Republic of Iraq

Ministry of Higher Education and Scientific Research
University of AL-Qadisiya
College of Computer Science and Information Technology
Multimedia department

DESIGN AND IMPLEMENTATION OF FIGHTING GAME

A graduation project is submitted to the Multimedia department in partial fulfillment of the requirements for the degree of Bachelor in Information Technology in Multimedia
By.

Sarah Saad **Abbas Sabah**

SUPERVISOR
Associate lecture
Mohammed rajih

2019

بسم الله الرحمن الرحيم

(قَالَ الَّذِي عِنْدَهُ عِلْمٌ مِنَ الْكِتَابِ أَنَا آتِيكَ بِهِ قَبْلَ أَنْ يَرْتَدَّ إِلَيْكَ طَرْفُكَ فَلَمَّا رَآهُ

مُسْتَقِرًّا عِنْدَهُ قَالَ هَذَا مِنْ فَضْلِ رَبِّي لِيَبْلُوَنِي أَأَشْكُرُ أَمْ أَكْفُرُ وَمَنْ شَكَرَ فَإِنَّمَا

يَشْكُرُ لِنَفْسِهِ وَمَنْ كَفَرَ فَإِنَّ رَبِّي غَنِيٌّ كَرِيمٌ قَالَ نَكِّرُوا لَهَا عَرْشَهَا نَنْظُرْ أَتَهْتَدِي

أَمْ تَكُونُ مِنَ الَّذِينَ لَا يَهْتَدُونَ).

صدق الله العلي العظيم

سورة النمل (40)

الإهداء

الى من تجرع الكأس فارغا ليسقيني قطرة حب

الى من كلت أنامله ليقدم لنا لحظة سعادة

الى من حصد الأشواك عن دربي ليمهد لي طريق العلم

الى القلب الكبير والدي العزيز
الى رمز الحب وبلسم الشفاء

الى القلب الناصع بالبياض والدتي الحبيبه

الى من بها اكبر وعليها اعتمد

الى شمعه متقدة تنير ظلمة حياتي
ورفقاء دربي منذ ان حملنا
الحقائب الصغيرة وسرنا على الدرب خطوة بخطوة

الى اخوتي وأصدقائي

الى من وقف على المنابر وأعطى من حصيلة فكره لينير دربنا اهديكم مشروع تخرجي.

شكر وتقدير

(الى قسم الوسائط المتعددة)

الحمد لله رب العالمين والصلاة والسلام على أشرف الأنبياء والمرسلين النبي محمَّد (صلى الله عليه و سلم).

فإني أشكر الله تعالى على فضله حيث أتاح لي إنجاز هذا العمل بفضله، فله الحمد أولاً وآخراً.

ثم أشكر أولئك الأخيار الذين مدوا لي يدَ المساعدة، خلال هذه الفترة، وفي مقدمتهم أستاذي المشرف على البحث الأستاذ (محمد راجح) الذي لم يدَّخر جهداً في مساعدتي وكان يحثّني على البحث، ويرغِّبني فيه، ويقوّي عزيمتي عليه فله من الله الأجر ومني كل تقدير حفظه الله ومتّعه بالصحة والعافية ونفع بعلومه.

Abstract

The game designers craft is very young if compared to filmmaking and software development. The knowledge base and formal techniques of these areas is far more comprehensive. Even after decades of evolution of the games production software, the range of design centered techniques and tools is still limited, as observed by many authors. Thereby, efforts have been made towards the establishment of game design formal methods. These efforts converge to two approaches: the build of a shared design vocabulary and a game design modeling language. While valuable, the existing implementations of these approaches are not mature enough to gain industry adepts, serving only as reference to future works. Moreover, it is needed to discover the designers particular methods, which may contribute to-wards the constitution of a unified design tool box .

Contents

Chapter Three	Page No.
3.1 To execute our game we have several steps	31
3.2 characters of this game	33
3.2.1 How to implement any MD2 model?	33
3.3 Sound of this game	34
3.4 Fonts of this game	34
3.5 Controls this game	35
3.6 Graphics	36
3.7 Challenges and Rules of this game	36
3.8 Collision detection	38
3.9 The story	39

Chapter Four	Page

Chapter one

1.1 Introduction

Designing games is hard. Although games have been around for a very long time, it was the rise of the computer game industry over the past decades that caused this problem to become prevalent. During its short history the computer game industry has grown from individual developers and small teams towards multi-million dollar projects involving hundreds of employees. In the contemporary game industry there is little room for mistakes: the financial stakes have grown too high.

Today, more than any time before, there is a need for a better understanding of the process of designing a successful game in order to prevent such mistakes; there is a need for better applied theory and intellectual tools to aid game designers in their task.

At the same time, more people are playing video games than ever before. A wider audience means that there is an ever increasing demand for games with quality game play. As game players get more experienced they grow an appetite for ever more sophisticated games. Compared to other forms of art and media, computer games are a fairly recent invention. There is still plenty of room for development and innovation.

Level design has been one way to harness a game's emergent behavior, by restricting the game play to a series of relatively simple tasks loosely strung together by a storyline. However, high-quality content is expensive to produce. Games with many hand-crafted levels are expensive to produce, and fail to exploit the true expressive power of open game worlds that emerge from rule systems.

What are games design?

Game design is the art of applying design and aesthetics to create a game for entertainment or for educational, exercise, or experimental purposes. Increasingly, elements and principles of game design are also applied to other interactions, in the form of gamification.

Game design creates goals, rules and challenges to define a board game, card game, dice game, casino game, role-playing game, sport, video game, war game or simulation that produces desirable interactions among its participants and, possibly, spectators.

Academically, game design is part of game studies, while game theory studies strategic decision making (primarily in non-game situations). Games have historically inspired seminal research in the fields of probability, artificial intelligence, economics, and optimization theory. Applying game design to itself is a current research topic in meta-design.

Chapter two

2.1 Game Concepts

A game concept is an idea for a game, briefly describing its core play
mechanisms, that the players represent, and how they win or lose. The process of design the game begins by collect various game play ideas and concepts. Game concepts have to be putted down to Earth and given shape and definition.

These concepts, when working together will form a game. They will be required to work together forcefully. The professional game design always occurs in parallel with a multitude of constraints, objectives, and considerations.

There are many game concepts, in summary we will states the following:

- Level construction: The process of creating game environments.
- Lighting, texturing, effects, and audio: How we detail our game environments.
- Items, and behaviors: How we set up our game environments
- Camera considerations How we handle camera issues
- . player feedback loops: How we test and refine game titles.[3]

2.2 What is a Game Designer?

Game design is an incredibly broad field. The professional designers
sometimes have trouble explaining what they do. Part of the reason for this is that they do so many things. Here are some analogies we have seen when trying to explain what it is like to be a game designer:

- Game designers are artists. The term “art” is just as difficult to define as the word “game”. However, if games can be a form of art, then designers would be artists.
- Game designers are architects. Architects do not build physical structures; they create blueprints. Video game designers also create “blueprints” which are referred to as “design docs”. Game designers are party hosts. As designers, we invite players into our space and try our best to show them a good time.
- Game designers are research scientists. We create games in a manner that is very close to the scientific method.
- Game designers are gods. We create worlds, and we create the physical rules that govern those worlds.

- Game designers are lawyers. We create a set of rules that others must follow.
- Game designers are educators. In the theory of fun the entertainment and education are strongly linked, and games are (at least sometimes) fun because they involve learning new skills.

If game design is all these things, where would it fit in a college curriculum? It could be justified in the school of education, or art, or architecture, or theology, or recreation management, or law, or engineering, or applied sciences, or other things.

Is a game designer all these things? It is open for discussion, but game design has elements of many other fields, but it is still its own field. In addition, you can see just how broad the field is! As the field of game design advances, we may see a day where game designers are so specialized that "game design" will be like the field of "science", students will need to pick a specialty (Chemistry, Biology, Physics, etc.) rather than just "majoring in Science".[4]

2.3 Elements of games

This depends on who you ask. There are several schemes of classification but none is perfect, but by looking at all of them we can see some emerging themes that can shed light on the kinds of things that we need to create as game designers if we are to make games.

What follows are some parts of games and some of the things designers may consider when looking at these elements:

Players

How many players does the game support? Must it be an exact number (1 player only), or a variable number (1 to 5 players)? Can players enter or leave during play? How does this affect play? What is the relationship between players: are there teams, or individuals? Can teams be uneven? Here are some example player structures; this is by no means a complete list:

- Solitaire (1 player vs. the game system). Examples include the card game Klondike (sometimes just called "Solitaire") and the video game Minesweeper.
- Head-to-head (1 player vs. 1 player). Chess and Go are classic examples.

- "PvE" (multiple players vs. the game system). This is common in MMOs like World of Warcraft. Some purely-cooperative board games exist too, such as Knizia's Lord of the Rings, Arkham Horror, and Pandemic.

- One-against-many (one player vs. multiple players). The board game Scotland Yard is a great example of this; it pits a single player as Mr. X against a team of detectives.

- Free-for-all (1 player vs. 1 player vs. 1-player vs. …). Perhaps the most common player structure for multi-player games, this can be found everywhere, from board games like Monopoly to "multiplayer death match" play in most first person shooter video games.

- Separate individuals against the system (1 player vs. a series of other players).The casino game Blackjack is an example, where the “House” is playing as a single player against several other players, but those other players are not affecting each other much and do not really help or hinder or play against each other.

- Team competition (multiple players vs. multiple players [vs. multiple players...]). This is also a common structure, finding its way into most team sports, card games like Bridge and Spades, team-based online games like “Capture the Flag” modes from first-person shooters, and numerous other games.

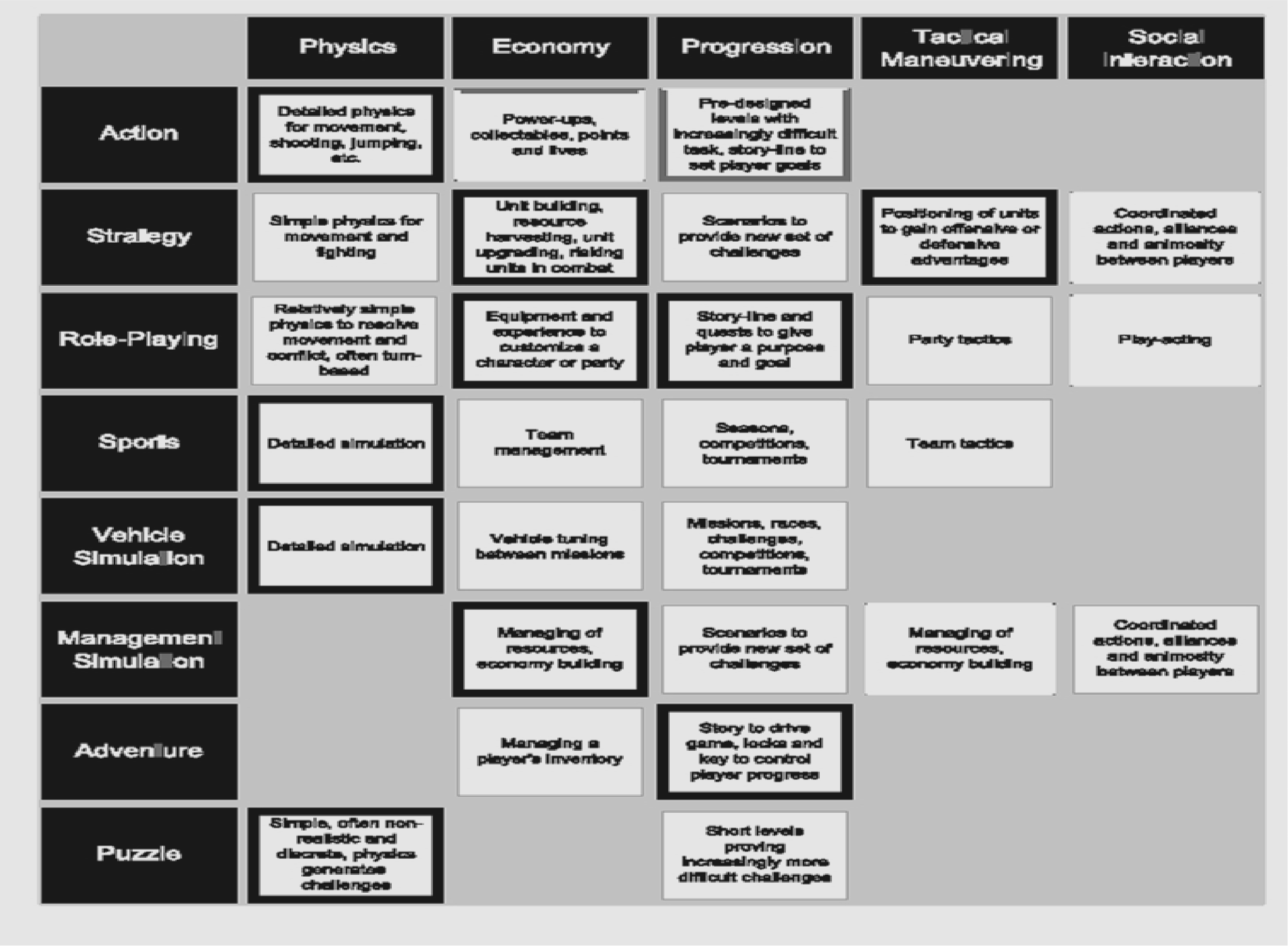
- Predator-Prey. Players form a (real or virtual) circle. Everyone’s goal is to attack the player on their left, and defend themselves from the player on their right .The college game Assassination and the trading-card game Vampire: the EternalStruggle both use this structure.

