

Android Game Ping Pong Using Unity Game Engine Software

Submitted to the Computer Science and Engineering Department, East West University In partial fulfillment of the requirements for the award of the degree of Bachelor of Science in Engineering.



By
A.F.M. Ahasan Habib Sheehabe.
ID. 2010-2-60-004.

Supervised By
Dr. Md. Nawab Yousuf Ali
Associate Professor,
Department of Computer Science and Engineering,
East West University, Dhaka-1212, Bangladesh

10 September 2015

ABSTRACT

My project name is “Android Game Ping Pong Using Unity Game Engine Software” based on Android platform. It is a brain storming game to make our brain sharper and it’s really enjoyable. In my project, game type is 2D. Here I used 2D type’s tools and materials from unity game engine and also use game logic by using java scripts language. In my game I include one ball, 2 bat, midline side wall scouring timing and replay button. Hence i use different types of logics for each objects for run my game smoothly. My experimental results show that the games run properly without any intervention.

This report provides detailed description on how can I created the ping pong game and also how I eased the process of android development while creating the game in unity5 game engine.

Letter of Acceptance

“Android Game Ping Pong Using Unity5 Game Engine Software” submitted by A.F.M. Ahsan Habib Sheehabe, ID_2010-2-60-004, to the Department of Computer Science and Engineering, East West University, Dhaka-1212, Bangladesh is accepted by the Department for the partial fulfillment of requirements for the degree of Bachelor of Science in Computer Science and Engineering on May10, 2015.

Board of Examiners

Dr. Md. Nawab Yousuf Ali
Associate Professor,
Department of Computer Science and Engineering,
East West University, Dhaka-1212, Bangladesh

Dr. Shamim Hasnat Ripon,
Chairperson and Associate Professor,
Department of Computer Science and Engineering,
East West University, Dhaka-1212, Bangladesh

Acknowledgements

“Android Game Ping Pong Using Unity Game Engine Software” would run successfully, the valuable encouragement and guidance from my supervisor Dr. Md. Nawab Yousuf Ali, Associate Professor, Department of Computer Science and Engineering, East West University. He enlightened, encouraged and provided us with the ingenuity to transform my vision into reality. I am particularly grateful to Dr. Shamim Hasnat Ripon, Chairperson and Associate Professor, Department of Computer Science and Engineering, East West University, for his encouragement. I am also grateful to Siddhartha Sarkar, Teaching Assistant for his guidance and counseling. I am also grateful to all my teachers and many of my friends who provided me with necessary hardware and suggestion to test the software as it developed. I would like to thank my God to finish my work in just time and also thanks to Google for creating android platform and keeping it free.

Table of Contents:

Abstract.....	2
Letter of Acceptance.....	3
Acknowledgement.....	4
Chapter 1	
Introduction.....	7-8
1.1 Introduction.....	7
1.2 Motivations.....	7
1.3 Objectives.....	8
1.4 Contribution.....	8
1.5 Organization for the Project Report.....	8
Chapter 2:	
Literature Review.....	09-18
2.1 What is Android?.....	9
2.2 History of Android.....	9
2.2.1. Foundation.....	9
2.2.2. Acquisition by Google.....	10
2.2.3. Post-acquisition by Google.....	10
2.2.4. Open Handset Alliance.....	10
2.2.5. Android Open Source Project.....	10
2.2.6. Version history.....	11
2.3 Gingerbread.....	11
2.4 Design and Architecture of Android.....	11
2.5. Linux.....	12
2.6. Architecture of Android.....	12
2.7. Application.....	13
2.8. Android Runtime.....	13
2.9. Linux Kernel.....	14
2.10 Features of Android.....	14
2.11 Applications.....	16
2.12. Google Play.....	17
2.13. Software development tools.....	17
2.13.1. Android SDK.....	17
2.13.2. Native development kit.....	18
2.13.3. Android Open Accessory Development Kit.....	18
2.13.4. App Inventor for Android.....	18
Chapter 3:	
Introduction to Unity Game Engine.....	19-34
3.1 Introduction.....	19
3.2 Unity pro version converging.....	19
3.3 Features.....	20

3.4 Versions.....	21
3.4.1 Unity 1.....	21
3.4.2 Unity 2.....	21
3.4.3 Unity 3.....	21
3.4.4 Unity 4.....	22
3.4.5 Unity 5.....	22
3.5. Games using Unity.....	23
3.6. Unity-chain.....	24
3.7. Reception.....	25
3.8. What is unity 3D/2D ?.....	25
3.9. Cost.....	25
3.10. Installation Process.....	26
3.11. Runtime Application.....	26
3.12. Unity Projects.....	28
3.13. Assets.....	29
3.14. Screens.....	30
3.15. Scripting.....	32
3.16. Publishing.....	33
3.17. Conclusion.....	34

Chapter 4:

Implementation.....	35-41
4.1. Design.....	35
4.2. Implementation procedure.....	35
4.2.1. Hardware requirement.....	35
4.2.2. Android game development environment.....	36
4.3 Android game	36
4.3.1 Feature.....	36
4.3.2. Game screen.....	36
4.3.3 Background.....	37
4.3.4. Left win image.....	38
4.3.5. Right win image.....	39
4.3.6. Game tie.....	39
4.3.7. Ball.....	40
4.3.8. Bat.....	40
4.3.9. Scoring.....	41
4.4 Hardware Specification.....	41

Chapter 5:

Conclusion and Future Work.....	42-46
5.1 Conclusion.....	42
5.2 Future Work.....	42
References.....	43-46

Chapter 1

Introduction

Android is a mobile operating system (OS) based on the Linux kernel and currently developed by Google. With a user interface based on direct manipulation, Android is designed primarily for touch screen mobile devices such as smart phone and tablet computers, with specialized user interfaces for televisions (Android TV), cars (Android Auto), and wrist watches (Android Wear). The OS uses touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, and a virtual keyboard. Despite being primarily designed for touch screen input, it has also been used in game consoles, digital cameras, regular PCs, and other electronics. As of 2015, Android has the largest installed base of all operating systems.^[11]

As of July 2013, the Google Play store has had over one million Android applications ("apps") published, and over 50 billion applications downloaded.^[15] An April–May 2013 survey of mobile application developers found that 71% of them create applications for Android;^[16] another 2015 survey, found that 40% of full-time professional developers see Android as the "priority" target platform, which is more than iOS (37%) or other platforms.^[17] At Google I/O 2014, the company revealed that there were over one billion active monthly Android users, up from 538 million in June 2013.^[18]

Android's source code is released by Google under open source licenses, although most Android devices ultimately ship with a combination of open source and proprietary software, including proprietary software developed and licensed by Google.^[19] Initially developed by Android, Inc., which Google bought in 2005,^[20] Android was unveiled in 2007, along with the founding of the Open Handset Alliance – a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices.^[21]

Android is popular with technology companies which require a ready-made, low-cost and customizable operating system for high-tech devices.^[22] Android's open nature has encouraged a large community of developers and enthusiasts to use the open-source code as a foundation for community-driven projects, which add new features for advanced users^[23] or bring Android to devices which were officially released running other operating systems. The operating system's success has made it a target for patent litigation as part of the so-called "smart phone wars" between technology companies.^{[24][25]}

1.1 Motivations

Android is the world's most popular operating system for mobile devices and tablets. It makes a revolution in the world of technology. Most of the android applications are popular which are useful and entertaining for user. So I wanted to create a game which is developed by android. If designed properly, a mobile version based this game could be more fun for the user.

Thus, the following points describe the motivation for a mobile version game.

1.2 Objectives

The main objective is to develop a game which can be run any android platform device. For developing this game i will be able to:

- Build our own Android application.
- Explain the differences between Android and other mobile development environment.
- Understand how Android applications work, their life cycle, manifest, intents, and using external resources.
- Design and develop useful Android applications with compelling user interfaces by using, extending, and creating your own layouts and Views.
- Discuss the process how to create an android operating system supporting game.
- Develop a user friendly game which will entertain user.

1.3 Contribution

To develop the game all the requirement are collected. Software, hardware and game framework requirements are also discussed in the chapter to initialize an android development environment. After that move into our project to discuss about it in details.

- I collected the necessary information about Android.
- I learned Android programming technique.
- I also collected requirements for our project.
- I tested our application and it passed in all the method I applied.

1.4 Organization of the Project Report

Chapter 2

In this chapter I will discuss about the basic information, structure, history and foundation of the Android platform. It also describes all the related literature and previous work I used in our application to develop our android application.

Chapter 3

This chapter I also describes about the Unity Game Engine, installing Unity development process and tools. Here also describes the entire game like how I create the game, useful description of elements that the game contains how to run this application and also testing techniques of application.

Chapter 4

In This Chapter I also Describes my game applications feature procedure using tools, logics and Implementation procedures.#

Chapter 5

The Chapter 4 Describes conclusion and future work of our application.

Chapter 2

Literature Review

2.1. What is Android?

Android is one of the most popular as well as most widely used mobile operating system. It is a software stack for mobile devices that includes an operating system, middleware and key applications [9]. The android SDK provides the tools and APIs necessary to create application on the android platform using JAVA programming language. Android is an open source mobile OS platform that's running on top of Linux kernel. It uses a non-standard Java Virtual Machine called Dalvin, specialized to handle mobile device processes. The main programming language used for Android is JAVA with many libraries from J2SE and third party open source projects, such as Apache commons, SQLite, web kit, etc. Google purchased the initial developer of the software, Android Inc., in 2005. The unveiling of the Android distribution on November 5, 2007 was announced with the founding of the Open Handset Alliance, a consortium of 84 hardware, software and telecommunication companies devoted to advancing open standards for mobile devices. Google released most of the Android code under the Apache License, a free software license. The Android Open Source Project (AOSP) is tasked with the maintenance and further development of Android. Android consists of a kernel based on the Linux kernel, with middleware, libraries and APIs written in C and application software running on an application framework which includes Java-compatible libraries based on Apache Harmony. Developers write primarily in a customized version of Java. Android became the world's leading smart phone platform at the end of 2010. For the first quarter of 2012, Android had a 59% smart phone market share worldwide. At the half of 2012, there were 400 million devices activated and 1 million activations per day. Analysts point to the advantage to Android of being a multi-channel, multi-carrier OS. [10] At the beginning of 2013, Android captured 70% of smart phone market share worldwide.

2.2. History of Android

In this portion I will describe the history of android and I also show relation between my project and android.

2.2.1. Foundation

Android, Inc. was founded in Palo Alto, California, United States in October 2003 by Andy Rubin (co-founder of Danger), Rich Miner (co-founder of Wildfire Communications, Inc.), Nick Sears (once VP at T-Mobile), and Chris White (headed design and interface development at WebTV) to develop, in Rubin's words "...smarter mobile devices that are more aware of its owner's location and preferences". Despite the obvious past accomplishments of the founders and early employees, Android Inc. operated secretly, revealing only that it was working on software for mobile phones. That same year, Rubin ran out of money. Steve Perlman, a close friend of Rubin, brought him \$10,000 in cash in an envelope and refused a stake in the company.

