

A History of the Unity Game Engine

An Interactive Qualifying Project

Submitted to the Faculty of WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for graduation

John Haas

Project Advisor:
Brian Moriarty, IMGD

Abstract

Unity (commonly known as Unity3D) is a game engine and integrated development environment (IDE) for creating interactive media, typically video games. As CEO David Helgason put it, Unity “is a toolset used to build games, and it’s the technology that executes the graphics, the audio, the physics, the interactions, [and] the networking.” [1] Unity is famous for its fast prototyping capabilities and large number of publishing targets.

The first version of Unity (1.0.0) was created by colleagues: David Helgason, Joachim Ante and Nicholas Francis in Denmark. [1] The initial product launched on June 6, 2005. [2] The goal was to create an affordable game engine with professional tools for amateur game developers while “democratize[ing the] game development” industry.[3] The three were inspired by the easy workflow, simple asset pipeline, and drag-and-drop interface of Apple’s Final Cut Pro product.[1] When originally released, Unity was available solely for Mac OS X, and developers could only deploy their creations to a few platforms. The current version (4.3.1 as of this writing) is supported on both Windows and Mac OS X, and offers at least a dozen target platforms. Publishing 64-bit executables has been possible in Unity for a long time, but a 64-bit version of the Editor is not yet available, though highly anticipated from the development community.

Authorship

I have been using Unity for personal and academic projects since version 1.5 (June 2006) and have been beta testing it since version 2.5 (March 2009). As a 14 year old boy, I was very interested in developing my own game on Mac OS X (we were an Apple household). After researching various development methods on the Internet, I stumbled across Unity and found it to be exactly what I was looking for. Unity was a development tool that wouldn't be too complex for me to learn at that age, and would grow with me as I learned more about game development. I completed my first few projects quickly, but was unable to publish them until I was 18 because I was a minor. Currently, I use Unity as my main development environment and have several self-published titles using this technology. Over the years, I have grown alongside Unity and have witnessed its transformation into the “game development beast”[1] it is today.

Table of Contents

Abstract	1
Authorship	2
Overview of Unity's History	4
The Unity Editor	12
The Mac Factor	15
Asset Pipeline and Workflow	16
Non-Game Uses	17
Multiplatform	18
Globalization and Expansion.....	18
Scripting Languages.....	19
Union / Unity Games	22
Publications / Tutorials About Unity	23
Unity Cloud	23
Licensing Unity	24
Unite	25
Asset Store.....	26
Appendix I: Notable Publications	28
Appendix II: Awards and Accolades	32
Appendix III: Supported File Types	34
Appendix IV: Scripting Language Comparison	35
Appendix V: Unity Publication List	36

