules of the gameplay process and the survey is included in the time limit. The game rules of the gameplay sessions are also pre-determined (Table 4.2).

Table 4.2: Rulesets for each gameplay session.

Rule / Session	Session 1	Session 2
Level	Existing Scenario	Random Scenario
Goal	Reach 50 Residential Units	Reach 50 Commercial Units
Players	5	3
Grid Size	12 x 12	12 x 12
Cell Resistance	30%	40%
Time Limit	5 Turns	5 Turns

The sound and HUD are enabled from the settings. The existing scenario of the organized industrial site is used for the first session. In the second session, a random scenario is generated. Both gameplay sessions have three players. The winning condition for the first session is to reach the maximum amount of residential units and for the second session, it's to reach maximum commercial units. The grid size is set as 12 x 12, a total of 144 units and players compete to reach 50 residential or commercial units at the end of their gameplay session. The unit's resistance to change is 30% and 40% respectively. The time limit is set to 5 turns. For every turn, every 3 players and artificial intelligence make their moves.

After the gameplay session, the players see if they satisfy the goal. The sessions are limited to 10 minutes of gameplay and 10 minutes of surveying (Appendix B), giving every player approximately 60 seconds to end their turns. The participants will be playtesting a working version of the software before the final release. The measurement metrics of the playtesting sessions will be in two main topics of quantitative metrics, and qualitative data. Quantitative metrics include the task and time performance of the players. Task performance is the ability of the participants to complete the task such as making their move in their turn and ending their turn or ability to use the tokens. On the other hand, time performance is the amount of time the participants spend on the tasks. The playtesting process and handheld device's screen were recorded and the measurements of the quantitative data were gathered later by examining the video recordings of the gameplay sessions. Qualitative data consists of the impressions and experiences of the participants. Most of the qualitative data were gathered later in the survey section of the playtest.

The survey questions are focused on topics like the playability, usability and, replayability of the game in the participatory environment created (Appendix B). The survey results show that 40% of the players never used augmented reality-based applications before. Augmented reality has its requirements from players while using so these players' experiences with the gameplay were harder than the others. In a 1 to 5 scale table, 1 being a highly negative experience and 5 being a highly positive experience, the players have given the least average score of 3.2 to questions about the user experience with the cube token and the guidance of the game to the players during gameplay (Table 4.3).

Table 4.3: Average score table of user experiences during playtesting.

Questions	I had no trouble using the handheld device.	I had no trouble using the cube token.	The game directed me during gameplay.	The game provided useful in-game information during gameplay	I had no trouble understanding game rules.	I had no trouble interacting with other players and making strategic decisions	I consider playing the game again to achieve better a result
Average Score	3.4	3.2	3.2	4	4	4.4	4.4
1= Strongly Disagree 5= Strongly Agree							

The other lower average score is the user experiences with the handheld device. The average score to the evaluation about troubles using the handheld device is 3.4, showing that at least one player had troubles in using the device and playing the game. The playtest participants mentioned that they want to play the game again and they are overall satisfied (Table 4.4).

Table 4.4: Average satisfaction scores of the interactions.

Questions	Interaction with the Handheld Device	Interaction with the Game Tokens	Interaction with other Players	Overall experience
Average Score	3.4	4	4	3.8
1 = Very dissatisfied 5 = Very satisfied				

5. CONCLUSION

This thesis focuses on making a serious city building game that runs with cellular automata algorithms and creating different interaction possibilities in a participatory environment for the players. The main goal was to create game software using augmented reality technologies along with tangible board game elements to offer this kind of workplace. Towards this aim, games with urban design issues, game and serious game concepts, game elements, and generative design techniques in urban growth models were discussed. These topics created the foundation for the designed game, On the Grid, and provided useful information towards creating a city building game. The Cellular Automata algorithms used in this thesis created a strong background within the game with its nature of being unpredictable but at the same time rule-based. The players had the opportunity to face different challenges each time the gameplay started and divergent scenarios happened. At the same time, the literature review on how computational design techniques, especially cellular automata, used in previous urban growth model studies along with geographical information system, pave the way for using real-life scenarios within the game. For this purpose, a real-world scenario also implemented in the game by gathering the data from GIS and converting them into gameplay elements in a few steps. By generating a rule-based noise in the grids in every iteration, computational design techniques in a game have proven itself very a strong alternative method to hard-coded algorithms. The cellular automata running to simulate urban change had no problems during gameplay sessions and created the desired effects.