- Five-pointed Star. first was in a five-player Magic: the Gathering variant .Thegoal is to eliminate both of the players who are not on either side of you [4].

2.4 Game Classification

2.4.1 Rules

What type of rules drives the gameplay of a particular game varies a lot between games and genres. Some games derive their gameplay mostly from their economy, others from physics, level progression, tactical maneuvering or social dynamics. Categorizations of games in different genres by the game industry and game journalists is usually based on the type of gameplay (Veugen, 2011, 42), and thus by extension on the different types of rules that feature more or less prominently in these genres. Figure 1.1 provides an overview of a typical game classification scheme and how these genres and their associated gameplay relate to different types of rule systems. Note, however, that this classification is one of many. There is a serious lack of consensus among the several classification schemes in use. The point here is not to present a definitive genre classification. Rather, it is to indicate how different types of rules correlate to different types of gameplay. There are many more genres and sub-genres that can be derived from this basic classification. For example, first-person shooters are a particular sub-genre of action games, whereas action-adventure games are common hybrids of the action and adventure game genres.



	Physics	Economy	Progression	Tactical Maneuvering	Social Interaction
Action	Detailed physics for movement, shooting, jumping, etc.	Power-ups, collectables, points and lives	Pre-designed levels with increasingly difficult task, story-line to set player goals		
Strategy	Simple physics for movement and fighting	Unit building, resource harvesting, unit upgrading, risking units in combat	Scenarios to provide new set of challenges	Positioning of units to gain offensive or defensive advantages	Coordinated actions, alliances and animosity between players
Role-Playing	Relatively simple physics to resolve movement and conflict, often turn-based	Equipment and experience to customize a character or party	Story-line and quests to give player a purpose and goal	Party tactics	Play-acting
Sports	Detailed simulation	Team management	Seasons, competitions, tournaments	Team tactics	
Vehicle Simulation	Detailed simulation	Vehicle tuning between missions	Missions, races, challenges, competitions, tournaments		
Management Simulation		Managing of resources, economy building	Scenarios to provide new set of challenges	Managing of resources, economy building	Coordinated actions, alliances and animosity between players
Adventure		Managing a player's inventory	Story to drive game, locks and key to control player progress		
Puzzle	Simple, often non-realistic and discrete, physics generates challenges		Short levels proving increasingly more difficult challenges		

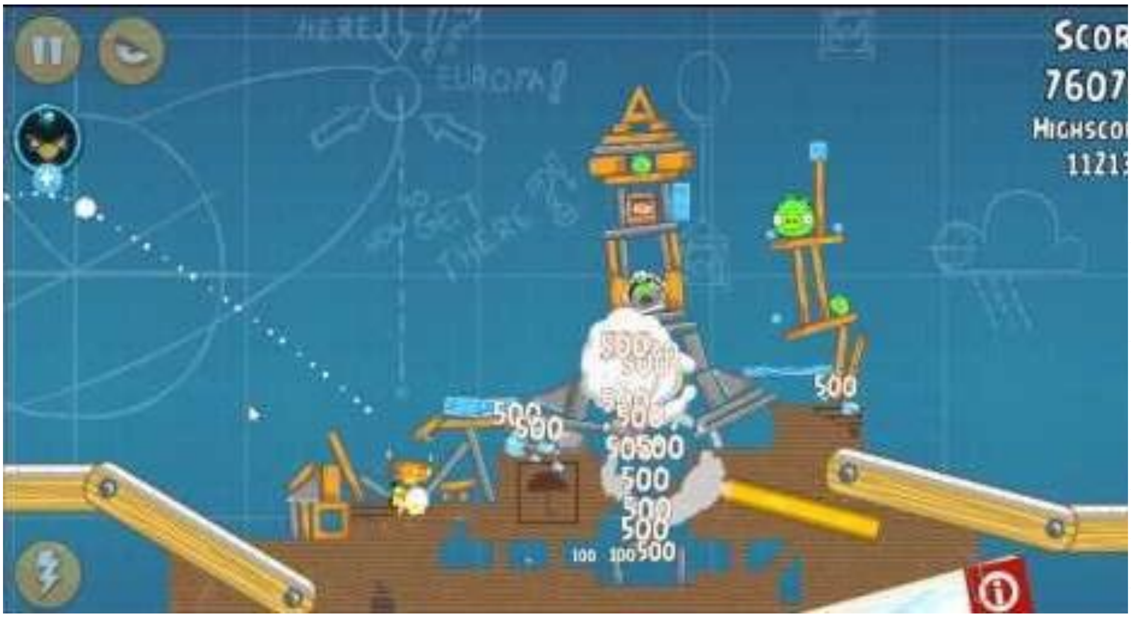
Figure 1.1: Games genres taken from Adams & Rollings (2007) and correlated to five diferent types of game rules or structures. The thickness and darkness of the outlines indicate relative importance of those types of rules for most games in that genre.

2.4.2 Objectives (goals)

- **Well-defined** – for instance, saving a princess or launching a bird at a pig.

- **Measurable** – you receive a score

.



- **Incremental** – you have immediate goals that get you to the next level

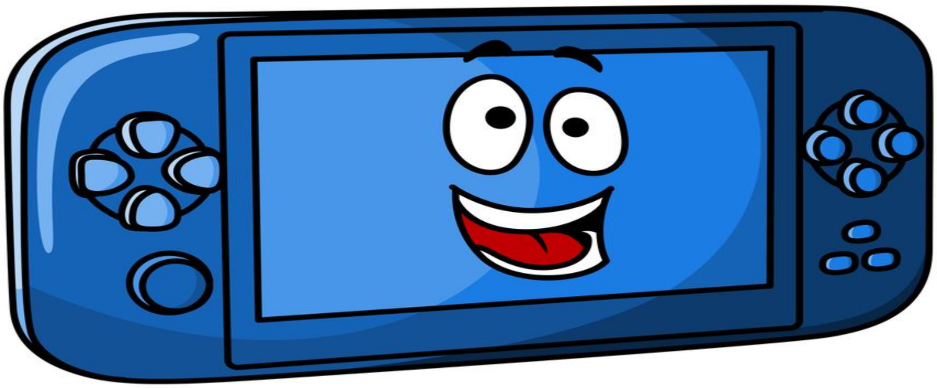
- **Feedback-supported**. You know the results of your actions right away.

- **Within a time frame**. A timer is counting down or a jingle is playing. You move.

- **Attainable.** The level of difficulty increases gradually. You experience success.

- **Not boring.** The level of difficulty increases gradually. You experience challenge.

- Emotionally safe. You can make mistakes.



- **Active**. You practice.

- **Rewarding.** Visual and audio stimuli provide variable surprise rewards and celebrate your success with you.

2.5 Resources and resource management

Resource management is about collecting, monitoring, and leveraging quantitative resources with incomplete information. Any game with complex resource management, that offers compelling choices, will involve imperfect decisions, leading to interesting strategy.

In simple English, this means you get money, ore, pylons, or whatever counter, and this has to be used wisely to compete. You never have enough information to make fully informed decisions, and must develop the best strategy you can with imperfect information. This mechanic is prevalent in games of almost every genre. In video games, you can see this as economies (EVE Online), resource gathering (many RTS, such as Command and Conquer), Auctions (many MMOs such as World of Warcraft), and even in ammo management and health in first person shooters, such as Quake or Doom.

Having a visible, quantitative counter of how much of a given resource is available can cause psychological stress and also satisfaction given different conditions. For example, a health bar is directly tied to a loss state in a game. Watching the health bar deplete will cause stress on the player, perhaps causing them to rethink their decision to run headlong into a room of enemies that they otherwise - at full health - wouldn't have blinked at. This stress state can be useful when trying to slow the pace of gameplay and force a more cerebral strategy. Be careful that this is not applied heavy handed, otherwise it can frustrate a player

2.6 Tasks and challenges facing translation and localization of games

The main types of localization or translation of a game start with linguistic and cultural factors first: the translation of linguistic and cultural references and maintain the user experience of the game and even make it more attractive for other countries.

Then hardware and software: for example changing the functions of keyboards that vary between country and country. Then the legal aspects of the game: Age rankings may vary depending on the country in which the game is played.

Then graphics and music: some games may carry different characters, or the same ones with a slightly different appearance in order to facilitate the identification of players. The music may also vary according to national trends or target market preferences, which often consist mostly of young people.