2.2.2. Acquisition by Google

Google acquired Android Inc. on August 17, 2005, making Android Inc. a wholly owned subsidiary of Google. Key employees of Android Inc., including Andy Rubin, Rich Miner and Chris White, stayed at the company after the acquisition. Not much was known about Android Inc. at the time of the acquisition, but many assumed that Google was planning to enter the mobile phone market with this move.

2.2.3. Post-acquisition by Google

At Google, the team led by Rubin developed a mobile device platform powered by the Linux kernel. Google marketed the platform to handset makers and carriers on the promise of providing a flexible, upgradable system. Google had lined up a series of hardware component and software partners and signaled to carriers that it was open to various degrees of cooperation on their part.

Speculation about Google's intention to enter the mobile communications market continued to build through December 2006. Reports from the BBC and The Wall Street Journal noted that Google wanted its search and applications on mobile phones and it was working hard to deliver that. Print and online media outlets soon reported rumors that Google was developing a Google-branded handset. Some speculated that as Google was defining technical specifications, it was showing prototypes to cell phone manufacturers and network operators.

In September 2007, InformationWeek covered an Evacuee serve study reporting that Google had filed several patent applications in the area of mobile telephony.

2.2.4. Open Handset Alliance

On November 5, 2007, the Open Handset Alliance, a consortium of several companies which include Broadcom Corporation, Google, HTC, Intel, LG, Marvell Technology Group, Motorola, NVidia, Qualcomm, Samsung Electronics, Sprint Nextel, T-Mobile and Texas Instruments unveiled itself. The goal of the Open Handset Alliance is to develop open standards for mobile devices. On the same day, the Open Handset Alliance also unveiled their first product, Android, a mobile device platform built on the Linux kernel version 2.6.

On December 9, 2008, 14 new members joined, including ARM Holdings, Atheros Communications, Asustek Computer Inc., Garmin Ltd, Huawei Technologies, PacketVideo, Softbank, Sony Ericsson, Toshiba Corp, and Vodafone Group Plc. [10]

2.2.5. Android Open Source Project

The Android Open Source Project (AOSP) is led by Google, and is tasked with the maintenance and development of Android. According to the project "The goal of the Android Open Source Project is to create a successful real-world product that improves the mobile experience for end users." AOSP also maintains the Android Compatibility Program, defining an "Android compatible" device "as one that can run any application written by third-party developers using the Android SDK and NDK", to prevent incompatible Android implementations. The compatibility program is also optional and free of charge, with the Compatibility Test Suite also free and open-source.

2.2.6. Version history

Android has been updated frequently since the original release of "Astro", with each fixing bugs and adding new features. Each version is named in alphabetical order, with 1.5 "Cupcake" being the first named after a dessert and every update since following this naming convention. [10]

List of Android version names:

1. Cupcake
2. Donut
3. Eclair
4. Froyo
5. Gingerbread
6. Honeycomb
7. Ice Cream Sandwich
8. Android 4.2 Jelly Bean (API level 17)
9. Android 4.3 Jelly Bean (API level 18)
10. Android 4.4 Kit Kat (API level 19)
11. Android 5 Lollipop (API level 21)

2.3 Gingerbread

It refined the user interface, improved the soft keyboard and copy/paste features, improved gaming performance, SIP support (VoIP calls), and added support for Near Field Communication.

3.0 Honeycomb was a tablet-oriented release which supports larger screen devices and introduces many new user interface features, and supports multicore processors and hardware acceleration for graphics. The Honeycomb SDK has been released and the first device featuring this version, the Motorola Xoom tablet, went on sale in February 2011.

3.1 Honeycomb was announced at the 2011 Google I/O on 10 May 2011. One feature focuses on allowing Honeycomb devices to directly transfer content from USB devices.

3.2 Honeycomb released at July 15 2011, is "an incremental release that adds several new capabilities for users and developers". Highlights include optimization for a broader range of screen sizes; new "zoom-to-fill" screen compatibility mode; capability to load media files directly from the SD card; and an extended screen support API, providing developers with more precise control over the UI. Android 3.2 Honeycomb is the latest Android version that is available to tablets. 4.0.x Ice Cream Sandwich released at December 16, 2011, it's easy multitasking, rich notifications, customizable home screens, resizable widgets, and deep interactivity and adds powerful new ways of communicating and sharing.

4.1.x Jelly Bean released at July 9, 2012 Based on Linux kernel 3.0.31, Jelly Bean was an incremental update with the primary aim of improving the functionality and performance of the user interface. The performance improvement involved "Project Butter", which uses touch anticipation, triple buffering, and extended vsync timing and a fixed frame rate of 60 fps to create a fluid and "buttery-smooth" UI. Android 4.1 Jelly Bean was released to the Android Open Source Project on 9 July 2012, and the Nexus 7 tablet, the first device to run Jelly Bean.

4.2. x Jelly Bean released at November 13, 2012 its API level is 17.

4.3. x Jelly Bean released at July 24, 2013 API level is 18.

4.4 Kit Kat released at October 31, 2013 API level is 19.

2.4. Design and Architecture of Android

Android consists of a kernel based on the Linux kernel, with middleware, libraries and APIs written in C and application software running on an application framework which includes Java-compatible libraries based on Apache Harmony. Android uses the Dalvik virtual machine with just-in-time compilation to run Dalvikdex-code (Dalvik Executable), which is usually translated from Java byte code.

The main hardware platform for Android is the ARM architecture. There is support for x86 from the Android x86 projects and Google TV uses a special x86 version of Android.

2.5. Linux

Android's kernel is based on the Linux kernel and has further architecture changes by Google outside the typical Linux kernel development cycle. Android does not have a native X Window System nor does it support the full set of standard GNU libraries, and this makes it difficult to port existing Linux applications or libraries to Android. Certain features that Google contributed back to the Linux kernel, notably a power management feature called wake locks, were rejected by mainline kernel developers, partly because kernel maintainers felt that Google did not show any intent to maintain their own code. Even though Google announced in April 2010 that they would hire two employees to work with the Linux kernel community, Greg Kroah-Hartman, the current Linux kernel maintainer for the -stable branch, said in December 2010 that he was concerned that Google was no longer trying to get their code changes included in mainstream Linux. Some Google Android developers hinted that "the Android team was getting fed up with the process", because they were a small team and had more urgent work to do on Android. However, in September 2010, Linux kernel developer Rafael J. Wysocki added a patch that improved the mainline Linux wakeup events framework. He said that Android device drivers that use wake locks can now be easily merged into mainline Linux, but that Android's opportunistic Suspend features should not be included in the mainline kernel. In August 2011, Linus Torvalds said that "eventually Android and Linux would come back to a common kernel, but it will probably not be for four to five years". In December 2011, Greg Kroah-Hartman announced the start of the Android Mainlining Project, which aims to put some Android drivers, patches and features back into the Linux kernel, starting in Linux 3.3. further integration being expected for Linux Kernel 3.4. [12]

2.6 Architecture of Android

The following diagram shows the components of the Android operating system:

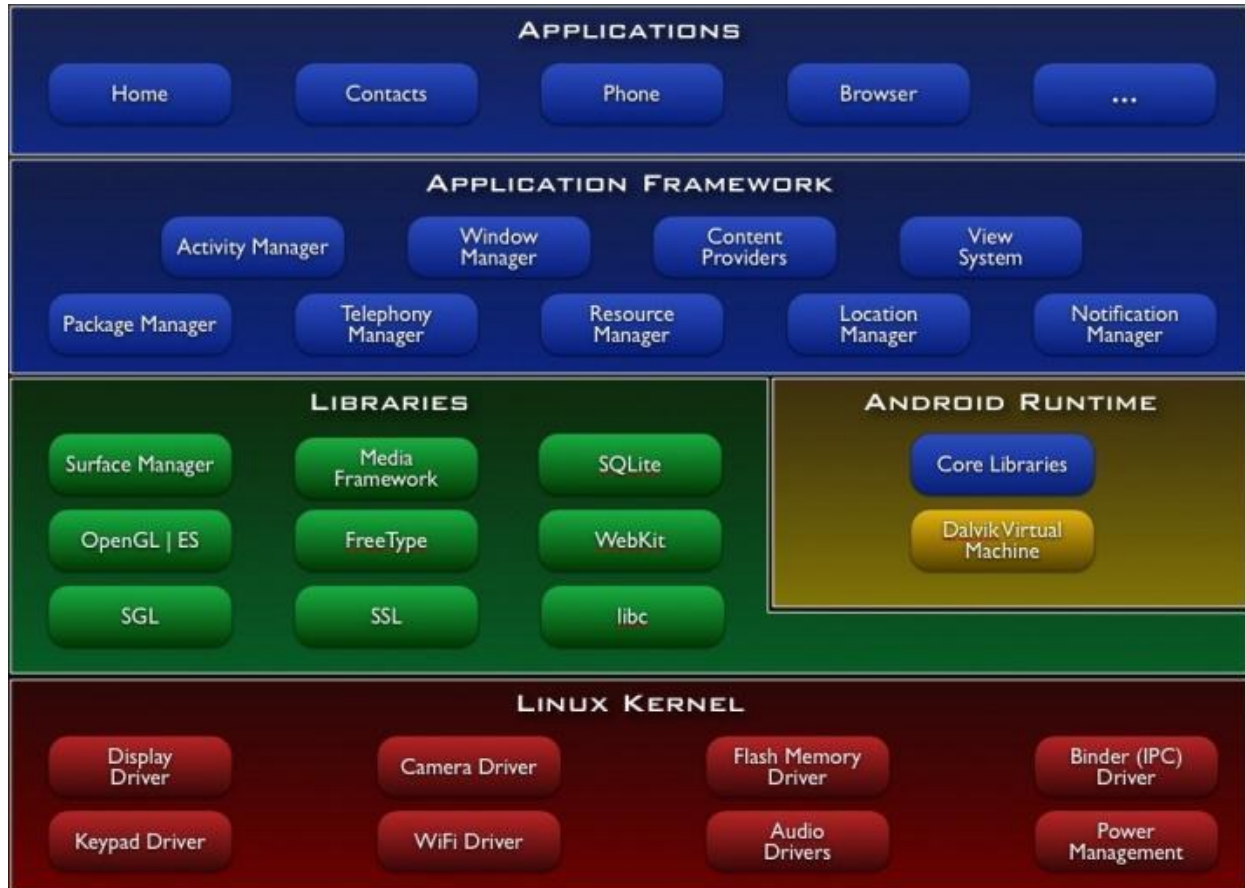


Figure 1: Android Architecture.

2.7 Application

Android will ship with a set of core applications including an email client, SMS program, calendar, maps, browser, contacts, and others. All applications are written using the Java programming language. Application Framework:

By providing an open development platform, Android offers developers the ability to build extremely rich and innovative applications. Developers are free to take advantage of the device hardware, access location information, run background services, set alarms, add notifications to the status bar, and much, much more.

Developers have full access to the same framework APIs used by the core applications. The application architecture is designed to simplify the reuse of components; any application can publish its capabilities and any other application may then make use of those capabilities (subject to security constraints enforced by the framework). This same mechanism allows components to be replaced by the user.

2.8. Android Runtime

Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language. Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently.