Overview of Unity's History

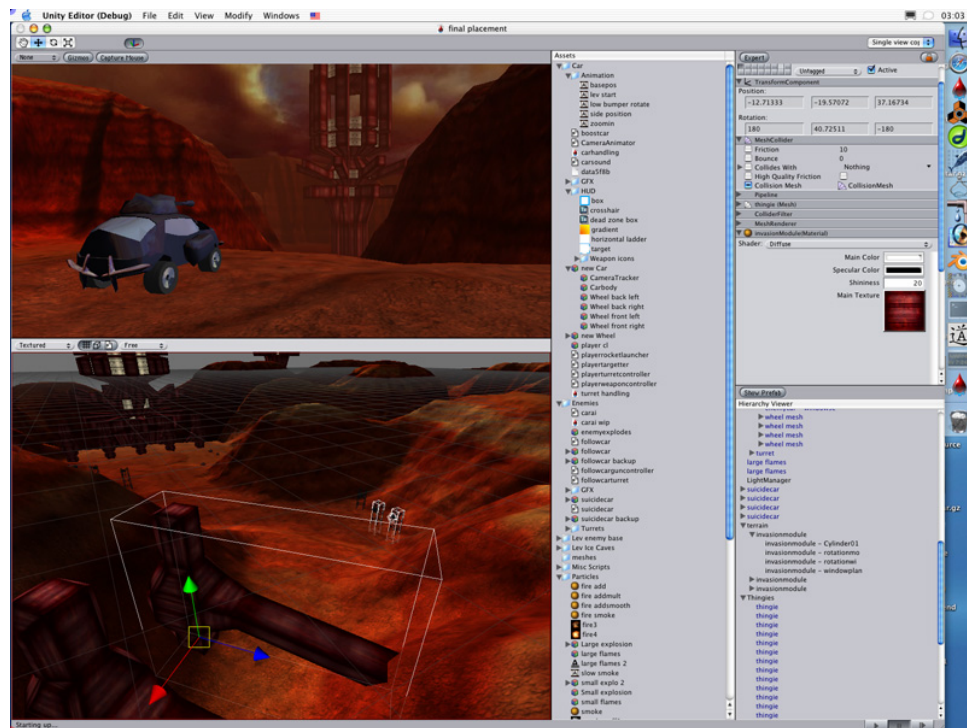
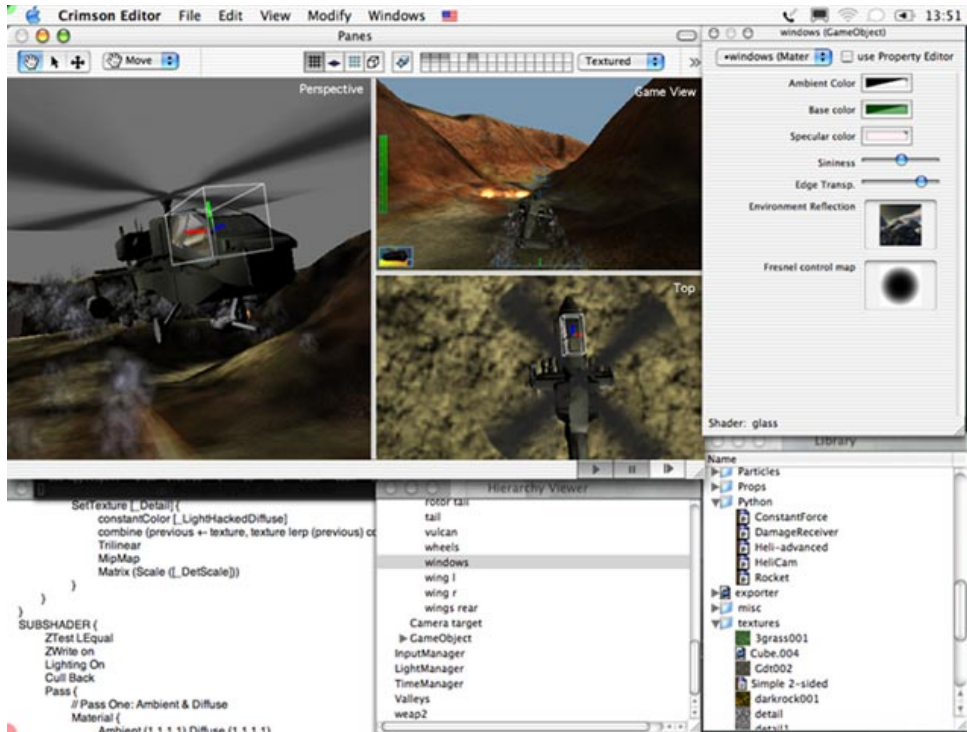
At 1:47am on May 21, 2002, Nicholas Francis, a Danish programmer, posted on the Mac OpenGL board asking for assistance with a shader system he was trying to implement into his game engine. [4] A shader is what gives a 3D object its look and feel. (See the Shaderlab section of this document for more information on Unity's shader system.) A few hours later, Joachim Ante, who lived in Germany at the time, responded to Nicholas' post. Their conversations resulted in the two developers collaborating to create a shader system that would work for both of their separate engines. [4] "After a while, we decided to scrap our individual engines and make an engine together, because it's more fun when there's two of you working on something," said Francis during an interview about Unity's origins. [4] David Helgason heard about the project and thought they were "really onto something, so [he] jumped aboard as the third developer." [4]

At first, they wanted to develop games for a living, but saw the need for a better underlining technology. The three always thought they would "make a game and then license out the technology" and that a "game was necessary to prove the tech." [4] In the end, they ended up not making games, but rather "making a tool to make games." [4]

Development really started to progress when Joachim and Nicholas rented a flat in Copenhagen. David lived just down the street, and worked at a cafe for extra food income. In the flat's basement, the three started developing what would become one of the most useful pieces of software in the video game industry. [1]

One of the major reasons for Unity's success was its support for independent developers who did not have the resources to license expensive game development technology. They knew they wanted to build some sort of game technology, but did not know which medium to use. Two years later, the trio found their mission: to create the definitive tool for 3D on the web. With this driving factor, the group took out substantial loans to open an office and hire a few engineers. The group was looking for a CEO, and initially thought that they would take turns fulfilling this position. They quickly realized that this would not work if they wanted to become successful. They put out an ad saying "Have great tech, need CEO." [4] They got a few applicants, but everyone they found was "slightly behind the curve - even though we weren't that far ahead," explains Helgason. [4] Helgason was the more social of the three, so the CEO position fell to him.