At this stage, the serious game designed for this thesis can be thought of as an explorative effort to create different interaction sections for participants playing an urban development scenario. The current state of augmented reality technologies, capabilities, and limitations are important factors that limit the current gaming experience and creates concerns and troubles while playing the game. The demanding computational operations of cellular automata, rendering workload on the graphical processing unit, running gameplay algorithms on the background and having a camera recording the world while tracing the geometries in realtime requires the use of AR-capable high-end handheld devices supporting GoogleAR frameworks. Even with a high-end handheld device, all these computations happening at the same time lag the

software from time to time and cause distractions or discomfort to the players during gameplay. Augmented Reality is a very powerful tool to use in software but in some of the tests during development; the lack of light, the lack of object detection, the difficulty in creating well-trackable but aesthetic game elements, the material attributes of the tracked objects such as objects' surfaces being too shiny or low contrast, human errors, or rapid device movements during gameplay have caused game-breaking issues with tracking or augmenting.

The playtest sessions of the designed game have provided useful information about the game. The game at this stage can be concerned as an untested finished product before release, far away from being a robust software, so a more detailed bug fixing, user experience improvements, placeholder removals, professional playtesting haven't been done. It helped to analyze the strong sides of the game and parts that should be more focused and developed. The evaluation process was made in two parts and the thoughts and personal experiences of the players were also taken into consideration. The users' previous experience with augmented reality-based software affected their reactions and behaviors during the gameplay. Inexperienced users happened to act in a way that the augmentation process halted more during their turns. Even though the game offers complex interactions using a handheld device and game tokens, most of the users completed their turns without any extra support.

For future use, this game offers great potential in being a tool that bridges different parties in more complex real-world examples with many variables and constraints and participants. The development of augmented reality tools and technologies can also benefit the final product in the future by improving user experience. For reducing the hardware dependency, using multiple handheld devices with a cloud connection can remove most of the computational load from the handheld device and improve stability. The gameplay exercises can be riched with more tokens, a more detailed city structure representation, and a more detailed cellular automata simulation to fortify it for the uses in real-world scenarios.

REFERENCES

Al-Ahmadi, K., See, L., Heppenstall, A., & Hogg, J. (2009). Calibration of a fuzzy cellular automata model of urban dynamics in Saudi Arabia. *Ecological Complexity*, 6(2), 80–101.

Al-Kheder, S., Wang, J., & Shan, J. (2007). Cellular automata urban growth model calibration with genetic algorithms. *2007 Urban Remote Sensing Joint Event*, Paris, 1-5.

Arai, T., & Akiyama, T. (2004). Empirical analysis for estimating land use transition potential functions - case in the Tokyo metropolitan region. *Computers, Environment and Urban Systems*, 28(1–2), 65–84.

Arneson, E. (2019). How to Play the Classic Game Chutes and Ladders. [online] *The Spruce Crafts*. Retrieved February 24, 2020, from https://www.thesprucecrafts.com/chutes-and-ladders-snakes-and-ladders-411609

Balling, R. J., Taber, J. T., Brown, M. R., & Day, K. (1999). Multiobjective urban planning using genetic algorithm. *Journal of Urban Planning and Development*, 125(2), 86–99.

Barredo, J. I., Demicheli, L., Lavalle, C., Kasanko, M., & McCormick, N. (2004). Modelling future urban scenarios in developing countries: An application case study in Lagos, Nigeria. *Environment and Planning B: Planning and Design*, 31(1), 65–84.

Barredo, J. I., Kasanko, M., McCormick, N., & Lavalle, C. (2003). Modelling dynamic spatial processes: Simulation of urban future scenarios through cellular automata. *Landscape and Urban Planning*, 64(3), 145–160.

Batty, M., Xie, Y., & Sun, Z. (1999). Modeling urban dynamics through GIS-based cellular automata. Computers, *Environment and Urban Systems*, 23(3), 205–233.

Beirão, J., & Duarte, J. (2005). Urban Grammars: Towards Flexible Urban Design. *In Digital Design: The Quest for New Paradigms 23nd eCAADe Conference Proceedings*, 491–500.