The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint. The VM is register-based, and runs classes compiled by a Java language compiler that have been transformed into the .dex format by the included "dx" tool.

The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

2.9. Linux Kernel

Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack. [12]

2.10 Features of Android

Features and current specifications of android are as follow:

□ Handset layouts

The platform is adaptable to larger, VGA, 2D graphics library, 3D graphics library based on OpenGL ES 2.0 specifications, and traditional Smartphone layouts.

□ Storage

SQLite, a lightweight relational database, is used for data storage purposes.

□ Connectivity

Android supports connectivity technologies including GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.

□ Messaging

SMS and MMS are available forms of messaging, including threaded text messaging and now Android Cloud to Device Messaging Framework (C2DM) is also a part of Android Push Messaging service.

□ **Multiple language support**

Android supports multiple human languages. The number of languages more than doubled for the platform 2.3 Gingerbread, 4.0.x Ice Cream Sandwich, 4.1.x Jelly Bean and 4.2.x Jelly Bean

□ **Web browser**

The web browser available in Android is based on the open-source WebKit layout engine, coupled with Chrome's V8 JavaScript engine. The browser scores a 95/100 on the Acid3 Test.

□ **Java support**

While most Android applications are written in Java, there is no Java Virtual Machine in the platform and Java byte code is not executed. Java classes are compiled into Dalvik executables and run on Dalvik, a specialized virtual machine designed specifically for Android and optimized for battery-powered mobile devices with limited memory and CPU. J2ME support can be provided via third-party applications.

□ **Media support**

Android supports the following audio/video/still media formats: WebM, H.263, H.264 (in 3GP or MP4 container), MPEG-4 SP, AMR, AMR-WB (in 3GP container), AAC, HE-AAC (in MP4 or 3GP container), MP3, MIDI, OggVorbis, FLAC, WAV, JPEG, PNG, GIF, BMP.

□ **Streaming media support**

RTP/RTSP streaming (3GPP PSS, ISMA), HTML progressive download (HTML5 <video> tag). Adobe Flash Streaming (RTMP) and HTTP Dynamic streaming are supported by the Flash plug-in. Apple HTTP Live Streaming is supported by RealPlayer for Mobile, and by the operating system in Android 3.0 (Honeycomb).

□ **Additional hardware support**

Android can use video/still cameras, touch screens, GPS, accelerometers, gyroscopes, magnetometers, dedicated gaming controls, proximity and pressure sensors, thermometers, accelerated 2D bit blitz (with hardware orientation, scaling, pixel format conversion) and accelerated 3D graphics.

□ **Multi-touch**

Android has native support for multi-touch which was initially made available in handsets such as the HTC One X. The feature was originally disabled at the kernel level (possibly to avoid infringing Apple's patents on touch-screen technology at the time). Google has since

released an update for the Nexus One and the Motorola Droid which enables multi-touch natively.

□ **Bluetooth**

Supports A2DP, AVRCP, sending files (OPP), accessing the phone book (PBAP), voice dialing and sending contacts between phones. Keyboard, mouse and joystick (HID) support is available in Android 3.1+, and in earlier versions through manufacturer customizations and third-party applications.

□ **Video calling**

Android does not support native video calling, but some handsets have a customized version of the operating system that supports it, either via the UMTS network (like the Samsung Galaxy S) or over IP. Video calling through Google Talk is available in Android 2.3.4 and later.

Gingerbread allows Nexus S to place Internet calls with a SIP account. This allows for enhanced VoIP dialing to other SIP accounts and even phone numbers. Skype 2.1 offers video calling in Android 2.3, 4.0.x, 4.1.x, 4.2.x, including front camera support.

□ **Multitasking**

Multitasking of applications is available.

□ **Voice based features**

Google search through voice has been available since initial release. Voice actions for calling, texting, navigation, etc. are supported on Android 2.2 onwards.

□ **Tethering**

Android supports tethering, which allows a phone to be used as a wireless/wired Wi-Fi hotspot. Before Android 2.2 this was supported by third-party applications or manufacturer customizations.

□ **Screen capture**

Android does not support screenshot capture as of 2011. This is supported by manufacturer and third-party customizations. Screen Capture is available in 4.0.x+ and through a PC connection using the DDMS developer's tool. [10]

2.11 Applications

Applications are usually developed in the Java language using the Android Software Development Kit, but other development tools are available, including a Native Development Kit for applications or extensions in C or C++, Google App Inventor, a visual environment for novice programmers and various cross platform mobile web applications frameworks.

Applications can be acquired by end-users either through a store such as Google Play or the Amazon Appstore, or by downloading and installing the application's APK files from a third-party site.

2.12. Google Play

Google Play is an online software store developed by Google for Android devices. An application program ("app") called "Play Store" is preinstalled on most Android devices and allows users to browse and download apps published by third-party developers, hosted on Google Play. As of February 2013, there were more than 800,000 apps available for Android, and the estimated number of applications downloaded from the Play Store exceeded 20 billion. The operating system itself is installed on 500 million total devices.

Only devices that comply with Google's compatibility requirements are allowed to preinstall and access the Play Store. The app filters the list of available applications to those that are compatible with the user's device, and developers may restrict their applications to particular carriers or countries for business reasons.

Google offers many free applications in the Play Store including Google Voice, Google Goggles, Gesture Search, Google Translate, Google Shopper, Listen and My Tracks. In August 2010, Google launched "Voice Actions for Android", which allows users to search, write messages, and initiate calls by voice.

2.13. Software Development Tools

In this portion I will describe about android app development tools and the way to development procedure.

2.13.1 Android SDK

The Android software development kit (SDK) includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, Windows XP or later. The officially supported integrated development environment (IDE) is Eclipse using the Android Development Tools (ADT) Plug-in, though developers may use any text editor to edit Java and XML files then use command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely). Enhancements to Android's SDK go hand in hand with the overall Android platform development. The SDK also supports older versions of the Android platform in case developers wish to target their applications at older devices. Development tools are downloadable components, so after one has downloaded the latest version and platform, older platforms and tools can also be downloaded for compatibility testing.

2.13.2 Native Development Kit

Libraries written in C and other languages can be compiled to ARM or x86 native code and installed using the Android Native Development Kit. Native classes can be called from Java code running under the Dalvik VM using the System. Load Library call, which is part of the standard Android Java classes.

Complete applications can be compiled and installed using traditional development tools. The ADB debugger gives a root shell under the Android Emulator which allows native ARM code or x 86 codes to be uploaded and executed. ARM or x 86 codes can be compiled using GCC on a standard PC. Running native code is complicated by the fact that Android uses a non-standard C library (libc, known as Bionic). The underlying graphics device is available as a frame buffer at /dev/graphics/fb0. The graphics library that Android uses to arbitrate and control access to this device is called the Skia Graphics Library (SGL), and it has been released under an open source license. Skia has backend for both win32 and UNIX, allowing the development of cross-platform applications, and it is the graphics engine underlying the Google Chrome web browser.

Unlike Java App development based on the Eclipse IDE, the NDK is based on command-line tools and requires invoking them manually to build, deploy and debug the apps. Several third-party tools allow integrating the NDK into Eclipse and Visual Studio.

2.13.3 Android Open Accessory Development Kit

The Android 3.1 platform (also back ported to Android 2.3.4) introduces Android Open Accessory support, which allows external USB hardware (an Android USB accessory) to interact with an Android-powered device in a special "accessory" mode.

When an Android-powered device is in accessory mode, the connected accessory acts as the USB host (powers the bus and enumerates devices) and the Android-powered device acts as the USB device. Android USB accessories.

2.13.4 App Inventor for Android

On 12 July 2010, Google announced the availability of App Inventor for Android, a Web-based visual development environment for novice programmers, based on MIT's Open Blocks Java library and providing access to Android devices' GPS, accelerometer and orientation data, phone functions, text messaging, speech-to-text conversion, contact data, persistent storage, and Web services, initially including Amazon and Twitter. "We could only have done this because Android's architecture is so open," said the project director, MIT's Hal Abelson. Under development for over a year, the block-editing tool has been taught to non-majors in computer science at Harvard, MIT, Wellesley, Trinity College (Hartford,) and the University of San Francisco, where Professor David Wolber developed an introductory computer science course and tutorial book for non-computer science students based on App Inventor for Android.

Chapter 3

Introduction to Unity Game Engine

3.1 Introduction:

Unity3D is a powerful cross-platform 3D engine and a user-friendly development environment. Learn how Unity3D can help you create games in this article! for Unity has released the latest version of its cross-platform game engine. At the 2015 Game Developers Conference, CEO John Riccitiello announced a final version of Unity 5, which is now available on the company's site. Unity 5 was announced in mid-2014 and released in beta last October, and it offers some notable updates over the past iteration, including upgraded animation and physics, a new lighting system, and more audio options. As one might expect, founder and CTO Joachim Ante calls it a "massive release" that touches just about every aspect of the engine. Besides promised improvements in graphics and efficiency, Unity 5 now supports a newer version of the PhysX physics engine, and there are more tools mixing audio. Unity is a cross-platform game engine developed by Unity Technologies[2] and used to develop video games for PC, consoles, mobile devices and websites. First announced only for OS X, at Apple's Worldwide Developers Conference in 2005, it has since been extended to target more than fifteen platforms.[3][4] It is now the default software development kit (SDK) for the Wii U.[5]

Unity Pro is available for a fee and Unity Personal has no fee; it is available for any use to individuals or companies with less than US\$100,000 of annual gross revenue.[6][7] On March 3, 2015 with the release of Unity 5.0, Unity Technologies made the complete engine available for free including all features, less source code and premium support. Unity is noted for an ability to target games to multiple platforms.

3.2 Unity's free and pro versions are Converging:

Unity started as an attempt to democratize game development; it's not the most powerful engine on the market, but it's flexible and offers broad support for desktop computers, mobile platforms, and major consoles. Riccitiello says Unity 5 currently supports 21 platforms, including iOS, Android, Windows, BlackBerry, all major consoles, and the Samsung Gear VR. It's also added support for the browser-based WebGL platform, which makes games very broadly available. Unity's tech demos still don't have the breathtaking graphical fidelity of a top-notch AAA game, because they're not supposed to. But at GDC, we saw a remastered version of high-gloss iPad game *Republique*, focused heavily on the lighting — it's definitely a notable improvement, with far softer and more natural shadows.

Unity has been particularly conspicuous in one new field: virtual reality. Oculus founder Palmer Luckey came on stage to talk about his work with Unity, including full integration with Unity 5. This feature is in alpha currently, but Luckey says it'll be available to everyone as a beta in around a month.

Besides the updates themselves, one of the big changes to Unity is that it's closing the gap between big studios and personal users. The engine comes in two versions, Pro and Personal: the latter is free, and the former costs \$75 a month or \$1,500 for a perpetual license. Older versions of Unity stripped some options out of the free version, but Riccitiello has promised nearly the same feature set for both in Unity 5, with the personal license restricted to people and studios with less than \$100,000 in revenue or funding. Unity and its competitor Epic have been in a price war for the past year, and earlier this week, Epic announced that it was offering Unreal Engine 4 free for all developers. The catch is that Epic relies on royalties; after a game makes \$3,000 in sales, Epic will take 5 percent a quarter. Unity has derided this model. "Free plus five percent of your gross isn't free," said one developer on stage. "That's millions of dollars."