At this point, the group incorporated under the name Over the Edge Entertainment (OTEE) and started to form a business plan. They modeled their company after the British game development company Criterion, who was becoming very successful with the PlayStation 2 middleware market. [4] Criterion's model was not just to have good underlying technology, but also have multiple large titles using the technology. The thought was that consumers would not feel comfortable enough to license an expensive engine without seeing major game developers using it first.



Screenshots of early Unity version. Pre-initial release. ~Version 0.2b [41]

OTEE realized that casual and online gaming was only going to grow, and decided to focus their efforts on this market. After two years of coding, and many sleepless nights, Unity was nearing launch and the team decided they needed to make a full commercial game using their new engine. They did this not only to test the strengths and limits of their engine, but also to earn enough revenue for food and future development. [5] “Nobody really remembers how we survived in that period except we probably didn’t eat much,” said David Helgason. [1]

OTEE spent five months creating a game called *Gooball* using the new engine, which was still in beta. *Gooball* was published in March 2005 by Ambrosia Software. [6] The Unity team used this opportunity to tear apart their engine to find bugs, remove annoyances, and fix the interface before its official release. [5]



In-game screenshot of *Gooball*, [30]

Using the profits from *Gooball*, OTEE hired more developers to refine Unity before its initial 1.0 release in June 2005. They wanted to give Unity the best possible chance to succeed, so they smoothed out all the rough edges and also provided extensive documentation and support for the users. After its release, the company immediately began work on the next iteration of Unity.

“We were developers developing for other developers, we understood their problems and tried to fix them; we spoke to them. It’d be a case of not being able to sleep at night and thinking, ‘I wonder what’s going on the forums.’ It was all hours of the day. I think they recognized that, and then those customers we had started evangelizing it for us.” [1]

At this point, Unity was a baby to the video game industry. The majority of users consisted of hobbyists and independent developers. Many game development companies at the time would make a game that did not sell very well, and subsequently try to “dump” their engine technology. Potential customers were worried this might happen with Unity. It would take two more years before OTEE proved that Unity would be properly supported and updated. [1]

The first publicly available version of Unity only supported building projects to be run on Mac OS X. It wasn't until version 1.1, that Unity supported exporting to Microsoft Windows and web browsers. Browser support arrived just as the casual gaming scene exploded. Previously, Adobe’s Flash, with its clunky vector-based system, was one of the very few options for developers looking to publish to the web. Unity brought hardware-accelerated 3D graphics to the browser game industry. Version 1.1 also added support for external C/C++ plugins, allowing developers to extend the engine to use hardware and software not supported by Unity out of the

box. It was around this time the three co-founders felt that their product was up to their projected standards. [1] The team spent the next three months debugging, optimizing, and creating workarounds for older PCs that were plagued with graphics card bugs and driver issues. These were integrated into the Unity 1.5 release. [7]

Immediately after Unity's initial release, Unity 2.0's features were being developed on the side. [5] Two years of development later, Unity's "biggest step ever," version 2.0, was released. The focus of this release was to solidify Windows support and improve the web player compatibility across platforms. To accomplish this, the team added Microsoft DirectX support, which was preinstalled on Windows Vista systems, as opposed to OpenGL, which had to be downloaded separately. [5] DirectX added around a 30 percent speed increase on Windows systems in addition to the general performance optimizations afforded by Unity 2.0. [5] Other major features included in this release included: web streaming, real-time soft shadows, networking, a terrain engine, the Unity Asset Server, and a new code based GUI system. [5] Unity 2.0 was released during the 2007 Unite Developer Conference. [5]

With the rise of smartphone applications, and the release of the Apple iPhone and consequently the AppStore, Unity Technologies decided to develop a version of Unity that enabled publishing for the iPhone. Marketed as a separate product, Unity iPhone was released in December 2008 and was available in Basic and Pro versions, similar to the base Unity product. [18]

Between 2008 and 2009, the Unity Team, now known as Unity Technologies, noticed that their customers were buying Macintosh computers in order to use Unity. It quickly became

clear to the Unity team that they needed to support the editor on Windows as well. In order to do this, they had to tear apart the current editor and re-write it from scratch to be platform independent. Unity Technologies launched the first version of Unity with Windows support (Unity 2.5) at the 2009 Game Developers Conference.