Bekker, T., Sturm, J., & Eggen, B. (2010). Designing playful interactions for social interaction and physical play. *Personal and Ubiquitous Computing*, 14(5), 385–396.

Blecic, I., Cecchini, A., & Trunfio, G. A. (2013). Cellular automata simulation of urban dynamics through GPGPU. *Journal of Supercomputing*, 65(2), 614–629.

Brown, F. E., & Johnson, J. H. (1985). An interactive computer model of urban development: the rules governing the morphology of medieval London. *Environment & Planning B: Planning & Design*, *12*(4), 377–400.

Cao, K., Huang, B., Wang, S., & Lin, H. (2012). Sustainable land use optimization using Boundary-based Fast Genetic Algorithm. Computers, *Environment and Urban Systems*, 36(3), 257–269

Chapin, F. S., & Weiss, S. F. (1968). A probabilistic model for residential growth. *Transportation Research*, 2(4), 375–390.

Clarke, K. C., & Gaydos, L. J. (1998). Loose-coupling a cellular automaton model and GIS: long-term urban growth prediction for San Francisco and Washington Baltimore. *International Journal of Geographical Information Science*, 12(7), 699–714.

Couclelis, H. (1985). Cellular worlds: a framework for modeling micro-macro dynamics. *Environment & Planning A*, 17(5), 585–596.

Chan, C.-W., & Chiu, M.-L. (2000). A Simulation Study of Urban Growth Patterns With Fractal Geometry. *Proceedings of the Fifth Conference on Computer Aided Architectural Design Research in Asia*, (1), 55–64. Retrieved from http://papers.cumincad.org/data/works/att/30f5.content.pdf

Deadman, P., Brown, R. D., & Gimblett, H. R. (1993). Modelling Rural Residential Settlement Patterns with Cellular Automata. *Journal of Environmental Management*, *37*(2), 147–160.

Duarte, J. P., Ducla-Soares, G., Caldas, L. G., & Rocha, J. (2006). An urban grammar for the Medina of Marrakech. *In Design Computing and Cognition 06*, 483–502.

Duarte, J. P., & Rocha, J. (2001). A Grammar for the Patio Houses of the Medina of Marrakech. Towards a Tool for Housing Design in Islamic Contexts. In *Communicating Space(s), 24th eCAADe Conference Proceedings* (pp. 860–866). Volos (Greece). Retrieved from http://papers.cumincad.org/data/works/att/2006_860.content.pdf

Feng, C. M., & Lin, J. J. (1999). Using a genetic algorithm to generate alternative sketch maps for urban planning. *Computers, Environment and Urban Systems*, 23(2), 91–108.

Feng, Y., Liu, Y., Tong, X., Liu, M., & Deng, S. (2011). Modeling dynamic urban growth using cellular automata and particle swarm optimization rules. *Landscape and Urban Planning*, 102(3), 188–196.

Flemming, U. (1987). The Role of Shape Grammars in the Analysis and Creation of Designs. In *Compatibility of Design*, Y. Kalay (ed.) New York: John Wiley & Sons, 1986. (pp.245–272).

Foroutan, E., & Reza, M. (2012). Urban Growth Modeling Using Genetic Algorithms and Cellular Automata ; a Case Study of Isfahan Metropolitan Area, Iran. *GIS Ostrava 2012*.

Fullerton T., Swain C., & Hoffman S. (2004). *Game design workshop: A playcentric Approach to Creating Innovative Games*. CMP Books, San Francisco

Gardner M., Mathematical games: The fantastic combinations of John Conway's new solitaire game "life", *Scientific American*, 223(4), 120-123., 1970.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Boston: Addison-Wesley.

Hart, M. (2019). Charting the Top Selling Video Games for the Past 30 Years. Retrieved March 10, 2020, from https://nerdist.com/article/top-selling-video-game-all-time/

Holland, J. H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. Ann Arbor University of Michigan Press 1975 (pp. viii, 183 p.). University of Michigan Press.

Huizinga, J. (2014). *Homo Ludens: A Study of the Play-Element in Culture*. Angelico Press. Kettering, Ohio, USA.