Where Epic has worked on creating state-of-the-art graphics, Unity started with the goal of making game development universally accessible. Since then, they've been moving towards each other. And Unity 5 is a long-awaited step towards that future.

3.3 Features:

With an emphasis on portability, the engine targets the following APIs: Direct3D on Windows and Xbox 360; OpenGL on Mac and Windows; OpenGL ES on Android and iOS; and proprietary APIs on video game consoles. Unity allows specification of texture compression and resolution settings for each platform the game engine supports,[3] and provides support for bump mapping, reflection mapping, parallax mapping, screen space ambient occlusion (SSAO), dynamic shadows using shadow maps, render-to-texture and full-screen post-processing effects.[8] Unity's graphics engine's platform diversity can provide a shader with multiple variants and a declarative fallback specification, allowing Unity to detect the best variant for the current video hardware; and if none are compatible, fall back to an alternative shader that may sacrifice features for performance.[9]

The game engine's scripting is built on Mono, the open-source implementation of the .NET Framework.^[10] Programmers can use UnityScript (a custom language with ECMAScript-inspired syntax, referred to as JavaScript by the software),[11][12] C#, or Boo (which has a Python-inspired syntax).[13]

Unity is notable for its ability to target games to multiple platforms. Within a project, developers have control over delivery to mobile devices, web browsers, desktops, and consoles.^[3] Supported platforms include BlackBerry 10, Windows Phone 8, Windows, OS X, Android, iOS, Unity Web Player (including Facebook[14]), PlayStation 3, PlayStation 4, PlayStation Vita, Xbox 360, Xbox One, Wii U, Nintendo 3DS line[15][16][17] and Wii. It includes an asset server and Nvidia's PhysX physics engine. Unity Web Player is a browser plugin that is supported in Windows and OS X only.[18] Unity is the default software development kit (SDK) for Nintendo's Wii U video game console platform, with a free copy included by Nintendo with each Wii U developer license. Unity Technologies calls this bundling of a third-party SDK an "industry first".[5][19]

3.4 Versions:

3.4.1 Unity 1

Unity 1.0 was announced at the Apple Worldwide Developers Conference in 2005 and was released on June 8, 2005. The initial release was limited to OS X, both for authoring and game publishing. Features at the time included a shader-oriented OpenGL renderer, physics powered by the Novodex (now PhysX) physics engine, audio support and C# scripting support.[20] The package was available in both a low-priced 'Indie' edition, and a higher-priced 'Pro' edition which included some extra features.

Over subsequent 1.x versions, significant additions included support for building games to run on Windows, both as standalone applications and also in browsers via a browser plugin.[21]

3.4.2 Unity 2

Unity 2.0 was released on October 11, 2007, during the first annual Unite conference.[22] Key additions to the feature set included a terrain engine, networking system (based on RakNet), real-time dynamic shadow rendering, and a system for building game UI. This release also saw the introduction of Unity Asset Server, an add-on product intended to allow teams of developers to share project assets more easily.

On October 4, 2008, the iPhone publishing add-on was announced. This allowed developers to author their games on Mac and publish to iPhone.[23]

In Unity 2.5, released March 19, 2009, support for authoring games on Windows was released.[24]

At the October 2009 Unite conference, it was announced that Unity Technologies would no longer charge for the 'indie' edition of Unity, but would instead make it freely available.[25]

3.4.3 Unity 3

Unity 3.0 was released on October 4, 2010. This brought integrated lightmapping support using Illuminate Labs' Beasttechnology, occlusion culling powered by Umbra, realtime audio processing effects, and support for C# 3.5. The release also included a 'preview quality' release of support for Android game publishing, sold as a separate add-on; Android support was officially declared 'released' on March 1, 2011.

On November 10, 2010, the Unity Asset Store was launched as an online marketplace for Unity users to sell project assets - artwork, code systems, audio, etc. - to each other.[26]

Unity 3.4 introduced integrated support for Allegorithmic's "Substance" procedural material system.

Unity 3.5, released February 14, 2012, was a notable release for bringing several new features quite late in the version lifecycle: a new particle system named "Shuriken," an integrated pathfinding and navigation framework, level-of-detail management for 3D models, high-dynamic-range rendering, new global illumination features, and a rewrite of the occlusion culling support. This release also saw the introduction of preview support for both Adobe Flash and Google Native Client as publishing platforms.[27]

3.4.4 Unity 4

Unity 4.0 was officially released on November 13, 2012. Major new features include new 'Mecanim' animation system, DirectX 11 support and real-time shadows on mobile platforms.[28]

With the release of Unity 4.0, the company announced a shift towards a release cycle which would see versions be released with fewer features but at a faster rate.[29] As such, subsequent releases in the 4.X line delivered new features as follows:

- Unity 4.1, released March 13, 2013: Memory profiling, support for AirPlay on iOS, and a slew of smaller feature updates for Mecanim and shader editing.
- Unity 4.2, released July 22, 2013: Support for Windows Phone 8, Windows Store and BlackBerry as publishing platforms; OpenGL ES 3.0 support for mobile platforms; integrated version control support for Perforce; and the ability to cancel the build process once started. (The last of these got a round of applause when it was announced at 2013's Unite Nordic).
- Unity 4.3, released November 12, 2013: a new 2D framework, including both 2D rendering support and a 2D physics engine (powered by Box2D).
- Unity 4.5, released May 27, 2014: no significant new features were introduced, as instead this release focused on fixing bugs, reporting more than 450 fixes in the release.
- Unity 4.6, released November 26, 2014: a new UI Framework. Also, version 4.6.2, released on January 29, 2015, added support for 64-bit applications on iOS.

On May 21, 2013, CEO David Helgason announced that the 'basic' editions of the iPhone and Android mobile add-ons would now be available for free from Unity 4.2 onwards.

3.4.5 Unity 5

Unity 5.0 was released for free on March 3, 2015, adding the much anticipated real-time global illumination based on the Geomerics Enlighten technology. Other major changes include physically-based shaders, HDR sky-boxes, reflection probes, a new audio mixer with effects and enhanced animator workflows.

Unity's Cloud Build system was introduced (for \$25/month for non-pro users) as well as 'Game Performance Reporting' and the beta 'Game Analytics' (also \$25/month for non-pro users) which logs players usage and performance on released games, something that many developers found hard to implement in Unity 4.x. Previously, a game developer needed to code support for player logging directly into their game engine.

Smaller additions include: A 64-bit editor to handle large projects, iOS 64-bit support, new deferred rendering, graphics command buffers, improved linear lighting, HDR, skybox and cubemap workflows, improved job scheduling system, a new 'CPU Timeline Profiler' lets you see and investigate multicore usage, improved NavMesh pathfindingsystem, inth.

Up until Unity 5.0 the engine was using a fairly outdated version of Nvidia's PhysX physics middleware. Unity 5.0 included version 3.3, which is standard among Triple-A games.

Unity 5.0 brings support for Windows, OS X, Unity Webplayer, Android, iOS, BlackBerry 10, Windows Phone 8, Tizen, WebGL, PlayStation 3, PlayStation 4, PlayStation Vita, Wii

U, Nintendo 3DS line,[15][16][17] Xbox 360, Xbox One,Android TV, Samsung Smart TV, Oculus Rift, HTC Vive[30] and Gear VR.

On August 26, 2015 an experimental version was released for Ubuntu Linux with an unofficial platform-agnostic installer for most modern 64 bit Linux versions, and official support only for 64-bit Ubuntu 12.04 or newer.[31][32]

3.5 Games Using Unity.

Here I also add game list of 2014 and 2015 which is released.

Game List 2014:

- Angry Birds Epic
- Blitzkrieg 3
- Broforce
- CounterSpy
- Crawl
- Crossy Road
- The Dark Eye: Blackguards
- DreadOut
- Dreamfall Chapters: The Longest Journey
- Exogenesis: Perils of Rebirth
- Endless Legend
- The Forest
- Fract OSC
- Hearthstone: Heroes of Warcraft
- Isabelle Dance Studio
- Jazzpunk
- Jungle Rumble
- The Last Tinker: City of Colors
- Lifeless Planet^[2]
- The Long Dark
- Might & Magic X: Legacy
- Monument Valley
- Never Alone
- Oddworld: New 'n' Tasty!
- Plague Inc: Evolved
- realMyst: Masterpiece Edition
- Scrolls
- Shadow Blade
- Shadow Blade: Reload
- Tesla Effect: A Tex Murphy Adventure
- Threes
- Victory at Sea (game)

Game list 2015:

- Angry Birds 2
- Armikrog
- Cities: Skylines
- Defunct
- Dex
- Earthlock: Festival of Magic
- Fallout Shelter
- Firefly Online
- Ghost of a Tale
- Grow Home
- HuniePop
- Inside
- Kerbal Space Program
- Mobius Final Fantasy
- Ori and the Blind Forest
- Pillars of Eternity
- Project Phoenix
- RollerCoaster Tycoon World
- Shadowrun: Hong Kong
- Shroud of the Avatar: Forsaken Virtues
- SpongeBob HeroPants
- Superhot
- Sylvio
- Torment: Tides of Numenera
- Verdun
- War for the Overworld

3.6 Unity-Chan:

In Japan, Unity has a moe mascot character named Unity-chan (ユニティちゃん Yuniti-chan[?]), real name Kohaku Otori (大鳥 こはく Ōtori Kohaku[?]) (voiced by Asuka Kakumoto). The company allows the use of Unity-chan and related characters for use in secondary projects under certain licenses.^[33] For example, Unity-chan appears as a playable character in Runbow.^[34]

3.7 Reception.

In 2012, VentureBeat said, "Few companies have contributed as much to the flowing of independently produced games as Unity Technologies." [5]

For the Apple Design Awards at the 2006 WWDC trade show, Apple, Inc. named Unity as the runner up for its Best Use of Mac OS X Graphics category, a year after Unity's launch at the same trade show. [35] Unity Technologies says this is the first time a game design tool has ever been nominated for this award. [36] A May 2012 survey by Game Developer magazine indicated Unity as its top game engine for mobile platforms. [37] In July 2014, Unity won the "Best Engine" award at the UK's annual Develop Industry Excellence Awards. [38]

3.8 What is Unity3D?

Unity3D is a powerful cross-platform 3D engine and a user friendly development environment. Easy enough for the beginner and powerful enough for the expert; Unity should interest anybody who wants to easily create 3D games and applications for mobile, desktop, the web, and consoles.

3.9 Cost.

Indie developers rejoice, Unity is free! Well, almost. There is a Pro edition that comes with more features and tools, but will set you back \$1,500. Considering the feature set and how permissive the Unity publishing license is, this price is actually very reasonable. However, the free version will let you get your feet wet, build complete games, and even publish them to the desktop and the web without paying a cent! The only caveat is that games published under the free edition will have a small Unity watermark.

Anything built in Unity will work exactly the same way in Unity Pro. This means you can choose to upgrade at any point if you need the additional features, or want to publish to more platforms such as iOS and Android. There's also a 30 day Pro trial you can sign up for in order to test drive all the extra features!