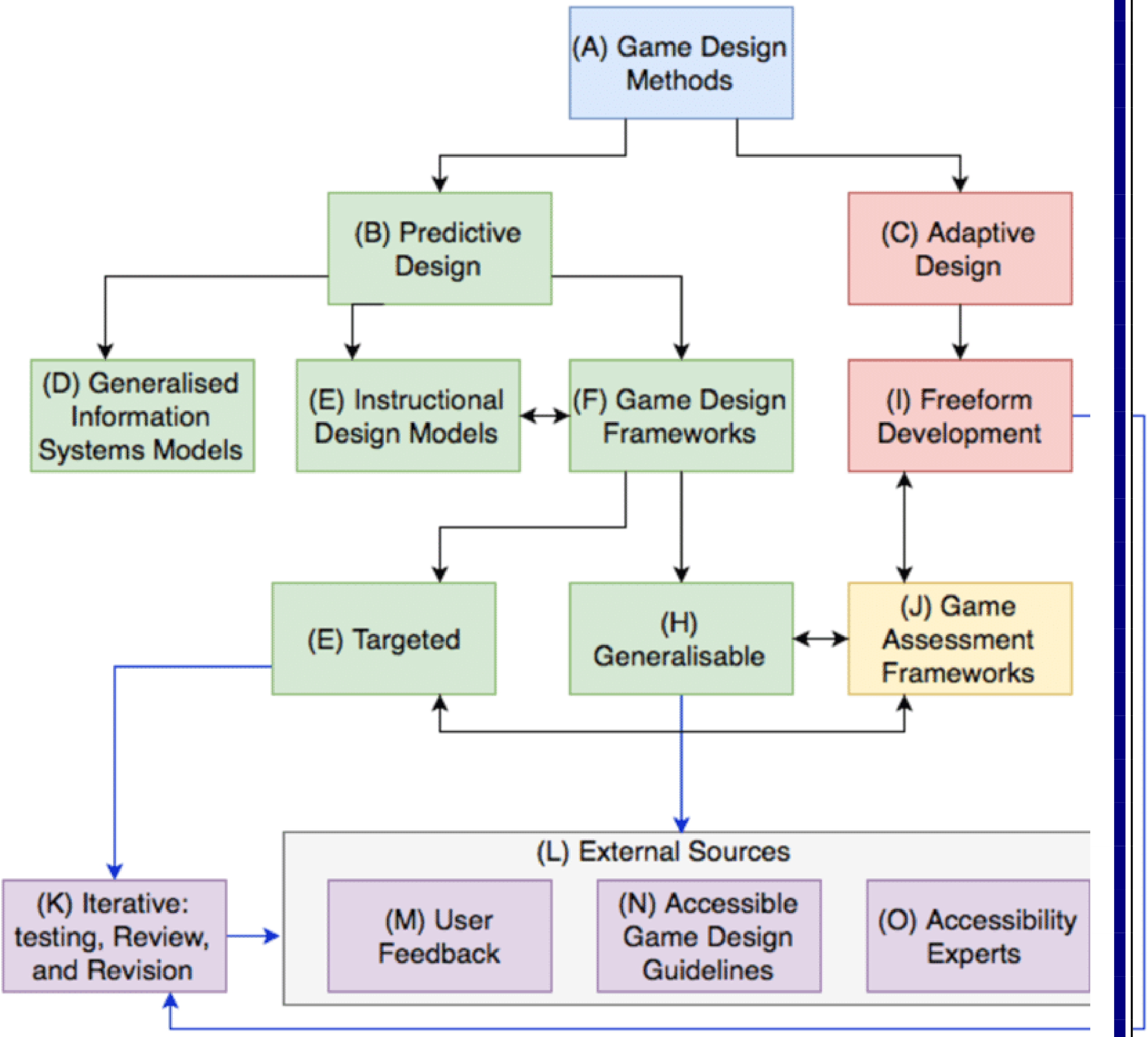
Often the process of localization of a game is expanded to include modification of what may trigger specific cultural sensitive points. Many developers resort to specialized translation companies such as Pro Translate to implement these changes and localize and translate the game in a way that fits the culture of the country that the company intends to distribute the game.

Poor translation and localization of the game means that it is not suitable for the target country which may result in significant problems which may lead to the impossibility of releasing the games from customs sometimes and this of course may cost the game developers heavy losses.

For example, translated games exported to Germany often have very significant and fundamental changes because of the country's strict anti-blood policies, obscenities, symbols associated with racial hatred, and the most sensitive and most sensitive "Nazi symbolism.

2.7 Game Design Methods

Game design method implies an approach to a problem which is likely to lead to a successful solution. It is backed by a certain amount of imagination, creativity and intuition, and a more or less systematic investigation of the problem with the


(A) Game Design Methods
(B) Predictive Design
(C) Adaptive Design
(D) Generalised Information Systems Models
(E) Instructional Design Models
(F) Game Design Frameworks
(I) Freeform Development
(E) Targeted
(H) Generalisable
(J) Game Assessment Frameworks
(L) External Sources
(K) Iterative: testing, Review, and Revision
(M) User Feedback
(N) Accessible Game Design Guidelines
(O) Accessibility Experts

help of techniques, 'best practices', or 'tricks of the trade

2.8 The Aim Of Game Design Methods

Compared with the vast body of operational knowledge found in the world of filmmaking, the game design community is just beginning to articulate the concepts and techniques specific to our medium in order to establish methods of game design. What should we expect of a game design method? To borrow from Doug Church [4], game design methods should:

1. Relate to game design. The method has to be applicable to the actual interaction structure and mechanics of a game, not to concerns related to marketing, production, or management. This restriction is debatable (as it is easily violated), but it does define the scope of this article as well as the roundtables. While methods addressing the development process and its constraints are certainly needed (whether or not their substance is specific to games), they are not considered "design methods".

2. Have utility - it should be a "tool". A method has to be more than just a list of concrete examples or a definition of a building block. A method involves a procedure, a step-by-step recipe, at least parts of which can be applied by simple, even automatic repetition. In particular, it should address specific and concrete issues occurring during the design stage of game development.

3 . Be abstract. A method has to apply to a large, presumably infinite number of game situations or instances. The actual level of abstraction can vary (e.g., genre-specific, or applicable to any interactive medium, etc.) but it has to be at least one step removed from the concrete instance (game or game element).

4. Be formalized. A method needs some degree of formal structure, some amount of specific organization. Typically this

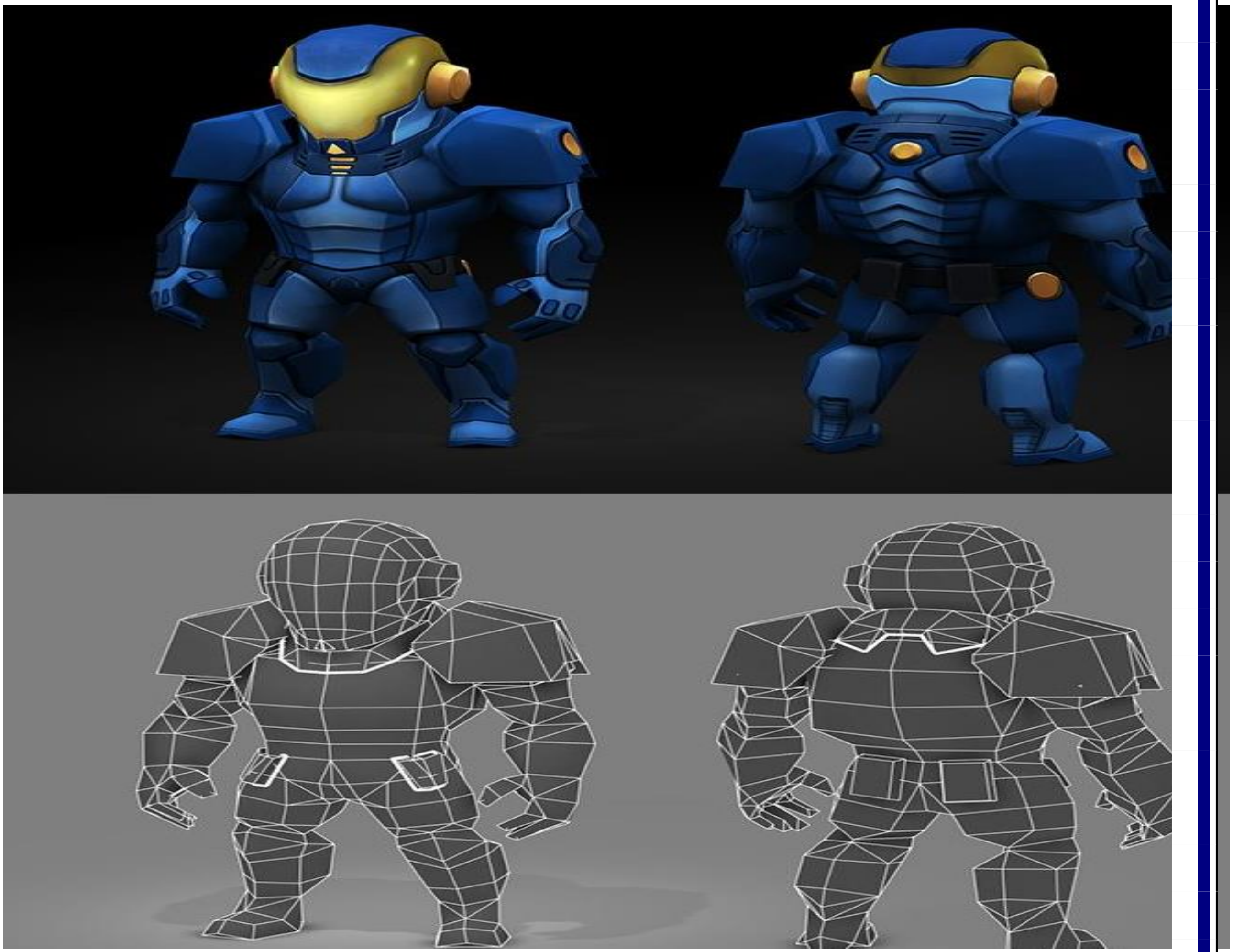
consists of a template structure used repeatedly to contain information. Rollings' use of fictional dialog and anecdote [22] and Rouse's reliance on interview [29] are examples of forms found outside the context of game design

2.9 What is 3d modeling ?

3D modeling is the use of computers to create images and graphics that look to have three dimensions.

Advanced 3D animation software programs like Autodesk Maya and 3ds Max as well as some coding knowledge are required to be able to create your own 3D models.

The basic process involves connecting sets of points with lines, curved surfaces, and other geometric data to make wireframe



models.

2.9.1 Is 3D Modeling for You?

One of the most common traits found in professional 3D modelers is creativity. Much like drawing and animation, 3D modeling requires a lot of imagination and innovation to create characters and worlds that will stand out from the rest. This includes having a sharp attention to detail so your work is always polished and of the highest quality possible.

3D modelers also usually need to have some familiarity with computer oding, and sometimes scripting languages as well. 3D software programs aren't the easiest to learn, and 3D modeling itself can have a sharp learning curve, which means having a talent for using computers to perform tasks propriately is a must.

Last but not least, people who thrive and enjoy working in a team setting will enjoy 3D modeling. Whether you're in a top game studio like Blizzard Entertainment or working on Pixar's next masterpiece, you can expect to work closely with artists, animators, and several other members on projects. Since the video game industry in particular can be very demanding, people who don't do well with tight schedules and long hours probably shouldn't become a 3D modeler to make video games

2.9.2 How to Learn 3D Modeling ?

While there are people out there doing 3D modeling for a living without ever going to school, most break into the industries after earning a degree from a reputable computer animation school or college program.

In fact, it’s getting harder and harder breaking into the video game and film industry as a 3D animator without a college degree or proof that you’ve gone through a good program

This is because a degree is proof to employers that you’ve received hands-on training with 3D programs while being trained by people capable of educating others Any good degree or program will teach you the fundamental principles of 3D modeling, leaving you with a solid foundation in what you need to stand out from the crowd and get hired. Graduates also usually end up with a good demo reel to show potential employers their work.

3D Modeling Careers Today, 3D modeling is used in a vast number of fields. The medical industry uses detailed 3D models of organs, including 2-D image slices from a CT or MRI scan. Architects and engineers also make use of 3D software programs to show proposed buildings, landscapes, devices, structures, vehicles, and more.

Even scientists have begun using making use of 3D geological models. Seismologists, for example, use them to predict events within the crust of the earth due to shifting plates, erosion, etc.

Of course, most people find an interest in 3D modeling thanks to two of the biggest entertainment industries today.

The first is movies and motion pictures, which use computer-generated characters, objects, and environments in animated and live-action films as well as 3D imaging to create an optical illusion of depth.

The other industry is none other than video games. Most modern games use 3D models and environments to create the virtual worlds that players interact with and explore.

Anyone interested in any of the above fields will want to learn how to use 3D modeling software programs.

.

2.9.3 Texture Artists and Texture Mapping

Don't be surprised if you visit a video game development team and find the texture artist just standing and staring at the wall. The wall is a very important element in video game environment and the video game texture designer has to know exactly what different walls and materials look like in all kinds of light. Remember, the texture on a wall, a vehicle, or even on a creature's scaly skin is one of the elements that makes a difference between a really three-dimensional video game and just another kiddie cartoon.

The texture artist may scan actual photographs of the appropriate textures into the computer and then apply that texture to the creatures, environments or objects created by other members of the team.