Jan, H., Kunze, A., & Schmitt, G. (2008). Using shape grammars for master planning. In *Design Computing and Cognition '08* - Proceedings of the 3rd International Conference on Design Computing and Cognition (pp. 655–673).

Jenerette, G. D., & Wu, J. (2001). Analysis and simulation of land-use change in the central Arizona - Phoenix region, USA. *Landscape Ecology*, *16*(7), 611–626.

Jianquan, C. (2003). *Modelling spatial and temporal urban growth*. Utrecht; Enschede: Utrecht University; ITC.

Khalilnia, M. H., Ghaemirad, T., & Abbaspour, R. A. (2013). Modeling of urban growth using cellular automata (CA) optimized by Particle Swarm Optimization (PSO). ISPRS - *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-1/W3, 231–234.

Kocabas, V., & Dragicevic, S. (2006). Assessing cellular automata model behavior using a sensitivity analysis approach. *Computers, Environment and Urban Systems*, *30*(6), 921–953.

Lau, K. H., & Kam, B. H. (2005). A cellular automata model for urban land-use simulation. *Environment and Planning B: Planning and Design*, *32*(2), 247–263.

Lee, Y. H., Heeter, C., Magerko, B., & Medler, B. (2012). Gaming mindsets: Implicit theories in serious game learning. *Cyberpsychology, Behavior, and Social Networking*, 15(4), 190–194.

Liu, Y., Tang, W., He, J., Liu, Y., Ai, T., & Liu, D. (2015). A land-use spatial optimization model based on genetic optimization and game theory. *Computers, Environment and Urban Systems*, *49*, 1–14.

Magnussen, R., & Elming, A. L. (2017). Student re-design of deprived neighborhoods in Minecraft: Community-driven urban development. In *Computer-Supported Collaborative Learning Conference*, CSCL (Vol. 1, pp. 271–278). International Society of the Learning Sciences (ISLS).

McDaniel, T. (2018). Block by block: The use of the video game "Minecraft" as a tool to increase public participation. Masters of Public Administration, Texas State University, San Marcos, TX.

Ménard, A., & Marceau, D. J. (2005). Exploration of spatial scale sensitivity in geographic cellular automata. Environment and Planning B: Planning and Design. Pion Limited.

Nacke, L. (September 12, 2014). The formal systems of games and game design atoms. The Acagamic. Retrieved May 4, 2020, from http://acagamic.com/game-design-course/the-formal-systems-of-games-and-game-design-atoms/

Naghibi, F., Delavar, M. R., & Pijanowski, B. (2016). Urban growth modeling using cellular automata with multi-temporal remote sensing images calibrated by the artificial bee colony optimization algorithm. *Sensors* (Switzerland), 16(12).

Nebel, S., Schneider, S., & Rey, G. D. (2016). Mining learning and crafting scientific experiments: A literature review on the use of Minecraft in education and research. *Educational Technology and Society*, 19(2), 355–366.

Nutt, C. (2016). Did you know Stockholm used Cities: Skylines for urban planning? Retrieved March 9, 2020, from https://www.gamasutra.com/view/news/267926/Did_you_know_Stockholm_used_Cities_Skylines_for_urban_planning.php

Packard, N.H., Wolfram, S. (1985). Two-dimensional cellular automata. *Journal of Statistical Physics*, 38, 901–946.

Ritchie, H., & Roser, M. (2020). Urbanization. *OurWorldInData.org*. Retrieved March 9, 2020, from https://ourworldindata.org/urbanization

Scholten, H., Fruijtier, S., Dias, E., Hettinga, S., Opmeer, M., van Leeuwen, W. S., Fruijtier, C. (2017). Geocraft as a means to support the development of smart cities, getting the people of the place involved - youth included -. Quality Innovation Prosperity, 21(1), 119–150.

Shafizadeh Moghadam, H., & Helbich, M. (2013). Spatiotemporal urbanization processes in the megacity of Mumbai, India: A Markov chains-cellular automata urban growth model. Applied Geography, 40, 140–149.

Singh, V., & Gu, N. (2012). Towards an integrated generative design framework. Design Studies, 33(2), 185–207.

Stewart, T. J., & Janssen, R. (2014). A multiobjective GIS-based land use planning algorithm. Computers, Environment and Urban Systems, 46, 25–34.