A complete feature comparison between Unity and Unity Pro can be found here:

<http://unity3d.com/unity/licenses>

3.10 Installation

Installation is a painless two step process. First, download and run the Unity installer.

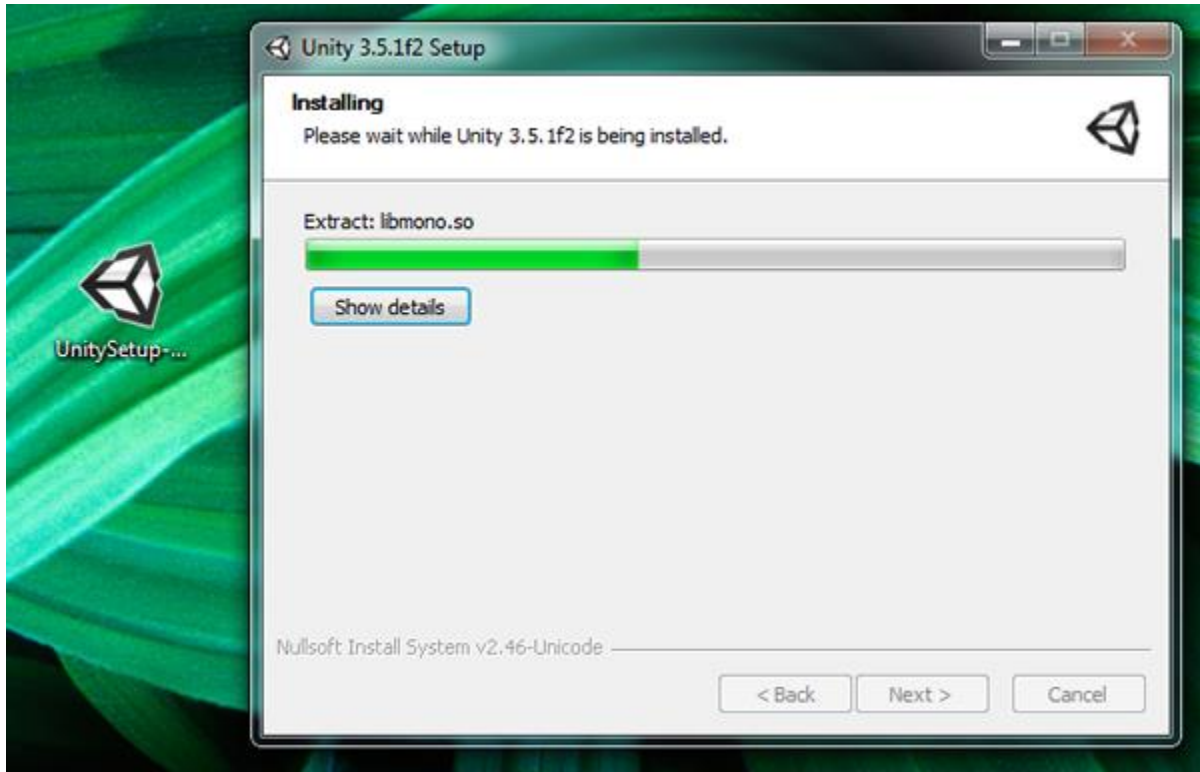


Figure 2: Installation Process Wizard.

Second, when you start Unity for the first time, it will open a web browser and prompt you to register using your email address. It will let you choose which version you want to run. You can select either the free version or a Pro trial that will fallback to the free version after 30 days.

3.11 The Application

The Unity application is a complete 3D environment, suitable for laying out levels, creating menus, doing animation, writing scripts, and organizing projects. The user interface is well organized and the panels can be fully customized by dragging and dropping.

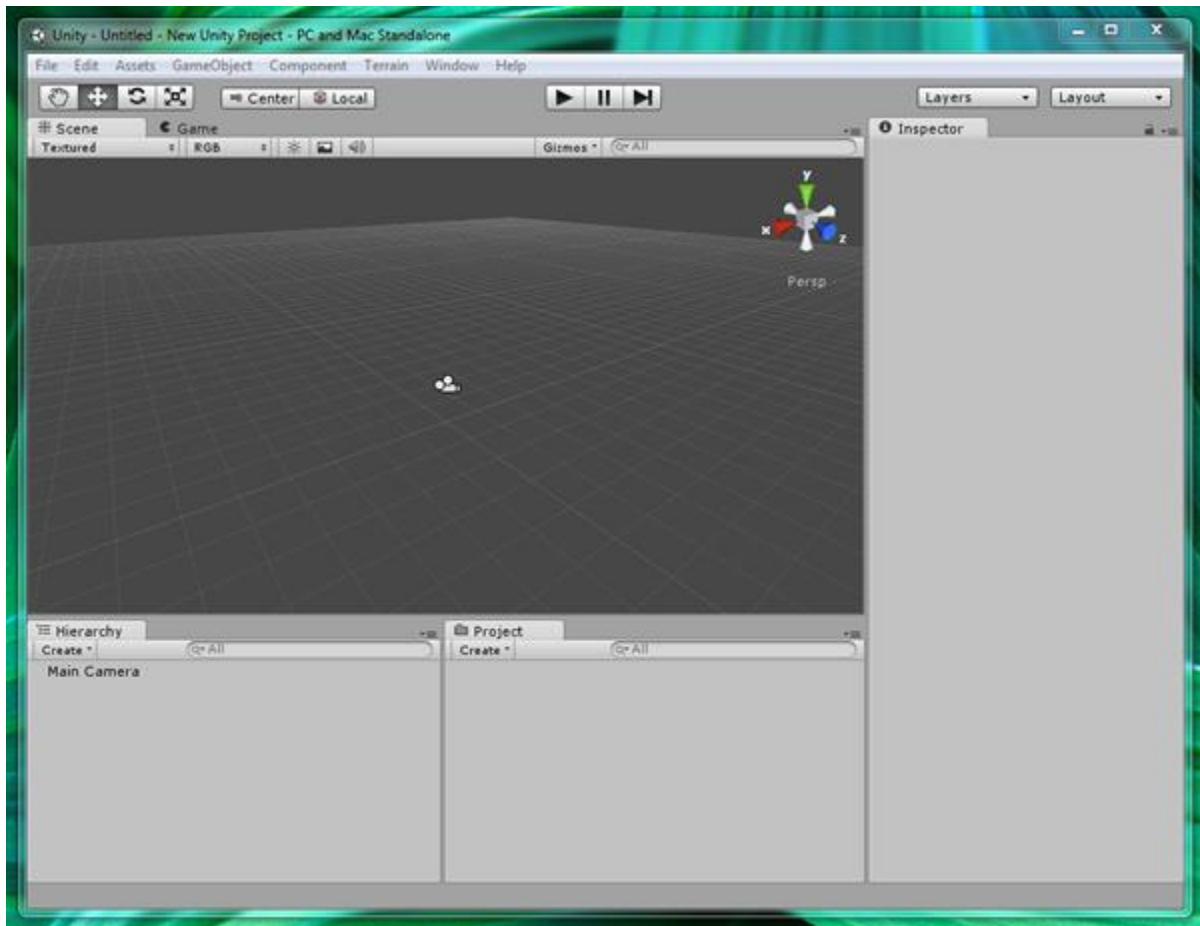


Figure 3: Project work Area.

The Project panel is where all the assets within a project are stored. When assets are imported, they will first appear here.

The hierarchy panel is where assets are organized in a scene. Assets from the Project panel can be dragged into the Hierarchy panel to add them to the current scene.

The Inspector panel lets you inspect and adjust all the attributes of a selected asset. Everything from its position and rotation, to whether it's affected by gravity or able to cast a shadow.

The Scene panel is a 3D viewport where you can physically arrange assets by moving them around in 3D space. You can navigate the viewport by panning, rotating, and zooming the view. If you've used Maya at all, you should find these hotkeys familiar:



When it comes to running your game, it couldn't be simpler. Just press the play button. To stop it, press the play button again. You can even pause your game during play to inspect your scene.



Figure 4: Game Modes.

3.12 Unity Projects

A Unity project is an ordinary folder containing every resource that belongs to your game. Creating a new project is a straightforward affair.

1. Click File > New Project
2. Click the Create New Project tab
3. Browse to a suitable folder
4. Click Create

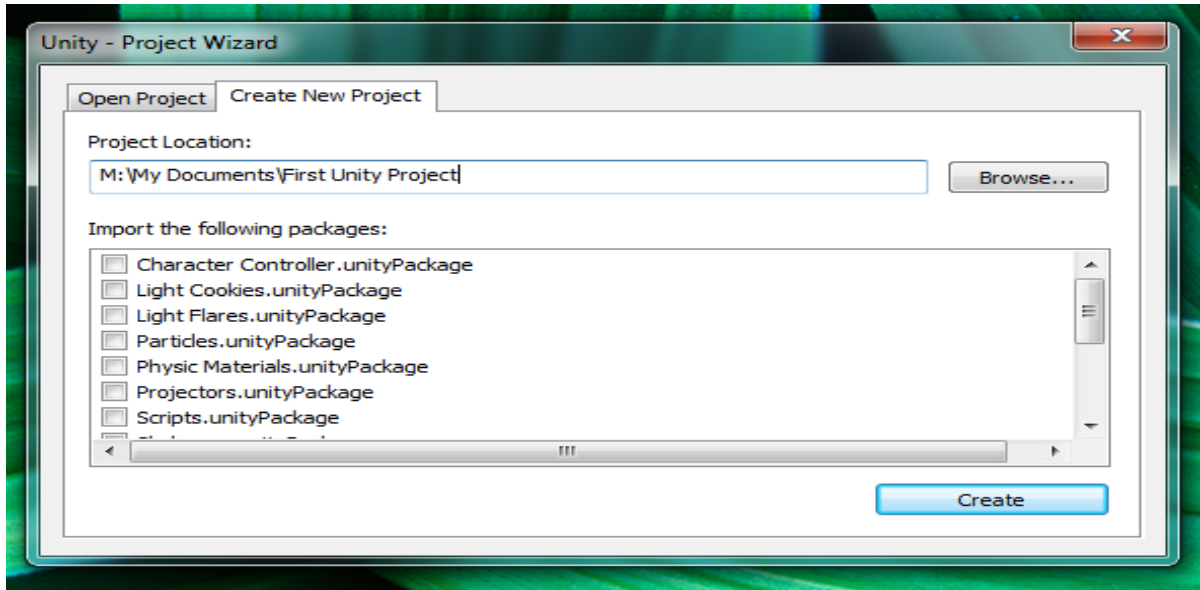
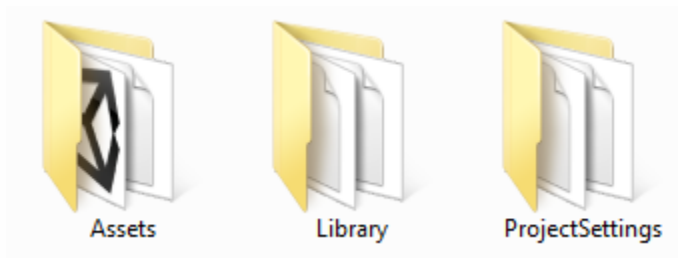


Figure 5: Project Wizard.