Some texture designers find ways of creating exotic, new textures that may go far beyond any surfaces occurring in nature-at least in this planet's nature

These textures have to be mapped onto the 3D objects, environments, and characters in the game. A sloppy job of texture mapping in a game will take away the visual edge. You can have great 3D environments, but the textures have to back them up. Boring repetitive textures will make the levels seem blah and give the gamer the "been there, done that" feel. This can be a game killer

2.10 Animation and Game Design

is a course of study that focuses on the process of creating motion and shape change illusion as well as the art of applying design and aesthetics for animation and game development. It facilitates the creative and deliberate use of traditional and new media to represent thoughts, issues, and messages to various audiences.

Through a collaborative, highly practical and industry-driven approach, Animation and Game Design will provide opportunities for students not only to develop their aesthetic and technical competence, but to express creativity and conceptualize projects that integrate technology, arts and media to address local and international issues while creating a source of education, entertainment and employment. This syllabus will also empower students for further studies or immediate entry into the job market whether through waged or entrepreneurial (self) employment, by providing goods or services on any feasible scale

2.11 What is C Sharp C #?

C Sharp is a modern object-oriented programming language developed in 2000 by Anders Hejlsberg in Microsoft, a generic language designed to develop applications on Microsoft's core operating systems and requiring .NET framework on Windows to work

C # is often seen as a hybrid that takes the best of C and C ++ to create a modern language; it is a targeted language; it aims to integrate computing power into C ++ with ease of programming Visual Basic.

Although .NET framework supports many other programming languages, C # Sharp has become one of the most common languages.

2.11.1 ***Why C Sharp was created C # ?***

You may wonder, as anyone who wants to know at least why it is creating such a new language, despite the existence of other software entities, and for similar purposes, such as Java

In fact, it was initially founded as a competitor to Java. Sun (later purchased by Oracle) did not want Microsoft to make changes to Java, so Microsoft chose to create its own language instead. In general, C # contains features similar to those in Java.

C Sharp has grown C # very quickly since it was first created, and this of course is under the broad support of Microsoft to help it get its big place today. Is now one of the most popular programming languages in the world

2.11.2 What is the origin of the term "C "# ?

The term C # is derived from the "#" musical key, which refers to a one-half-degree increase. This may reflect a touch of creativity that language gives to the programming world, as is the music world!

C # in English is often pronounced C Sharp, and Arabic is C Sharp.

2.11.3 The main uses of C Sharp C #

In fact, in short, almost anything! You can use C # to create Windows applications, client applications - server, database applications, and much more .

Its various uses can be distinguished as follows:

1. Windows Application Development:

Given .NET support on Windows, C Sharp is now the first choice for every developer to create Windows applications.

The new versions of Windows (Windows 8 / Windows 10) have strongly helped this "adoption"; the vast majority of third-party applications in Windows are written by C #.

Although the Windows Store has not been as successful as App Store and Google Play, it will undoubtedly be the future of Windows applications distribution, so C # has a bright future on Windows.

2. Create web apps typically via: WebForms (are part of the .NET Web application framework that is included with Visual Studio).

3. MVC (is an architectural style typically used to develop web applications).

Mobile devices and smartphones: When it comes to smart phones, C # is constantly growing.

Xamarin leads the way in this area, offering tools for all major mobile platforms as well as many other platforms that provide a lot of useful resources on how to "develop a mobile phone in C #.

Microsoft also has a long tradition of supporting C # on mobile devices. It recently released an update to Visual Studio 2013 that supports "public applications" - applications developed side by side for both desktop computers and all Windows devices

4. The games:

   C Sharp C # is widely used to create games using the Unity gaming engine, the most popular gaming engine today, making more than a third of the world's best games, and nearly 770 million active users of the games created with it.

   Unity also works in virtual reality, with 90% of all Samsung Gear and 53% of all Oculus Rift VR games developed using it. C # is a very popular tool for creating these applications, so it offers a great choice for any programmer hoping to break into the gaming industry or anyone interested in virtual reality.

5. The C # is flexible enough to run TCP / IP servers, and Raspberry Pi

2.11.4 Features C Sharp C#

First: C# has many features that make learning easier.
The C # structure is quite "expressive", but simple and easy to learn.
C # architecture simplifies many C ++ complexities.
C # is relatively easy to read.
In addition, C # processes most of the complex details of the device (computer) so you can focus on the purpose of programming rather than worrying about small details.

C # is also a language written consistently, so the code is checked before it is converted to an application, which makes it easy to find errors, something that can be especially useful for beginners.
Although C-Sharp C # is more logical and logical than C ++, there is still a lot to learn. Although C # is a high-level, relatively easy-to-read language, it is a lower-level language than other languages such as Python, meaning that there are many more complex tasks . However, anyone familiar with C, C ++, or Java can instantly recognize the C # structure, and developers who know any of these languages can start working as "producers" in C # in a very short time.

Second: Speed
As a static language, C # is faster than dynamically written languages because things are more clearly defined; therefore, when the app is running, your device resources will not be wasted when you check to identify something in your code. Because programmers can build on the "current" code in C #, instead of returning it repeatedly, it is expected to make C # faster.

Third: Community support and communication

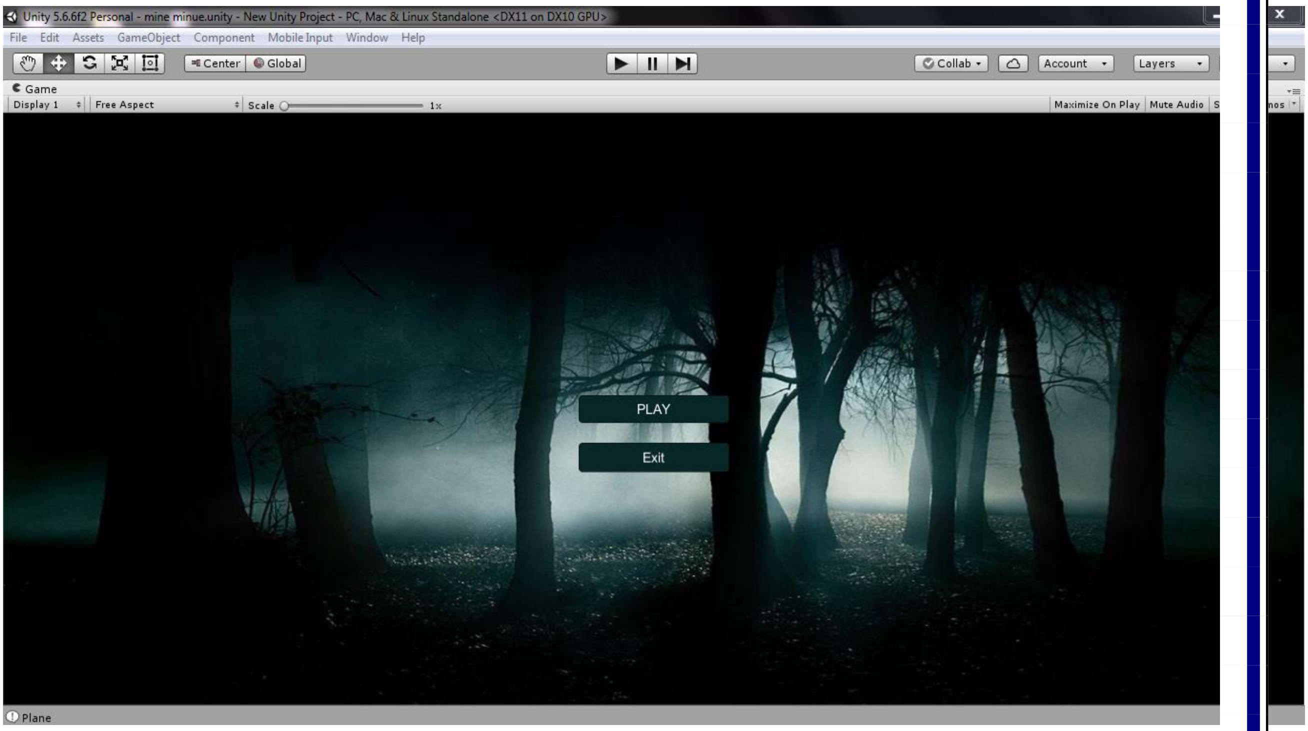
For any programmer or developer, the extent of dynamic support provided by the "developer community" is unrivaled! As the community of developers is also interested in providing and receiving assistance; the larger the community, the more people build useful tools to make development in that particular language easier.

So far, there are more than 600 programming languages around the world, so, with this context in mind, it is necessary to delve into some of the details of the C # support community

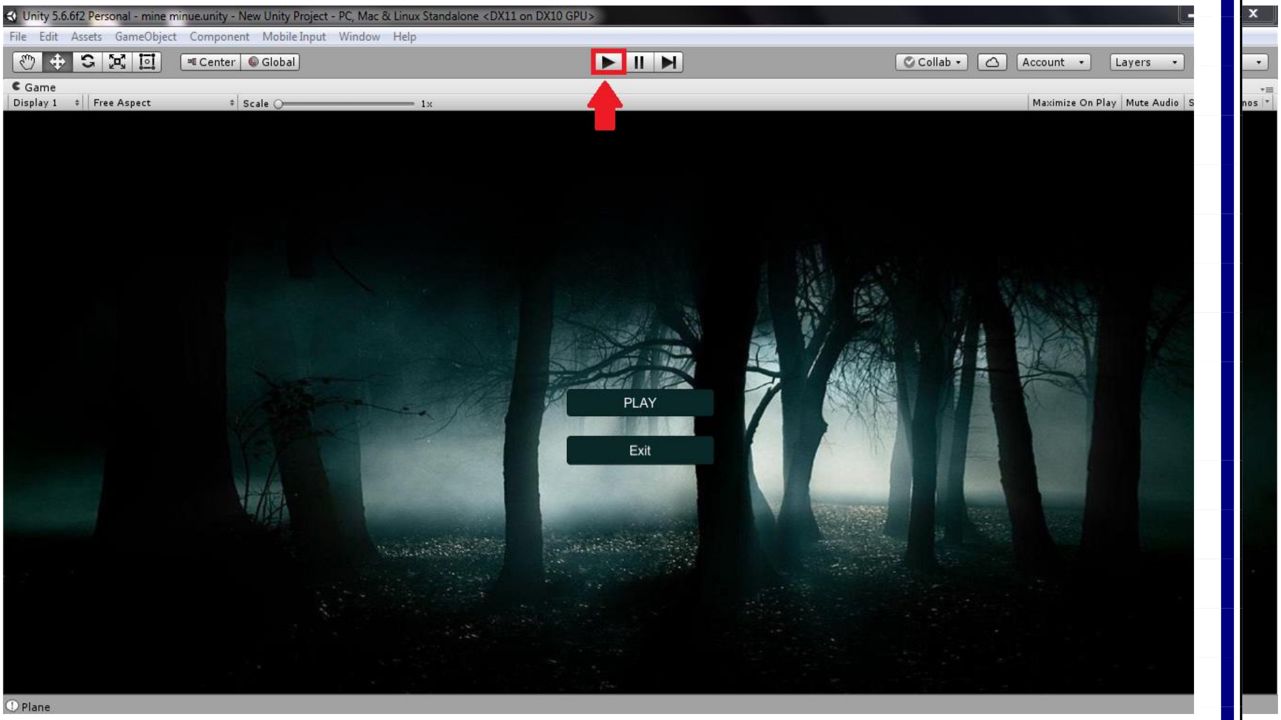
Chapter three

3.1 To Execute Our Game We Should Have Several Steps:

1- Open Unity 5.6.6f2 (32-bit)