Stewart, T. J., Janssen, R., & Van Herwijnen, M. (2004). A genetic algorithm approach to multiobjective land use planning. Computers and Operations Research, 31(14), 2293–2313.

Stiny, G. (1980). Introduction to shape and shape grammars. *Environment and Planning B: Planning and Design*, 7(3), 343–351.

Stiny, G., & Gips, J. (1971). Shape Grammars and the Generative Specification of Painting and Sculpture. *IFIP Congress*.

Suits, B., & Hurka, T. (2005). The grasshopper: games, life, and utopia. Peterborough, Ont: Broadview Press.

Tan, R., Liu, Y., Zhou, K., Jiao, L., & Tang, W. (2015). A game-theory based agent-cellular model for use in urban growth simulation: A case study of the rapidly urbanizing Wuhan area of central China. Computers, Environment and Urban Systems, 49, 15–29.

Tobler, W. R. (1970). A Computer Movie Simulating Urban Growth in the Detroit Region. Economic Geography, 46, 234.

Westerberg P. & von Heland, F. (2015). *Using Minecraft for Youth Participation in Urban Design and Governance*, Nairobi: United Nations Human Settlements Programme (UN-HABITAT).

Wahyudi, A., & Liu, Y. (2016). Cellular automata for urban growth modelling: A review on factors defining transition rules. International Review for Spatial Planning and Sustainable Development. SPSD Press.

Url-1 <https://morethanjustmonopoly.wordpress.com/2016/10/08/carcassonne-review-here-come-the-meeples/>, date retrieved 10.03.2020.

Url-2 <https://www.gamewatcher.com/news/2016-23-08-stockholm-city-planners-using-cities-skylines-to-plan-and-develop-real-world-district/>, date retrieved 10.03.2020.

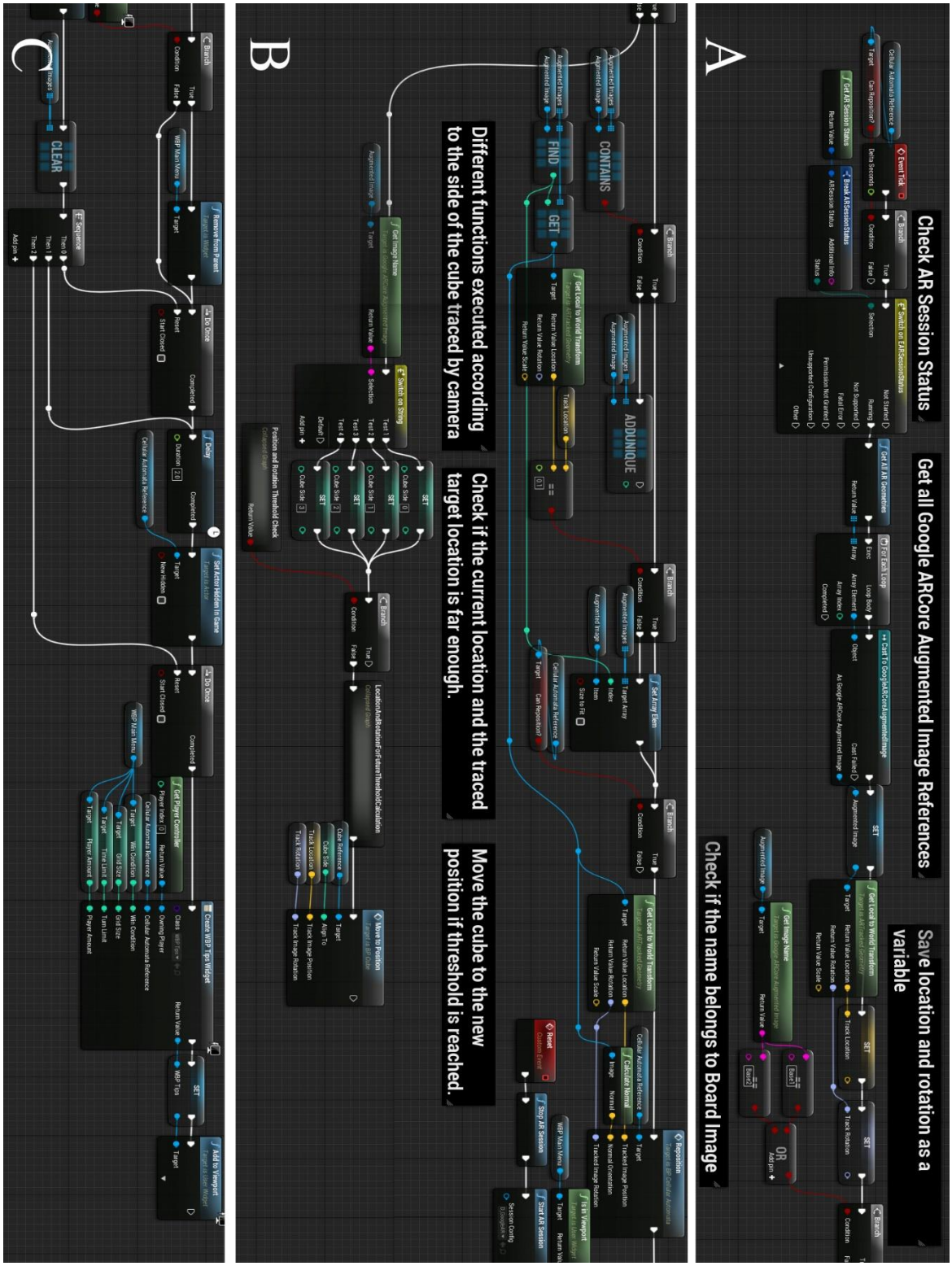
Url-3 <https://www.minecraft.net/en-us/article/a-grand-gallery/>, date retrieved 17.03.2020.