The result is a project folder containing subfolders named Assets, Library, and ProjectSettings.



3.12.1 Assets

Assets are any resource your game uses. These include 3D models, materials, textures, audio, scripts, and fonts, to name a few. Other than a few simple objects such as cubes and spheres, Unity can't actually create most of these assets. Instead, they must be created externally using 3D modeling applications and painting tools and then imported into Unity.

Thankfully, Unity's asset importing is robust and intelligent. Traditionally, 3D game engines have usually been finicky things and are very particular about what files you give them, forcing developers to carefully convert all their files. Not Unity. It will accept all popular 3D file formats including Maya, 3D Studio Max, Blender and FilmBox with all the rigging, materials and textures intact. Unity also supports all common image file formats, including PNG, JPEG, TIFF and even layered PSD files directly from Photoshop. When it comes to audio, Unity supports WAV and AIF, ideal for sound effects, and MP3 and OGG for music.

A complete list of all the formats Unity can import can be found here:

<http://unity3d.com/unity/editor/importing>

Let's import an asset so we have something to work with:

1. Download boxboy.zip
2. Unzip it to your desktop
3. Drag the boxboy folder (containing boxboy.fbx and texture.png) from your desktop into the Project panel
4. Drag the boxboy asset from the Project panel into the Hierarchy panel
5. Select boxboy in the Hierarchy panel
6. Press F to focus the Scene panel on the boxboy

Note: Unity has an Asset Store where you can purchase 3D models, characters, textures, sound effects, music, tools, and even scripts. The Unity Asset Store has quickly become an invaluable resource for game developers and a money making venture for artists and tool developers.

3.12.2 Scenes

Scenes are where you can drag in project assets and arrange them to make levels and game screens. The Hierarchy panel represents the contents of the current scene in a tree-like format. While the Scene panel is ideal for arranging your scene's assets in 3D space, the Hierarchy is where you'll spend most of your time actually organizing your scenes and keeping them tidy.

When you start a new project, Unity automatically creates a new scene for you. Scenes start out with nothing but a camera. If you were to run the game now, you won't see anything but the background color. To give us something to look at:

1. Drag the boxboy asset we imported from the Project panel into the Hierarchy panel

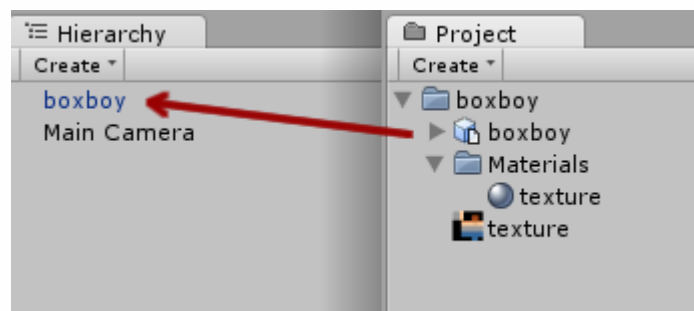


Figure 6: Hierarchy Panel Background.

2. Select the boxboy asset in the Hierarchy panel
3. In the Inspector, find the Transform component and adjust the position so that X, Y, and Z are all set to 0. This will ensure your asset is at the exact center of the 3D world.

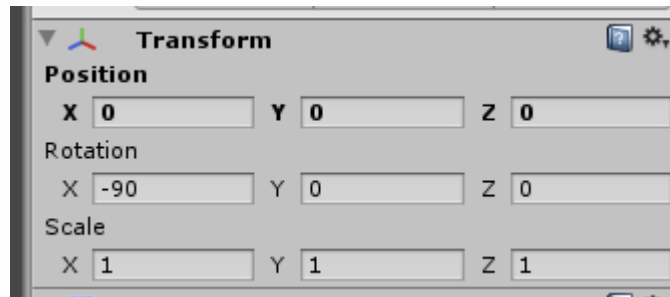


Figure 7: Transform Background .

4. The default camera position isn't very good, so let's give it a better angle. Select the camera, then reposition it using the move and rotate tools.

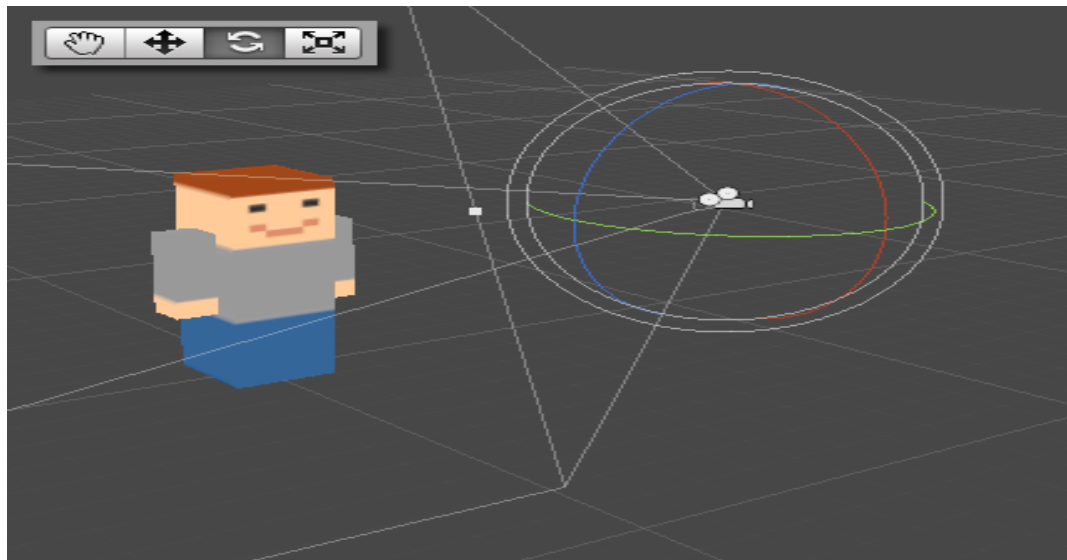


Figure 8: 2D/3D view Background.

Scenes are assets and should be saved in your project just like other assets. To save your scene:

1. Click File > Save Scene
2. Navigate to your project's Assets folder
3. Name your scene Main
4. Click Save

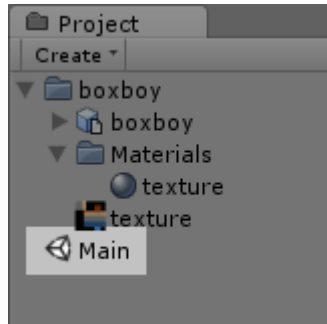


Figure 9: Project File Review.

3.13 Scripting

Scripts, known in Unity as behaviors, let you take assets in your scene and make them interactive. Multiple scripts can be attached to a single object, allowing for easy code reuse. Unity supports three different programming languages; Unity Script, C#, and Boo. UnityScript is similar to JavaScript and Action Script, C# is similar to Java, and Boo is similar to Python. Depending on your background you may feel more comfortable with one or the other.

Let's create a C# script:

1. Click Assets > Create > New C# Script
2. Rename the new script in the Project panel to PlayerScript
3. Double click the script to open it in MonoDevelop

The script should look just like this:

```
01 using UnityEngine;
02 using System.Collections;
03
04 public class PlayerScript : MonoBehaviour {
05     // Use this for initialization
06     void Start () {
07     }
08
09     // Update is called once per frame
10     void Update () {
```



```
11     }  
12 }
```

Note: C# class names must be the same as their file name and are case sensitive. Make sure your class name matches the file name exactly, excluding the file extension.

All scripts have a `start()` method and an `update()` method. The `start()` method is run once when the object is first created, while the `update()` method run once per frame. Our script needs to be constantly checking for arrow keys being pressed, so we'll add the following code to the `update()` method.

```
1 void Update () {  
2     float horizontal = Input.GetAxis("Horizontal");  
3     float vertical = Input.GetAxis("Vertical");  
4     transform.Translate(horizontal, vertical, 0);  
5 }
```

Now that our script is done, we need to assign it to our asset. Naturally, Unity makes this a simple affair:

1. Drag the script onto the boxboy asset in your scene

With the script assigned to our boxboy asset, we can run the game and move BoxBoy around by pressing the arrow keys.

Advertisement

3.14 Publishing

Unity is able to publish to Windows, OS X, and the web via the Unity Web Player. The Web Player is a browser plug in that works in all major browsers and offers the same performance available on the desktop.

You can download the Unity Web Player here:

<http://unity3d.com/webplayer/>

Not surprisingly, Unity Pro can publish to even more platforms, including iOS, Android, Wii, Xbox 360, Play station 3 and even a Flash version of the Web Player.

To publish our game for the Web Player:

1. Click File > Build & Run
2. Select Web Player from the list
3. Click Build And Run

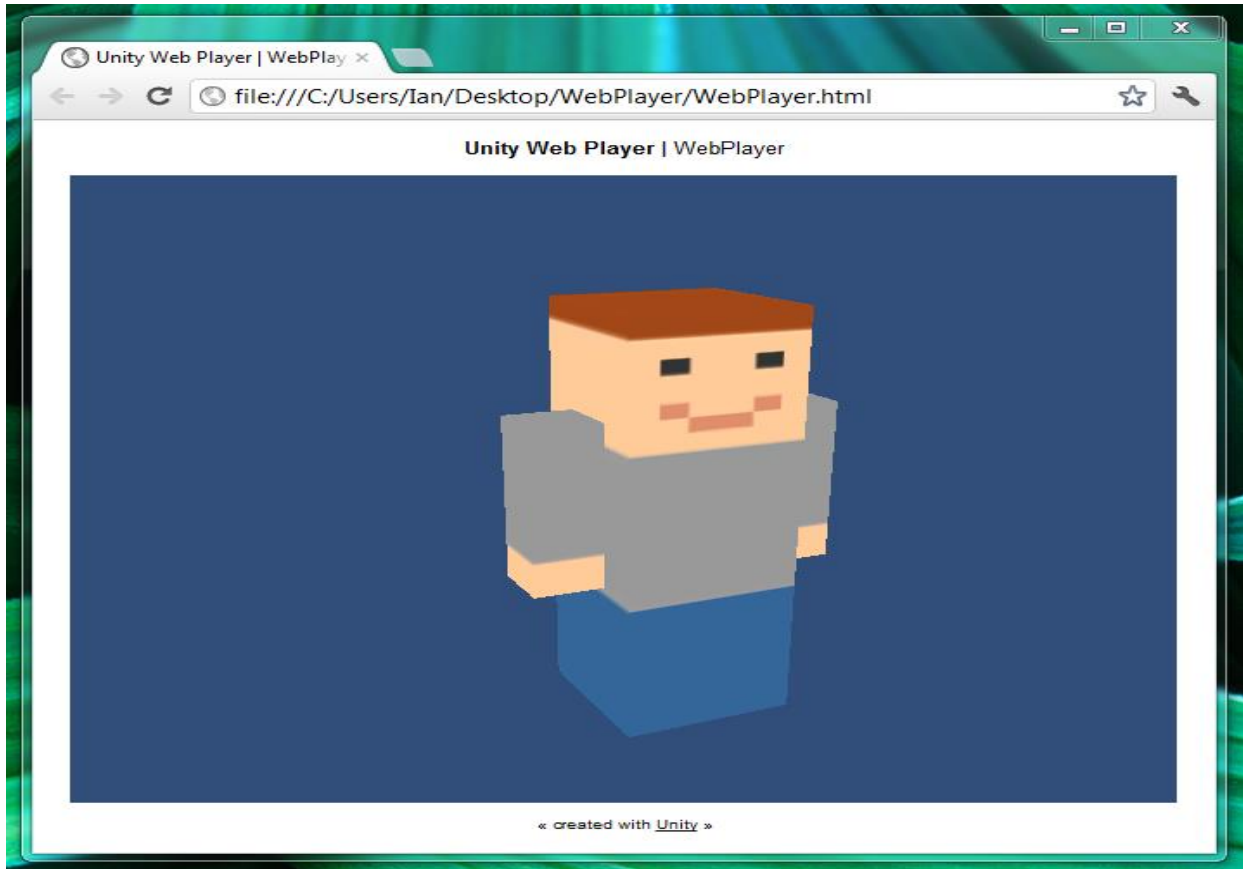


Figure 10: Unity Web Player Background.