2- Choose "Play" From Up Buttons Of Screen

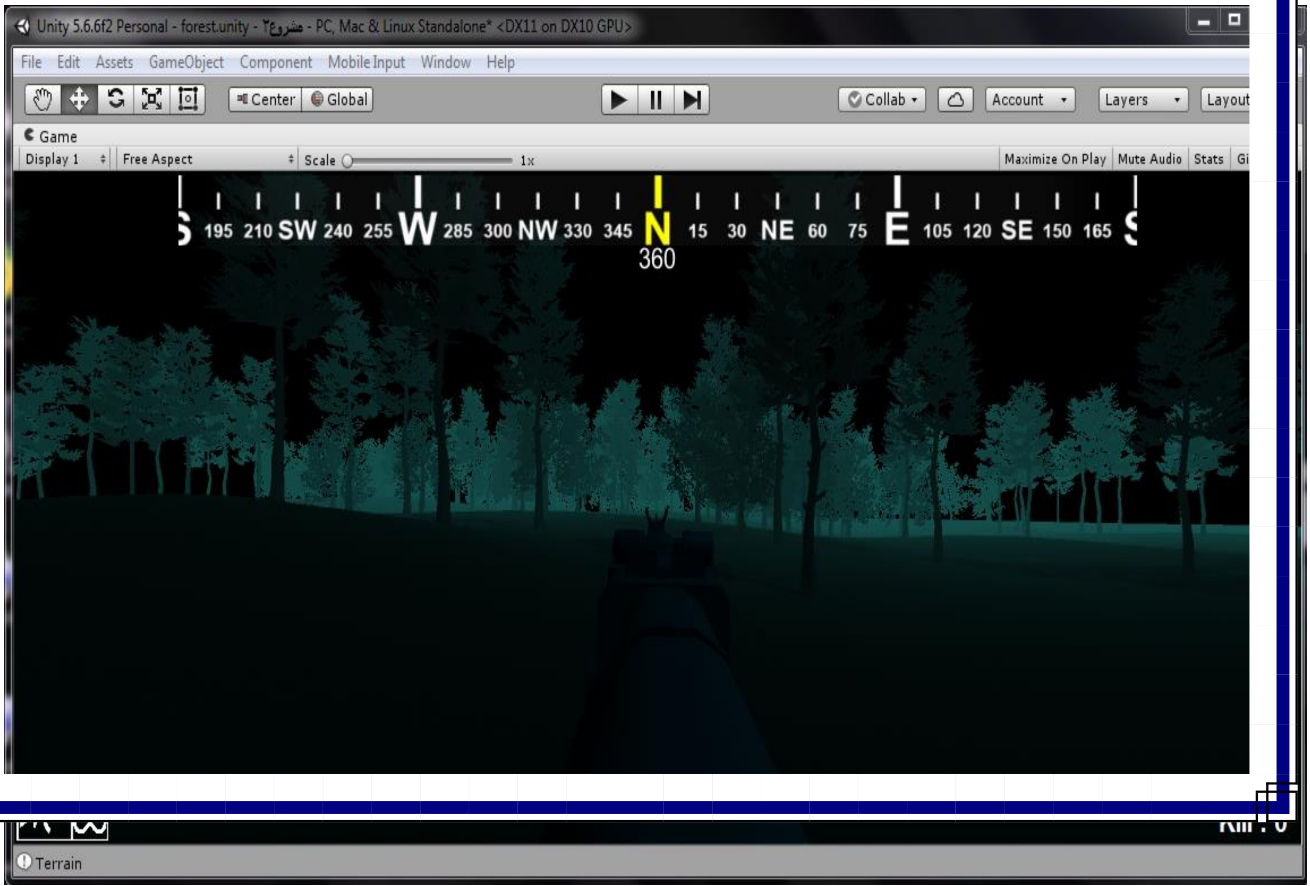

Unity 5.6.6f2 Personal - mine minue.unity - New Unity Project - PC, Mac & Linux Standalone <DX11 on DX10 GPU>
File Edit Assets GameObject Component Mobile Input Window Help
Center Global
Collab Account Layers
Game
Display 1 Free Aspect Scale 1x
Maximize On Play Mute Audio
PLAY
Exit
Plane

3- When Main Scene Appears <u>There Are Tow Buttons :</u>

a- "Play" To Play The Game

b- "Quit" To Terminate The Game

- When You choose "Play" From Main Scene It Will


Unity 5.6.6f2 Personal - forest.unity - مشروع٢ - PC, Mac & Linux Standalone* <DX11 on DX10 GPU>
File Edit Assets GameObject Component Mobile Input Window Help
Center Global
Collab Account Layers Layout
Game
Display 1 Free Aspect Scale 1x
Maximize On Play Mute Audio Stats Gi
S 195 210 SW 240 255 W 285 300 NW 330 345 N 15 30 NE 60 75 E 105 120 SE 150 165 S
360
Terrain

Take You To The Game After Some Second And Game Will Begins

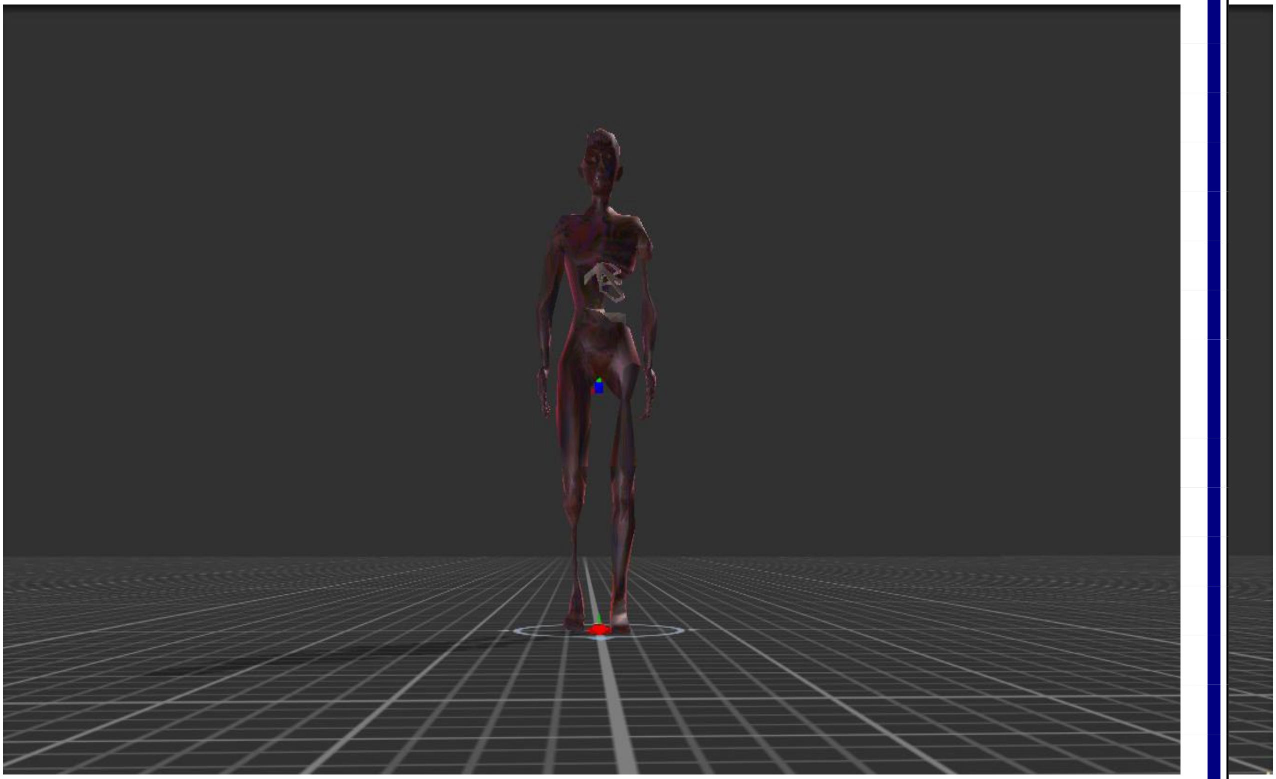
Here you will begin to hear sounds in the background of the game which are the sounds of (birds, wind, movement of leaves, the movement of the feet of the player when moving and jumping) , You will also hear the sound of the fire if you press the "Ctrl" button or if you press the left button with the mouse and you will see the effect of the "muzzle flash" which is a yellow color that represents the spark coming out of the barrel of the gun and the accompanying smoke , You will see the gun in the center of the screen and at the top of the screen there is a compass to help you locate the zombie and you will see at the bottom right of the screen the number of dead you killed from the zombie and to the left of the screen you will see a picture of the weapon with an Infiniti sign indicating the number of bullets is not over and after a few seconds you will see crowds The zombie is following you to kill you

3.2 Character Of The Game

In this game there are two characters around the story are :

- **Shooter : Which represents a rifle in the middle of the screen because the game is of a kind first person shooter (fps)**

- **Zombies :**



3.2.1How you deal with these characters

Shooter :

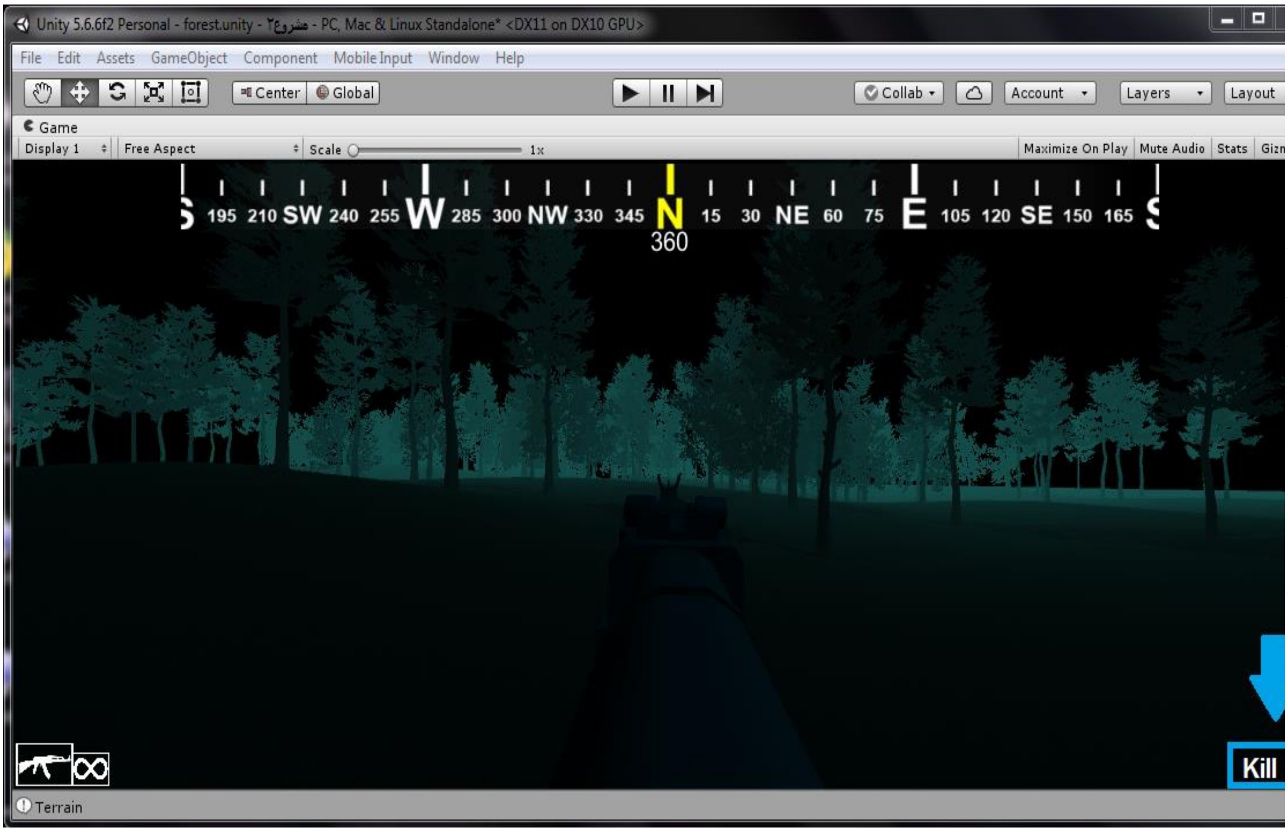
- When you click on button with "up sign" or click "W" from keyboard the shooter character move to straight ahead
- When you click on button with "down sign" or click "S" from keyboard the shooter character move to backwards
- When you click on button with "right sign" or click "D" from keyboard the shooter character move to right
- When you click on button with "left sign" or click "A" from keyboard the shooter character move to left
- When you move the mouse of computer The scene in front of you will move like a human eye
- **3.3 Sounds Of Game**

There are many voices in this game we will address :

- Shoot gun
- Foot steps
- Jump
- Birds
- Wind sound
- Sound of the movement of leaves
- Etc .....