Url-4 <https://www.playthecity.nl/page/8983/play-noord/>, date retrieved 21.03.2020

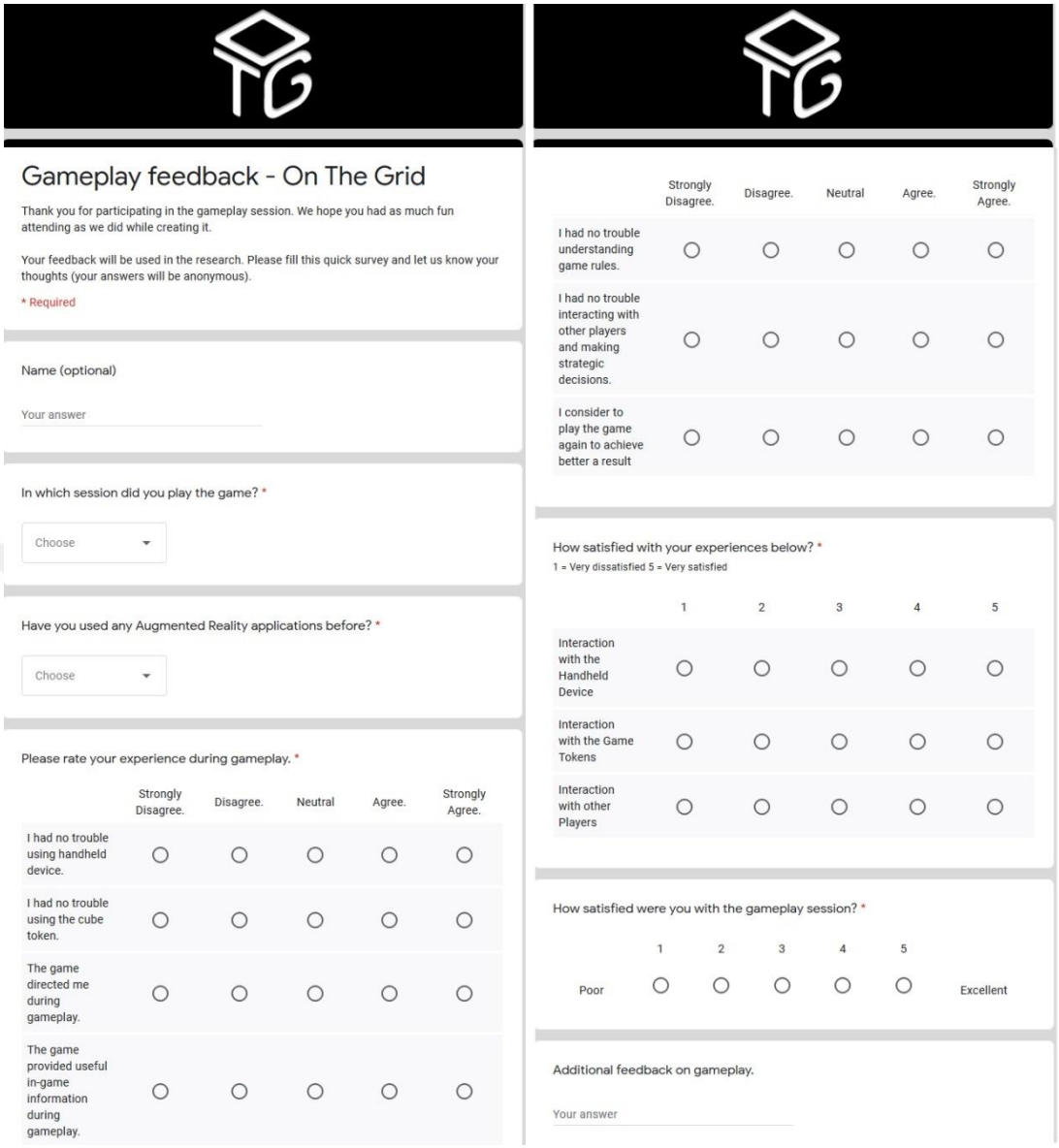
Url-5 <https://www.unrealengine.com/en-US/spotlights/unreal-engine-in-camera-vfx-a-behind-the-scenes-look>, date retrieved 27.03.2020