3.15 Conclusion

This article barely scratches the surface of what is possible with Unity. If this introduction has whet your appetite for more 3D game development, be sure to check out the following resources:

- Unity Answers
- Unity Scripting Reference

Chapter 4

Implementation

4.1 Design

Our application is a game for those devices which are run by Unity5 game engine software which is android based platform. The full project contains:

- A full game world which is played by the user.
- The ball can move anywhere with touching bat.
- Two player have two bat but it is fixed by the midline
- If the ball touches the opposite sides player's wall then it counts the point.
- Game score show the both player of their upper side of the screen.
- Score when the ball touches opposite side player's wall.
- Game over when the time is over.
- Pressing the replay button they can play again.

4.2 Implementation Procedure

Creating a fair playing field for android development:

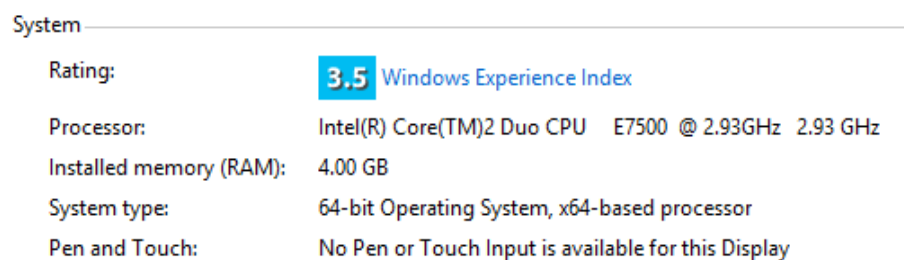
Here are some simple pre-requisites one must have to develop an android app.

Developer Requirement:

- Advanced knowledge of java script.
- Basic knowledge of XML

4.2.1 Hardware Requirement:

The PC where develop the application must be a fast one. I used Intel core (TM)2 Duo Cpu machine clocked at E7500 @ 2.93 GHz with 4GB ram clocked @ 1420Mhz. A big monitor or two is also helpful. During debugging it really eases the pain.



System	
Rating:	3.5 Windows Experience Index
Processor:	Intel(R) Core(TM)2 Duo CPU E7500 @ 2.93GHz 2.93 GHz
Installed memory (RAM):	4.00 GB
System type:	64-bit Operating System, x64-based processor
Pen and Touch:	No Pen or Touch Input is available for this Display

Figure 11: Computer properties

4.2.2 Android Game Development Environment

For developing application I had to create android development environment. Google basically supports the "Unity Game Engine" software for the maximum SDK of the smart phone OS system. But there are also other IDEs. I have used Unity5 game engine for my ping pong game application development. There have many other game engine software or application development software like as Eclipse, android stdio, Intel lite Idea etc. But I choose Unity 5 game engine because I wanted such IDE in which I could write java script code and at the same time using the same IDE I could work on GUI for android. Unity5 generates most of the important codes automatically by clicking the function or physics method and it also provides both of these features. So I use Unity5 Game Engine Software for my gaming project development.

4.3. Android Game

Android is a mobile operating system (OS) based on the Linux kernel and currently developed by Google. With a user interface based on direct manipulation, Android is designed primarily for touch screen mobile devices such as smart phones and tablet computers, with specialized user interfaces for televisions (Android TV), cars (Android Auto), and wrist watches (Android Wear). The OS uses touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, and a virtual keyboard. Despite being primarily designed for touch screen input, it has also been used in game consoles, digital cameras, regular PCs, and other electronics.

4.3.1 Feature

The features of this game are:

- Ping pong game which is played by the user.
- Touching the ball both of the player's target of their opposite side.
- Game score show the both player of their upper side of the screen.
- The collision between ball and opposite side's wall.
- Scoring when ball touch the opposite sides wall.

4.3.2 Game Screen

In this game screen user or player can easily sound off/on by touching the sound icon, then they can start the game pressing the start button.

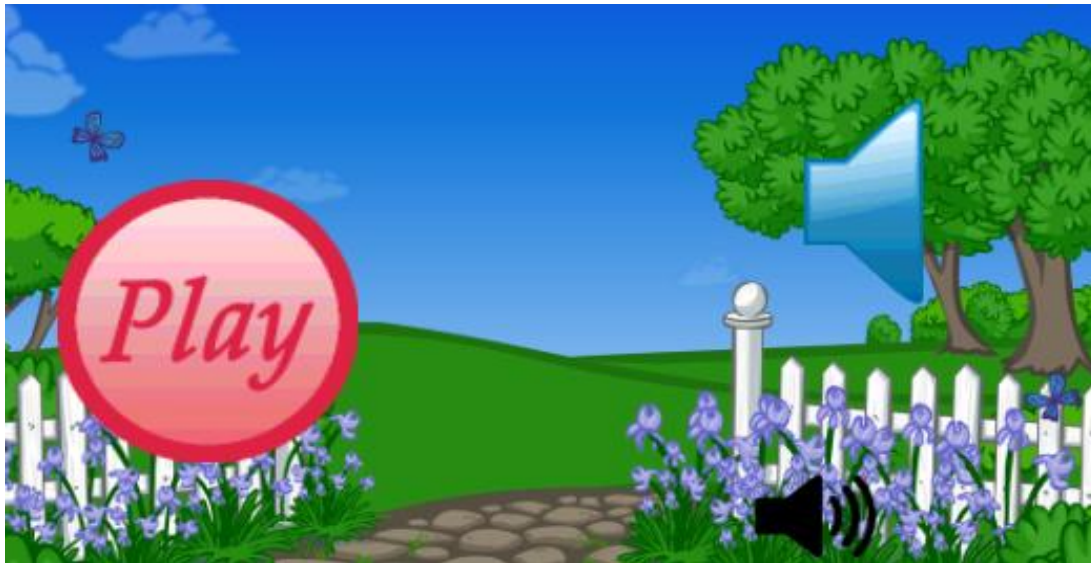


Figure 12: Game Screen

4.3.3 Background

After choosing the play button player can select the time, how many times they play the game.



Figure 13: Time selecting Background.

Game Start manu Image:

I tried to create a simple background which looks realistic and also comfortable for user.



Figure 14: Game start time Background.

4.3.4. Left Win Image

When the left sides player win the game the application show the message in the game background left win.

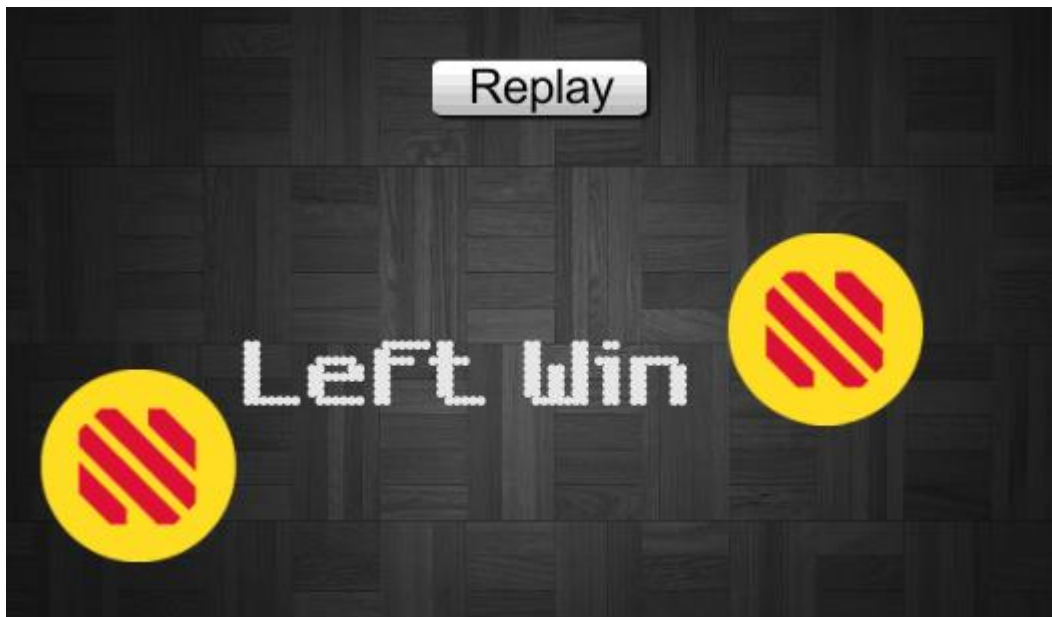


Figure 15: Left Win Message.

4.3.5. Right win image

When the right sides player wins the game the application show the message in the game background right win.

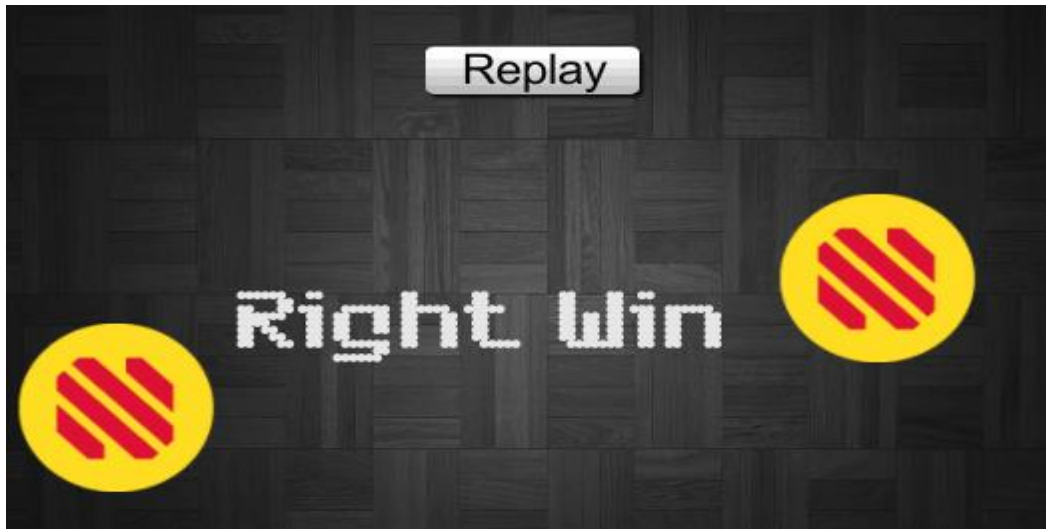


Figure 16: Right Win Message

4.3.6. Game tie image

When the left sides players and right side players score is same then the game the application show the message in the game background Game tie.