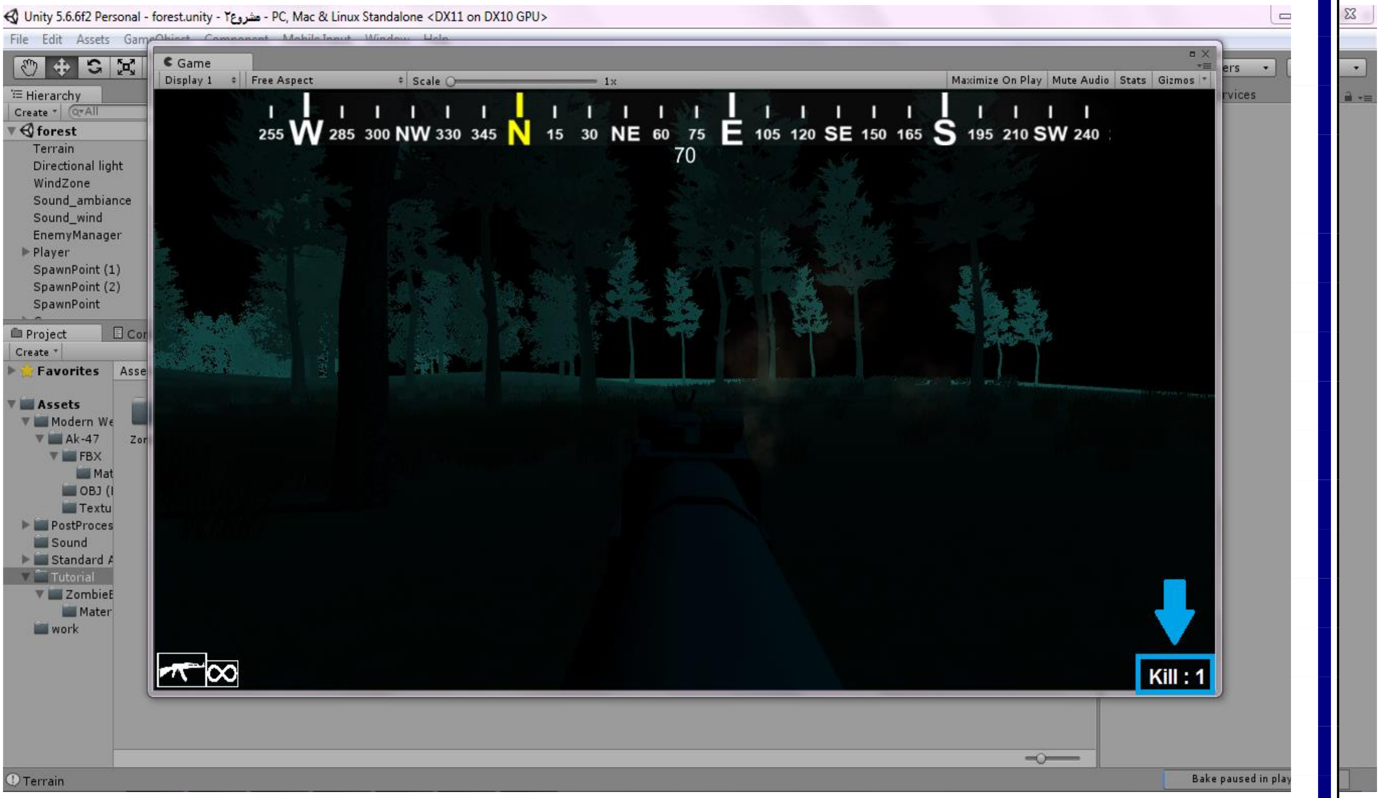
3.4 Fonts Of Game

The player will be seen at the bottom right of the screen. There is a " Kill : " indicating the number of dead zombies and this text will gradually increase as the player kills more zombies


Unity 5.6.6f2 Personal - forest.unity - مشروع۲ - PC, Mac & Linux Standalone* <DX11 on DX10 GPU>
File Edit Assets GameObject Component Mobile Input Window Help
Center Global
Collab Account Layers Layout
Game
Display 1 Free Aspect Scale 1x
Maximize On Play Mute Audio Stats Gizn
195 210 SW 240 255 W 285 300 NW 330 345 N 15 30 NE 60 75 E 105 120 SE 150 165
360
∞
Kill
Terrain

and the increase is naturally one

When player kill one from zombies the kill text Increases by one



3.5 Controls this game

In this game we will deal with mouse buttons and keyboard buttons

- Mouse : When you press the left mouse button the event will shoot and when you move the mouse the scene will move in front of the player as the human eyes move

- Keyboard : There are many ways to use the keyboard in games, but the most popular methods are two very common methods are

  1- Method how to use the four buttons "W, A, S, D"
  2- Method how to use the four trends "up ,down ,left ,right"

(*) And can use "Ctrl" to shoot too

With the tow Mouse and Keyboard can use to control :

- Control characters
- Camera control
- Start and end game

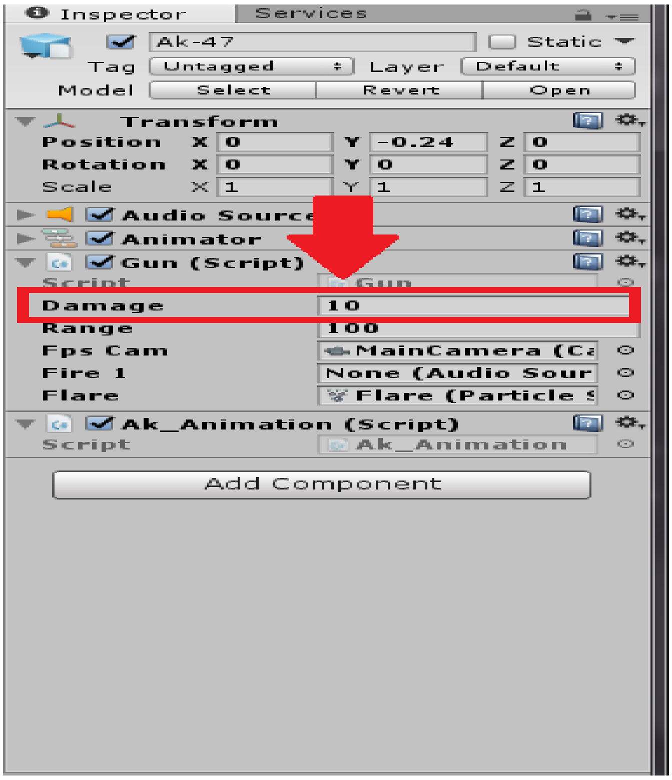
3.6 Game Graphics

The graphics used here are compatible with graphics used in modern 3D games on smart phones, Xbox, and play-station of the forms of players and their voices and visual effects used and movements and lights and take the necessary consideration of the reality of the environment, for example the forest environment in the game here we have taken the voices of animals and trees and the form of trees and the ground and grass in it

3.7 Challenges and Rules of game

Since this game can be included under the type of adventure games so it contains many of the challenges that will face the player, who must be quick to make his decision

- Kill the Zombies : The zombie killing is the main challenge on which the game is based and the zombie needs one to five shots to die, but their quick exit and catching the player may lead to the death of the player if not good focus because touching one of the zombies will kill him and lose the game



Ak-47 ☑ □ Static

Tag Untagged Layer Default

Model Select Revert Open

Transform

Position	X	0	Y	-0.24	Z	0
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1

☑ Audio Source

☑ Animator

☑ Gun (Script)

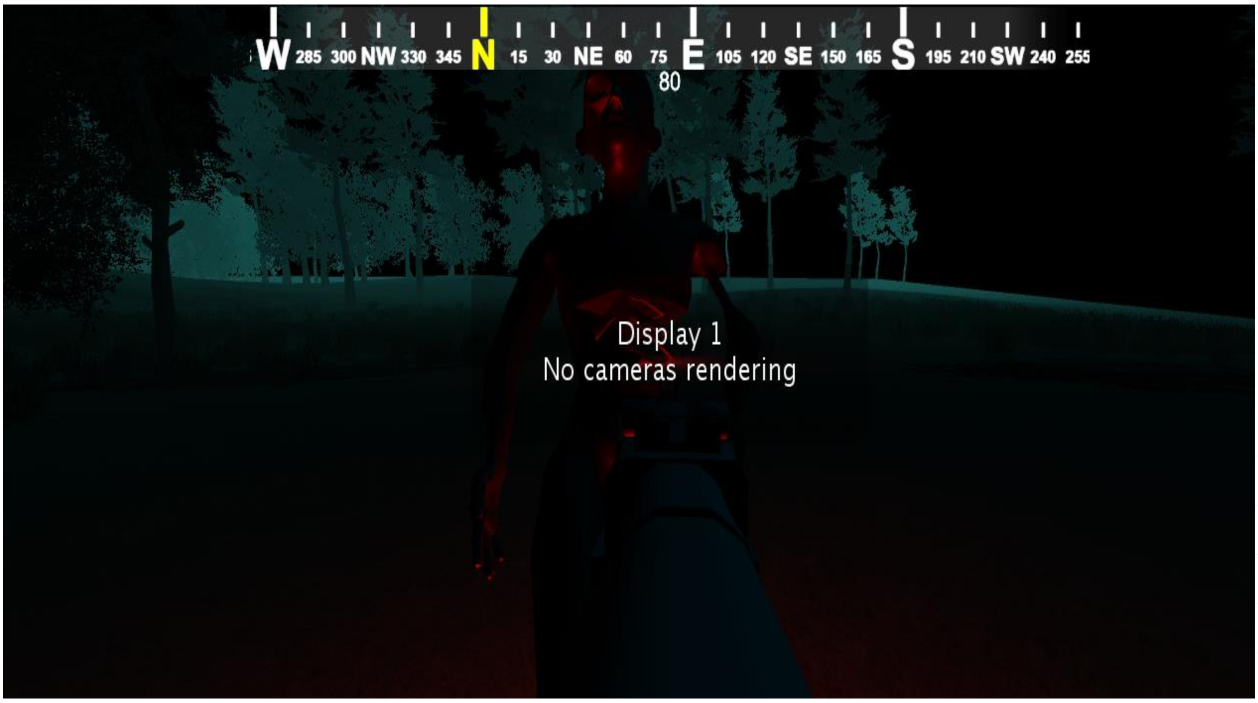
Script	Gun
Damage	10
Range	100
Fps Cam	MainCamera (Ca
Fire 1	None (Audio Sour
Flare	Flare (Particle S

☑ Ak_Animation (Script)

Script Ak_Animation

Note in the image above that the value of the decrease in the life of the zombie is done by 10 to kill one zombie will require five releases as we mentioned earlier