APPENDICES

APPENDIX A: Visual codes in AR Pawn.



APPENDIX B: Survey completed by the playtesters.



Gameplay feedback - On The Grid

Thank you for participating in the gameplay session. We hope you had as much fun attending as we did while creating it.

Your feedback will be used in the research. Please fill this quick survey and let us know your thoughts (your answers will be anonymous).

* Required

Name (optional)

Your answer

In which session did you play the game? *

Choose

Have you used any Augmented Reality applications before? *

Choose

Please rate your experience during gameplay. *

	Strongly Disagree.	Disagree.	Neutral	Agree.	Strongly Agree.
I had no trouble using handheld device.	○	○	○	○	○
I had no trouble using the cube token.	○	○	○	○	○
The game directed me during gameplay.	○	○	○	○	○
The game provided useful in-game information during gameplay.	○	○	○	○	○
I had no trouble understanding game rules.	○	○	○	○	○
I had no trouble interacting with other players and making strategic decisions.	○	○	○	○	○
I consider to play the game again to achieve better a result	○	○	○	○	○

How satisfied with your experiences below? *

1 = Very dissatisfied 5 = Very satisfied

	1	2	3	4	5
Interaction with the Handheld Device	○	○	○	○	○
Interaction with the Game Tokens	○	○	○	○	○
Interaction with other Players	○	○	○	○	○

How satisfied were you with the gameplay session? *

	1	2	3	4	5	
Poor	○	○	○	○	○	Excellent

Additional feedback on gameplay.

Your answer

CURRICULUM VITAE

Name Surname **: Şeref Atilla Gürbüz**

Place and Date of Birth **: Istanbul / 21.05.1993**

E-Mail **: serefatilla@gmail.com**

EDUCATION **:**

- **B.Sc.** **:** 2017, Istanbul Technical University, Faculty of Architecture, Architecture

PROFESSIONAL EXPERIENCE

- 2014 (2 months) Piramit Architecture, Istanbul, Intern
- 2015 (3 months) Serhan Ceyhan Architects, Istanbul, Intern
- 2016 (2 months) Avrasya Consultancy Yüksel Proje, Istanbul, Construction Intern
- 2017 (6 months) Hive TTO, Istanbul, Architect

REWARDS:

- 2015 Museum of Modern Art, Uneven Growth: Tactical Urbanisms for Expanding Megacities, Project Elected for Online Exhibition
- 2018 Finalist Project for the Global Warming Center Architectural Competition organized by Rethinking Architectural Competitions
- 2019 First Place in Turkey's first education-themed hackathon, Educathon 2019.

PUBLICATIONS:

- Gürbüz, Ş. A., and Yılmaz, F. (2018). Acoustic Panel Design with Computational Design Techniques, *12th Computational Design in Architecture National Symposium*, Turkey.