Figure 17: Game Tie Message.

4.3.7 Ball

The radius of the ball size is 0.23m. Here i also use only blue colors ball for the game. In ball logic I also use 2D physics reject body which can push power and also use 2D bounce which can push the ball along with the axis.



Figure 18: Ball.

- The ball has no accelerates due to gravity.
- Vertical velocity is never Attraction to the downwards.
- Ball always movement anywhere by touching the wall along with the axis or player push the ball touching their bat on that axis.

4.3.8. Bat

The radius of the bat size is 0.54m. Here i also use yellow and red colors combination bat for the game. In bat logic I also use 2D physics collider which can press power to the ball along with the axis.



Figure 19: Bat.

4.3.9 Scoring

In **this game** the score a point up or plus when the ball touch the opposite sides wall. Game score show the both player of their upper side of the screen. The collision between ball and opposite side's wall.



Figure 20: Scoring image.

4.4 Hardware Specification

I have used following devices for development, debugging and testing purpose

- Sony Xperia Neo L
- Walton Primo N1
- Symphony W70
- Samsung Galaxy S-3
- Samsung Galaxy a5
- Samsung Galaxy S-6

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Unity 3 game engine is the best game engine of Android platform and it is the most popular game engine. This project has successfully demonstrated a mobile based application. Android games are developed in Unity5 Game Engine and use logics on JavaScript, but Unity Game Engine are not a Complete JavaScript implementation. Many of the packages that I have used for OpenGL and other graphics are included in the Android software development kit. Some. But most of the important 2D tools, physics and logics are available in Unity5 game engine for this game, because I have developed a 2-D game. With each release of new Android SDK, more and more packages become available, and older once may be removed. I think this game help me to develop an application and learn a lot about Unity game engine based on android platform. I have made my application keeping some space for further development. This game will inspire new developers to create new games with different dimension.

5.2 Future Work:

Unity3 android based game engine is an open source operating system. So this platform also encourage us to develop android application and give us change to build our acknowledgement to modify our thinking level and keep sharp our brain to face the challenging new world.

Some future modifications I want to make on this game application, those are given below:

- ✓ I want to add many types of attractive types of new balls for the new levels.
- ✓ Modify new game levels which was locked but every step open the next new game level.
- ✓ I want to add many levels in this game (Easy, Medium, and Hard).
- ✓ I want to make an iOS version of this application. Because besides android, iPhone is also widely used smart phone across the world

Android Reference:

- [1] S.V. Rice, F.R. Jenkins, T.A. Nartker, The Fourth Annual Test of OCR Accuracy, Technical Report 95-03, Information Science Research Institute, University of Nevada, Las Vegas, July 1995.
- [2] R.W. Smith, The Extraction and Recognition of Text from Multimedia Document Images, PhD Thesis, University of Bristol, November 1987.
- [3] R. Smith, "A Simple and Efficient Skew Detection Algorithm via Text Row Accumulation", Proc. of the 3rd Int. Conf. on Document Analysis and Recognition (Vol. 2), IEEE 1995, pp. 1145-1148.
- [4] [http://en.wikipedia.org/wiki/Tesseract_\(software\)](http://en.wikipedia.org/wiki/Tesseract_(software)) [15-7-2015].
- [5] A. B. Cambra, A. C. Murillo: Towards robust and efficient text sign reading from a mobile phone. IEEE International Conference on Computer Vision Workshops, ICCV 2011 Workshops, Barcelona, Spain, November 6-13,2011, pp.64-71, ISBN 978-1-4673-0062-9 http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6130223 [4-5-2015]
- [6] Gautam Gupta . Making a Simple OCR Android App using Tesseract. <http://gaut.am/making-an-ocr-android-app-using-tesseract/> [10-7-15]
- [7] http://en.wikipedia.org/wiki/Optical_character_recognition [24-7-2015].
- [8] .Liaqat, Ahmad Gull. "Mobile Real-Time License Plate Recognition." (2011).
- [9] <http://venturebeat.com/2013/01/28/android-captured-almost-70-global-smartphone-market-share-in-2012-apple-just-under-20/> [08-04-2015]
- [10] http://en.wikipedia.org/wiki/Google_Play [21-04-2015]
- [11] [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system)) [5-05-2015]
- [12] <http://forum.xda-developers.com/showthread.php?t=1595487> [23-05-2015]
- [13] <http://developer.android.com/guide/components/fundamentals.html> [03-05-2015]
- [15] "Android's Google Play beats App Store with over 1 billion apps, now officially largest". Phonearena.com. Retrieved August 28, 2013.
- [16]"Developer Economics Q3 2013 analyst report". Retrieved July 2013.

- [17] Google shows off new version of Android, announces 1 billion active monthly users. Techspot. Retrieved June 30, 2014
- [18] Elgin, Ben (August 17, 2005). "Google Buys Android for Its Mobile Arsenal". Bloomberg Businessweek. Bloomberg. Archived from the original on February 24, 2011. Retrieved February 20, 2012. In what could be a key move in its nascent wireless strategy, Google (GOOG) has quietly acquired startup Android, Inc., ...
- [19] "Google's iron grip on Android: Controlling open source by any means necessary". Ars Technica. Retrieved December 8, 2013.
- [20] "Industry Leaders Announce Open Platform for Mobile Devices" (Press release).Open Handset Alliance. November 5, 2007. Retrieved February 17, 2012.
- [21] Brodtkin, Jon (November 5, 2012). "On its 5th birthday, 5 things we love about Android". Product News & Reviews. Ars Technica.
- [22] "Custom ROMs For Android Explained – Here Is Why You Want Them". August 20, 2012. Retrieved September 15, 2012.
- [23] Reardon, Marguerite (August 15, 2011). "Google just bought itself patent protection | Signal Strength – CNET News". News.cnet.com. Retrieved May 1, 2013.
- [24] Reardon, Marguerite (August 15, 2011). "Google just bought itself patent protection | Signal Strength – CNET News". News.cnet.com. Retrieved May 1, 2013.
- [25] Chris Welch (April 16, 2013). "Before it took over smartphones, Android was originally destined for cameras". The Verge. Retrieved May 1, 2013.

Unity Reference:

1. <http://blogs.unity3d.com/2015/08/26/unity-comes-to-linux-experimental-build-now-available/> Unity Comes to Linux: Experimental Build Now Available
 2. Riccitiello, John (October 23, 2014). John Riccitiello sets out to identify the engine of growth for Unity Technologies (interview). VentureBeat. Interview with Dean Takahashi. Retrieved January 18, 2015.
 3. "Unity - Multiplatform". Retrieved August 4, 2015.
 4. "How Unity3D Became a Game-Development Beast". Slashdot.org. Dice. June 3, 2013. Retrieved July 13, 2014.
 5. Helgason, David (November 2, 2012). Game developers, start your Unity 3D engines. GamesBeat. Interview with Dean Takahashi (VentureBeat). Retrieved July 13, 2014.
 6. "Unity Software License Agreement 4.x". Unity Technologies. July 23, 2013. Retrieved September 5, 2013.
 7. "Unity Game Engine Goes Free For iOS, Android And BlackBerry 10 Developers". TechCrunch. AOL. Retrieved January 18, 2015.
 8. "Using DirectX11 in Unity 4". Unity Technologies. Retrieved February 19, 2013.
 9. "Shaders". Unity Technologies. Retrieved February 19, 2013.
 10. "Companies using Mono". Retrieved January 18, 2015.
 11. UnityScript versus JavaScript. Unify Wiki. Retrieved 29. May 2014.
- Difference between UnityScript and JavaScript?. Unity Answers. Retrieved May 29, 2014.
12. "Using Scripts". Unity Technologies. Archived from the original on July 9, 2013. Retrieved February 19, 2013.
 13. "Unity technologies releases Facebook functionality update to unity 4". MarketWatch.com. March 26, 2013. Retrieved July 13, 2014.
 14. "UNITY TO SUPPORT NEW NINTENDO 3DS AND NEW NINTENDO 3DS XL". Unity Technologies. 2015-04-13. Retrieved 2015-05-06.
 15. P, Jenna (2015-04-13). "Unity Support is Heading to the New Nintendo 3DS". IGN. Retrieved 2015-05-06.
 16. "Unity Europe Announces New 3DS Details". NintendoWorldReport. 2015-08-13. Retrieved 2015-08-16.
 17. "Unity Web Player".
 18. McElroy, Griffin (August 20, 2013). "Unity for Wii U opens up GamePad hardware and more to developers". Polygon. Retrieved August 4, 2014.
 19. "Unity 1.0 Features". Unity Technologies. Retrieved February 8, 2015.
 20. "Release Archive". Unity Technologies. Retrieved February 8, 2015.
 21. "Unite 2007". Unity Technologies. Retrieved February 8, 2015.
 22. "Unity iPhone publishing announced!". Unity Technologies. Retrieved February 8, 2015.
 23. "Unity 2.5 for Mac and Windows now available!". Unity Technologies. Retrieved February 8, 2015.
 24. "A free Unity?". Unity Technologies. Retrieved February 8, 2015.
 25. "Unite 2010: Tech outfit introduces in-engine marketplace". Develop Online. Retrieved February 8, 2015.

26. "Unity 3.5 - What's New". Unity Technologies. Retrieved February 8, 2015.
27. "Unity 4.0 - What's New?". Unity Technologies. Retrieved February 8, 2015.
28. "Unity 4 is out!". Unity Technologies. Retrieved February 8, 2015.
29. "Unity and Unreal Engine Now Both Support HTC Vive Development". Road to VR. 2015-05-08. Retrieved 2015-08-29.
30. <http://blogs.unity3d.com/2015/08/26/unity-comes-to-linux-experimental-build-now-available/>
31. http://www.codeproject.com/News.aspx?ntag=19837497498935764&_z=9117240
32. http://unity-chan.com/contents/guideline_en/
33. http://www.nintendolife.com/news/2015/08/exclusive_check_out_runbows_complete_cast_of_colourful_crossover_characters
34. Smykill, Jeff (August 9, 2006). "Apple Design Award winners announced". ArsTechnica. Retrieved January 18, 2015.
35. "Unity Technologies Celebrates Five Years of Continual Leadership and Innovation in Making Cutting Edge Game Technology" (Press release). San Francisco, CA: Gamasutra. June 7, 2010. Retrieved January 18, 2015.
36. "Mobile game developer survey leans heavily toward iOS, Unity". Gamasutra. May 24, 2012. Retrieved July 14, 2014.
37. "Winners". Develop Online. Retrieved February 8, 2015.