The next major release was version 3.0 on September 27, 2010. [19] This update brought many desired features including editor unification, Beast lightmapping, deferred rendering, Umbra occlusion culling, low level debugging, and FMOD audio filters. [19] “With Unity 3, we're demonstrating that we can move faster than any other middleware company, that we're serious about the long term and that high technology made simple is a transformational force,” said Helgason in a press release. [19] In order to unify the editor, the Unity team took all the publishing platforms that required a different external editor (e.g. Unity iPhone, Wii) and incorporated them into a single editor along with the existing publishing options. At the time of Unity 3.0’s release there were more than 200,000 registered Unity developers. Unity became the #1 engine for educational purposes, and the most used technology on mobile platforms. [19]

On December 22, 2011 Unity Technologies began the first public open beta for Unity 3.5. This release included support for the long awaited Flash deployment option. [20] “Our vision has always been to let developers take their games to as many places as possible -- and Flash deployment greatly furthers that,” said Nicholas Francis. “While we normally wait to ship our software until it’s rock solid, the Flash publishing option is just so awesome that we had to do a developer preview. I can’t wait to see where our new and existing users will now take Unity.” [20] The latest release of Unity 3.5 was published on February 14, 2012. [18]

Unity Technologies opened pre-ordering for Unity 4 on June 18, 2012. This next release in the Unity cycle once again brought many new features, including Mecanim animation technology, Flash deployment, Linux publishing preview, and the Shuriken particle system. Customers who preordered also received early access to the beta list. [21] “We’ve been working on Unity 4 for a long, long time and are happy to finally be able to unveil its imminent arrival and outline its core features that will change the industry,” said Helgason. “The revolutionary animation system and add-on deployments to Adobe Flash and Linux are some of the critical features introduced in Unity 4, which together maintain Unity as the strongest, fastest evolving modern games development platform available.” [21] The final release of Unity 4.0 became available for developers to download on November 13, 2012.

The next long-awaited feature from Unity Technologies was 2D support. Previously, Unity developers could do “fake” 2D by putting textures on flat planes and utilizing other unconventional techniques. The next version of Unity, 4.3, included out-of-the-box support for sprites and 2D game development. [23] Developers were able to start authoring 2D games on November 12, 2013 when Unity 4.3 was released to the public. [23]

The Unity Editor

The Unity editor is comprised of many sub-windows. The most commonly used are: Project Browser, Inspector, Game View, Scene View, and Hierarchy. [46]

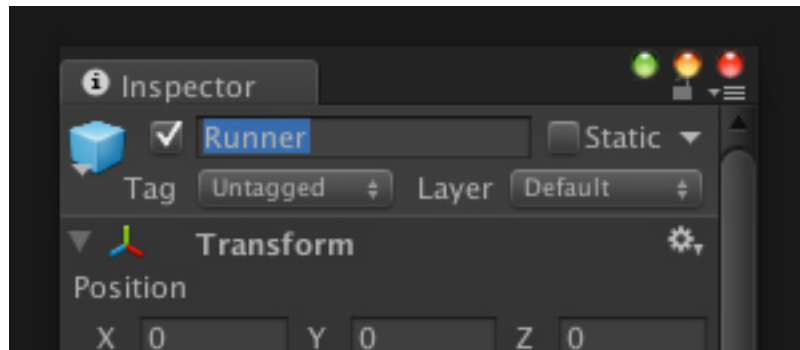
Project Browser



Screenshot of the Unity Project Browser [46]

The Project Browser is the window that contains all the assets that have been imported into Unity and are available for use. The layout is almost identical to Finder in Mac OS X, and Explorer in Windows. This makes developers feel more familiar and comfortable by providing an interface they already use on a daily basis.

Inspector



Screenshot of the Unity Inspector [46]

The Inspector window is where the details of every GameObject are viewed and modified. This is where the developer can tweak values to get just the right feel for their game. The Inspector shows all the Components that are attached to an object (e.g. Scripts, Physics, Colliders, Sound). This is also where variables exposed from scripts can be assigned or changed.

Game View



Screenshot of the Unity Game View [46]

The Game View provides the user with a WYSIWYG (what you see is what you get) preview of what their game will look like when they build it. It includes full input and permits the developer to test their changes without having to wait for their project to be compiled and deployed on the target platform.

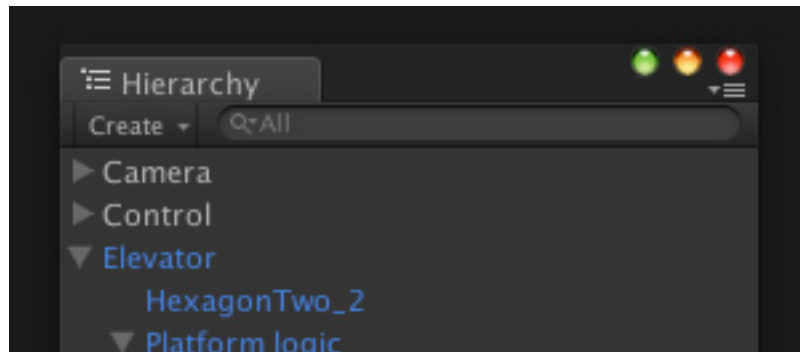
Scene View



Screenshot of the Unity Scene View [46]

The Scene View is where the game is constructed. The developer can drag and drop assets from the project view. Familiar 3D handle controls and grid snapping allows the user to place their objects in the perfect position, down to the pixel.