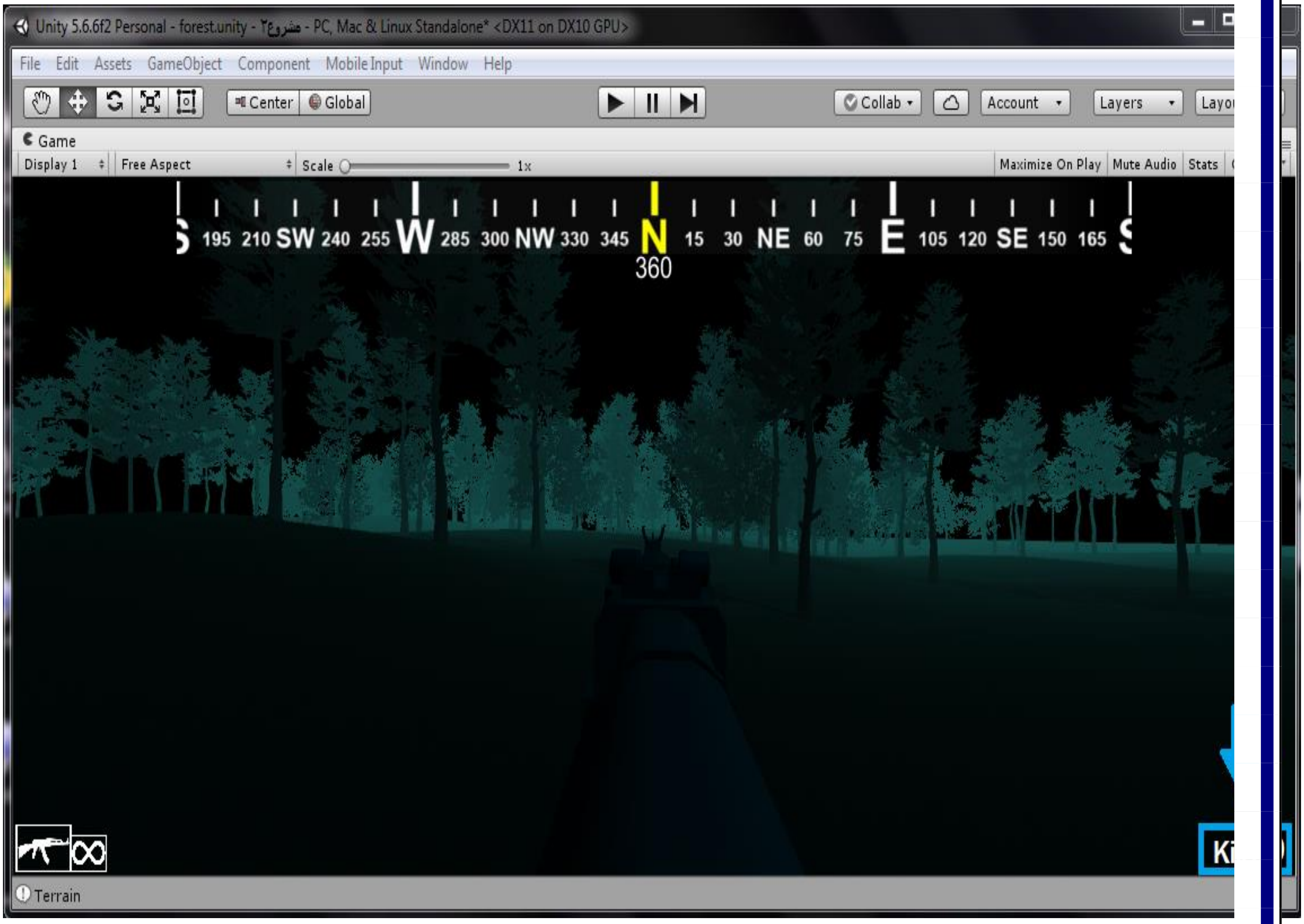
- Keep the player alive : One of the most important challenges that the player will face is to survive among the large zombie crowds. The most difficult is that zombie is capable of killing the player by touching it for only one time. This is because we are inspired by the idea of our game from one of the famous games which will be mentioned in the story section


W 285 300 NW 330 345 N 15 30 NE 60 75 E 105 120 SE 150 165 S 195 210 SW 240 255
80

Display 1
No cameras rendering

- Stay alive in a challenging forest environment : In this difficult environment full of trees that will certainly prevent the player from moving towards zombie crowds or escape away because the trees will play the role of cover for the zombie and repel the shooting towards them and it will be difficult to challenge the player who must be careful and determine the correct shooting location

- Number of kill : One of the most important challenges on which the game is to kill the largest number of zombies so the player must kill the largest number of zombies to reach a high degree may not be able to access another player and thus will be one of the game professionals, which is one of the hardest things dreamed by all Player of any game exists today


Unity 5.6.6f2 Personal - forest.unity - مشروع٢ - PC, Mac & Linux Standalone* <DX11 on DX10 GPU>
File Edit Assets GameObject Component Mobile Input Window Help
Center Global
Collab Account Layers Layo
Game
Display 1 Free Aspect Scale 1x
Maximize On Play Mute Audio Stats
S 195 210 SW 240 255 W 285 300 NW 330 345 N 15 30 NE 60 75 E 105 120 SE 150 165 S
360
Terrain

3.8 Collision detection

The collision detection is one of the most important features of the games, and almost never there is game without of this feature .But we don't need much collision detection in this game, because the enemies are transparent. So we needed to collide only at the trees and the zombies.

Which means that the player has physical properties and the environment around him as well as that the player has the weight and not driven by the wind and the physical characteristics also that it is not through trees, any movement of the player is similar to the movement of the human completely and physical characteristics are human characteristics.

3.9 The story

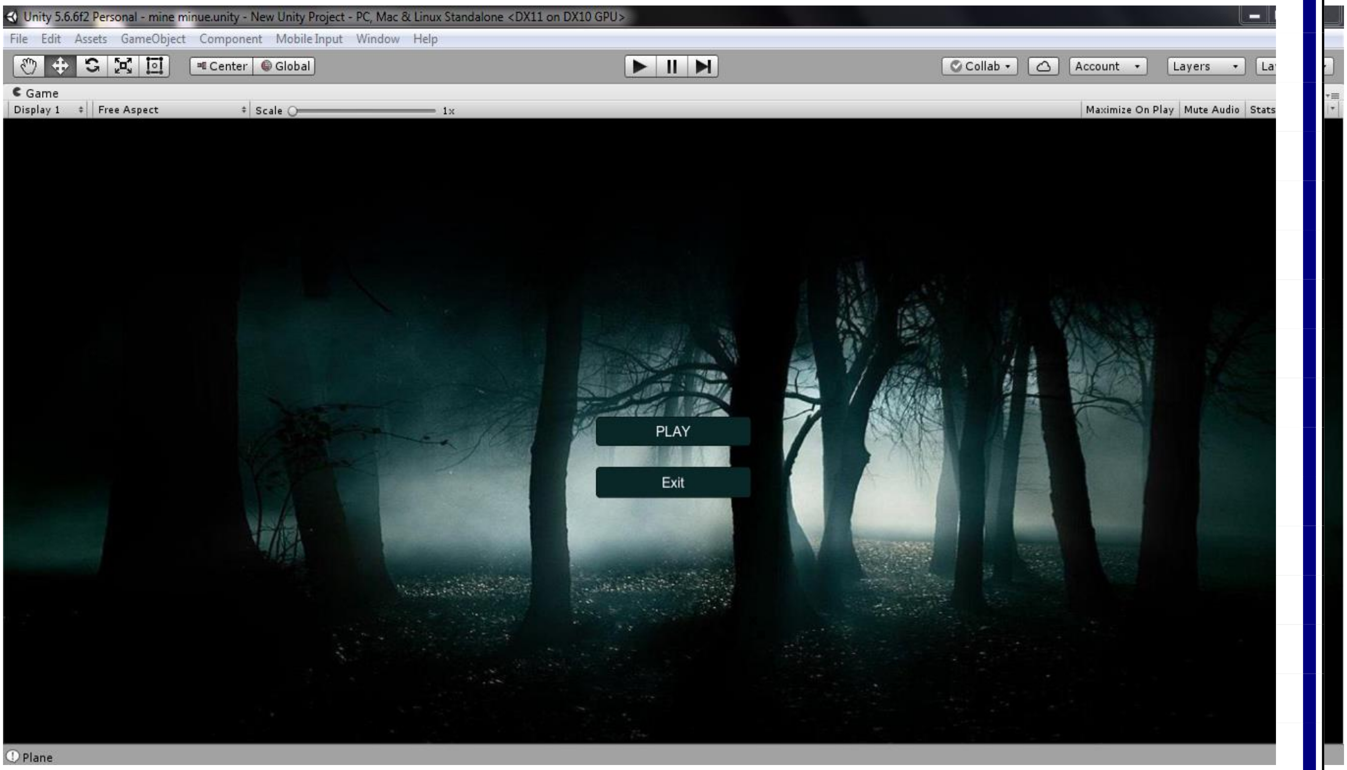
The story of the game inspired by the game " Into the Dead 2" but with great changes in graphics and sounds and in the environment of play, we can say a radical difference between our game and the game mentioned and the story of the game mentioned on Instead of being an endless runner where you're only competing to top your own high score, the game has added a narrative and some progression to the run and gun gameplay. The story is fairly simple: you play as a man named James who gets into a car accident in the middle of the road while trying to get back to his wife, Helen, and his daughter, Maggie. Of course, James is immediately swarmed by zombies forcing him to run and shoot his way to safety. Armed with a handgun, a walkie-talkie, and a strong desire to return home, James starts running through the hordes of zombies. While it may be a pretty simple story, the voice acting is surprisingly well done, and as I have been playing through the story I find myself invested in getting James back to his family. Due to the fact that there is actually a story, the main game is split up into chapters, and each chapter is split into different missions. You can complete objectives during each mission to earn stars, which allow you to earn loot boxes and keys. Loot boxes give you rewards like weapons, perks, and other items to help you survive, and keys unlock the final mission in each chapter. This progression gives the entire game a sense of purpose and upgrades the experience from simple endless runner to a more refined auto-runner experience that has heart. At the beginning of the game, all the zombies only take one shot to kill; however, as you progress you'll run into stronger living dead creatures that require more than one shot. Of course, you can always upgrade your weapons by finding weapon parts at the end of missions which will help you as you encounter or powerful zombies. The real challenge comes into play when you realize there's seemingly no way of telling which zombie will die with one shot and which won't. Choose to use your weapons sparingly, or find every ammo crate and blast your way through the entire mission, it's really up to you. Just know if you make the wrong choice (and you will at some point) you'll be treated to a fantastic visual of a group of zombies feasting on your innards. As for our story, it revolves around a soldier fighting zombie in a forest after being stranded alone after the death of all his friends by the zombie and trying to escape and exit from this jungle, but the exit is impossible as he discovers that on a small island and when he left her borders will attract something invisible Thus, only zombies will be killed and eliminated, and this may seem a bit impossible

Chapter four

Test and result :

The test and experience of the game was one of the hardest things because it requires a lot of precision and skill and the difficulties that we have is the position of the player in the environment, any arrangement signed within the environment and the size of zombies and coordination with the environment because of the presence of trees and some high areas and the most difficult things in the experiment and testing is coordinated Effects The accompaniment of the fire and the launch of the launch, as well as choosing the location of zombie exit and other things that are not mentioned now .

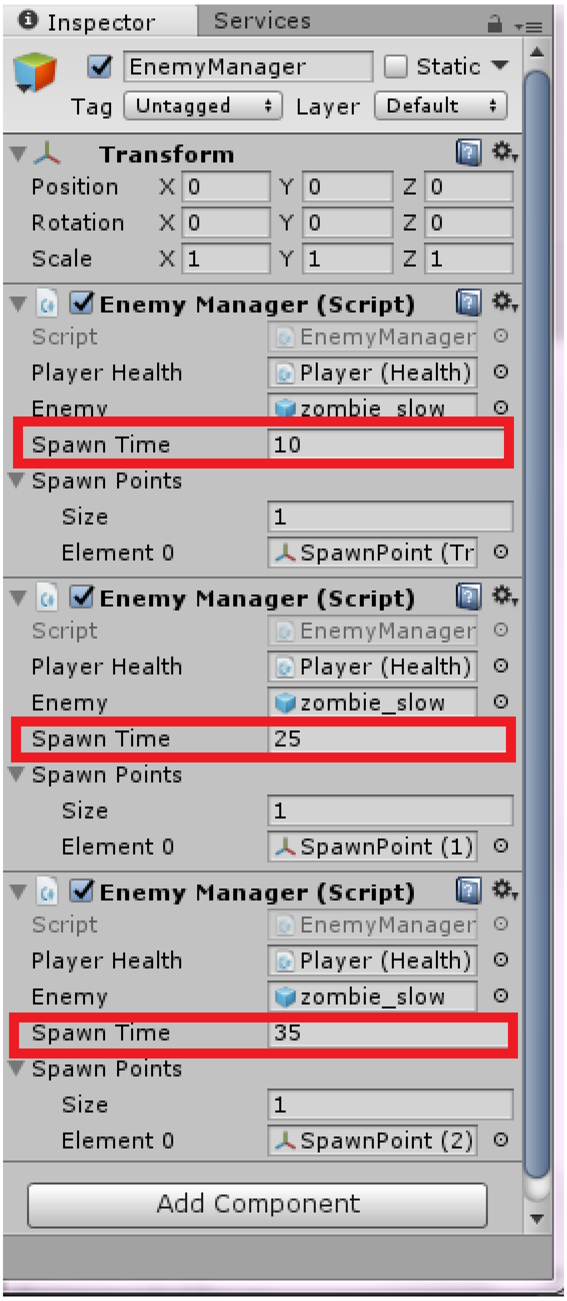
One of the things to mention is the artificial intelligence in this game, which plays an important role in its success, which is considered as one of the most difficult things that we have encountered during work and in order to maintain the similarity between it and the picture in the human mind about the zombie and the forest at night and the use of The weapon was one of our ambition is to work quality in the world of the player and by making the facade devoid of any writings or signs in order to feel the player and in the arena of real war, but we took into account the acceptance of players to such quality transfer in the presence of many common games and filled with screen Many Of the writings and symbols that express something as in our game the number of dead zombie and in order to forget as well as the movement of the reaction of the weapon, which is an important thing and also the movement of breath for the player


Unity 5.6.6f2 Personal - mine minue.unity - New Unity Project - PC, Mac & Linux Standalone <DX11 on DX10 GPU>
File Edit Assets GameObject Component Mobile Input Window Help
Center
Global
Collab
Account
Layers
Game
Display 1
Free Aspect
Scale
1x
Maximize On Play
Mute Audio
Stats
PLAY
Exit
Plane

Once you have reached the game interface as previously mentioned and pressing the "Play" button will take you to the game Or choose the "exit" command which will lead you out of the game

We will now explain how to play and what is induced in the scenes :

When the player enters the game after two seconds, the zombie crowd will begin to appear in the order and according to the given time



☑ EnemyManager □ Static

Tag Untagged Layer Default

Transform

Position X 0 Y 0 Z 0

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

☑ **Enemy Manager (Script)**

Script EnemyManager

Player Health Player (Health)

Enemy zombie_slow

Spawn Time 10

Spawn Points

Size 1

Element 0 SpawnPoint (Tr

☑ **Enemy Manager (Script)**

Script EnemyManager

Player Health Player (Health)

Enemy zombie_slow

Spawn Time 25

Spawn Points

Size 1

Element 0 SpawnPoint (1)

☑ **Enemy Manager (Script)**

Script EnemyManager

Player Health Player (Health)

Enemy zombie_slow

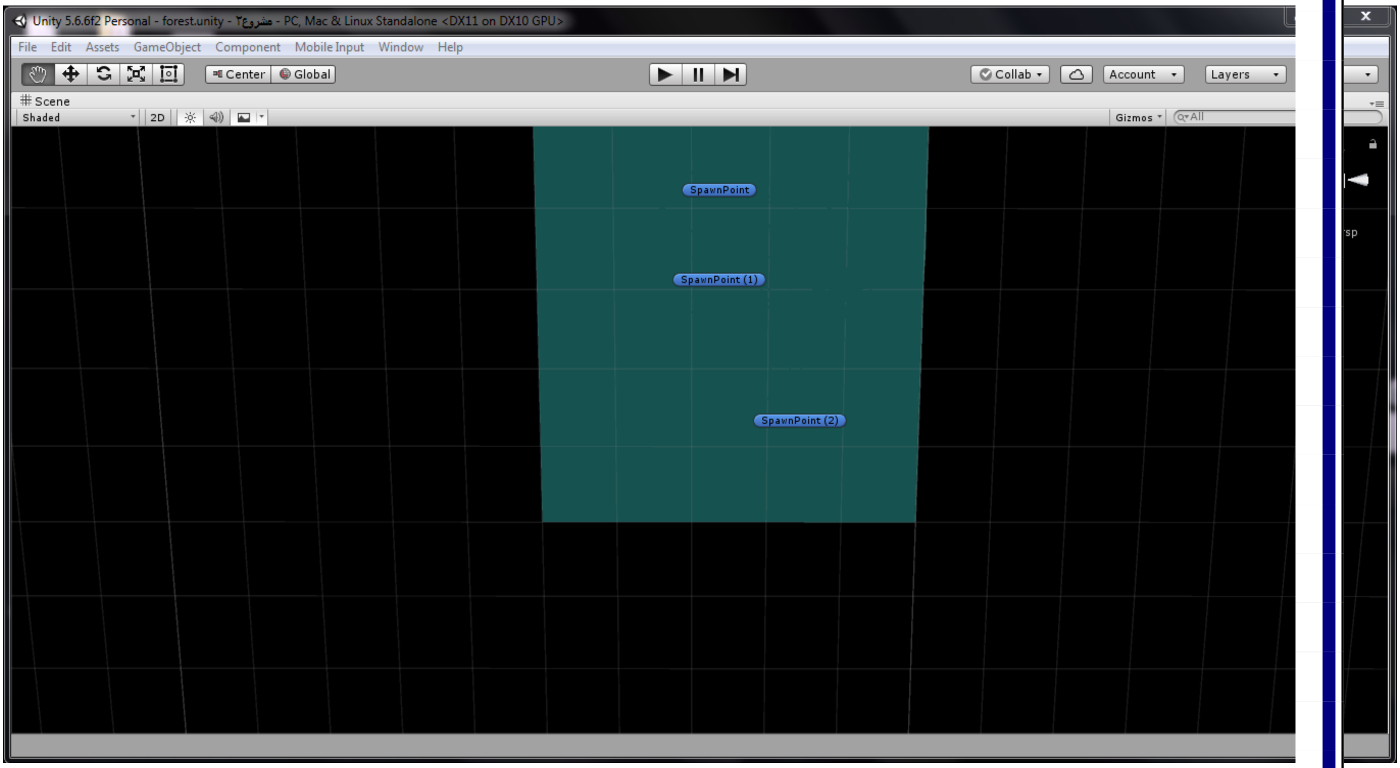
Spawn Time 35

Spawn Points

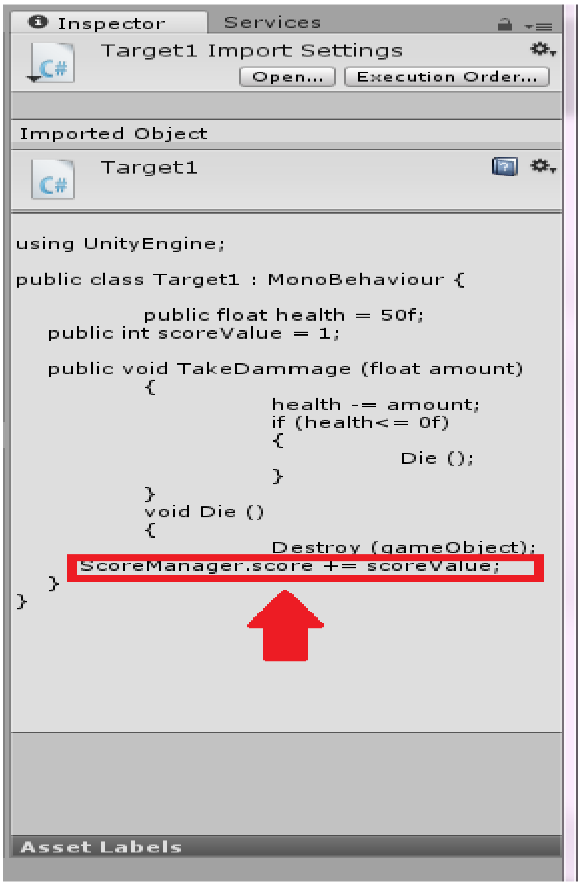
Size 1

Element 0 SpawnPoint (2)

In the picture above, the time given to each exit or generation point is shown to the zombie



These are the three points from which the zombie will be generated then the zombie will follow the player according to the code When a zombie is killed, the zombie will be destroyed. After that, the counter will be increased by one



```
using UnityEngine;

public class Target1 : MonoBehaviour {

            public float health = 50f;
   public int scoreValue = 1;

   public void TakeDammage (float amount)
            {
                        health -= amount;
                        if (health<= 0f)
                        {
                                    Die ();
                        }
            }
            void Die ()
            {
                        Destroy (gameObject);
     ScoreManager.score += scoreValue;
   }
}
```

Here we have completed the experiment and the test and we know how to eliminate the zombie and get the highest degree of (kill), which is the main objective of the game

Chapter five

5.1 Conclusion

1- The language " c # " with "unity" is of high compatibility and good speed and very helpful .
2- The design of the games needs to have a great experience in many things such as image and sound processing, model production and programming in C# language in addition to the unity engine .
3- The Sound with the movement of the fighter give a good interaction to the user of this game and assist him to enter the atmosphere of the game.
4- The graphics and sounds for this game are relatively good.

5.2 future work

1- Add other stages to the game or make it play online .

2- Add more environments such as city, desert or other environments.

3- It is possible to convert the game to virtual reality.

4- Add other forms to the zombie.

5- Add many different weapons.

6- Control the number of bullets in the weapon.

7- Add many other physical properties.

8- Add vehicles or animals like the horse to help the player escape the zombie crowds.

9- Make the game from a third person perspective.

10- Add many restrictions to the player to increase the difficulty of the game and make it more entertaining and more enthusiastic.

references

REFERENCES:

1- Stephanie O'Malley, "3D Modeling and Animation",CRC Press ,2009.

2- Dirk Hünniger, "Windows Programming", Microsoft press , 2013.

3- Kang, "Windows & DirectX Programming", Cengage Learning ,2009.

4- Joey de Vries, "Learn OpenGL", Apress , 2017.

5- Dave Shreiner & Graham Sellers & John Kessenich & Bill Licea-

6- 6- Kane & The Khronos OpenGL ARB Working Group, "OpenGL Programming Guide", Addison- Wesley, 2013.

7- Aarseth, E. J. (2001). Computer Game Studies, Year One. ***Game Studies*, *1*(1).**

8- URL http://www.gamestudies.org/0101/editorial.html

9- Aarseth, E. J. (2005). From Hunt the Wumpus to EverQuest: Introduction to Quest Theory. In ***Proceedings of the International Conference on Entertainment Computing, Sanda Japan, September 2005*, (pp. 496–506).**

10- Adams, E. (2000). A Letter From A Dungeon. ***Gamasutra*. URL**

11-

http://www.gamasutra.com/view/feature/3424/the_designers_

notebook_a_letter_.php

12- Adams, E., & Rollings, A. (2007). ***Fundamentals of Game Design*. Upper Saddle River, NJ: Pearson Education, Inc.**