Hierarchy



Screenshot of the Unity Hierarchy [46]

The Hierarchy window contains a list of all the objects that are in the current scene. This list is automatically updated when an object is brought into the scene. In this window, by dragging an object on top of another object, the developer can assign parents or children.

The Mac Factor

Apple and Mac OS X developers have always been striving for simplicity and ease of use. “We came from a Mac side, so I guess we were encoded with that philosophy of that polish, that simplicity, that clarity that Apple’s software has. That was with us from the beginning. And then, quite early on, we figured out those key concepts of workflow.” [4] Unity incorporates all the workflow processes that Apple customers are used to. This makes it easier for new developers to get up to speed and start developing inspiring creations, without the hassle of navigating around a clunky interface.

Asset Pipeline and Workflow

Unity's asset pipeline is leading the industry in ease of use, and compatibility with current asset development software. To import an asset into Unity, the user simply has to drag it into project window and Unity takes care of the rest. When editing an asset, Unity automatically updates the asset with the latest modifications and spreads them across the entire project. [24]

“That's the real innovation behind Unity – I mean, the rest is really well done, but it's the workflow was just super-innovative. Nobody had figured out this concept of swallowing in whole assets from any application, using them, and updating them when they're saved. That was in super early, really early, and the product grew around the pipeline. It was in so early that we could import a multi-layer Photoshop file before we could even draw a mesh”. [4]

Unity supports importing from many applications (see appendix for complete list of supported file types), but only natively supports Autodesk FBX and COLLADA for models with bones or animations. [24] When importing most models of this type, Unity opens the original program that created the model and exports it to a FBX file that Unity can use. [24] This is performed automatically and is completely behind the scenes. It saves time that developers would otherwise be spending trying to convert assets into the proper format. If the game developer needs even more precise control over the import settings, they can extend the editor to fit their needs using the UnityEditor API namespace. [24] Unity's workflow also allows developers, who are targeting multiple platforms, to use the same source files by allowing the

user to override the import settings for each platform targeted. [24] If the developer is working on a massive project with many assets, they may have trouble locating the correct one in a timely manner. The project window allows the user to search and locate objects very efficiently using many parameters, including user-defined tags. The editor also includes a preview window, enabling the developer to view the appropriate asset without having to import it into the scene or apply it to an object.

With Unity 3.4, support for algorithmic substances was added. Substances are highly customizable, dynamic and procedural textures that can be manipulated in real-time to produce unique effects. They have an extremely small file size, which makes them ideal for streaming. They are dynamically generated, so they can be scaled up without losing quality. For platforms which that do not support this technology yet (e.g. iPhone, Android), Unity can bake substances into a standard bitmap.

Non-Game Uses

Recently, developers are beginning to realize that game engines can be successfully used for non-game applications. For example, architects can easily prototype ideas, artists can create interactive art installations, or researchers can use them for data visualization. The inverse does not apply, however. It is practically impossible to create a decent game using a computer-aided design (CAD) program. In an interview, Francis suggested that this, along with Unity's ease of use and gentle learning curve, is one of the main reasons Unity has flourished. [4] In a poll that

was given to Unity customers, it was discovered that a third of them were not using Unity to make games. [4]

Multiplatform

Another reason Unity has become so popular is its ability to deploy to a wide variety of target platforms, using the same code and assets. [8] It takes two button clicks to get your game running on another supported platform. One to switch platforms, and the other to build and run. Unity currently supports four main categories: Mobile, Desktop, Web and Consoles. For mobile devices, Unity can build for iOS, Android, BlackBerry and Windows Phone 8. On the desktop, executables can be created for Windows, Widows Store, Mac OS X and Linux (explicitly supported on Ubuntu, but known to work fine on other distributions). For desktops, developers can choose between 64-bit and 32-bit binaries. On Mac OS X they also have the option to create a Universal build (includes both 32-bit and 64-bit executables). For web-based targeting, Unity supports three browser-based options: Unity Web Player, Google Native Client and Flash (no longer supported in Unity 4.x+). [8][29]

Globalization and Expansion

Unity Technologies started in a small flat in Denmark, but is now a global corporation. The main office is located in San Francisco, CA, but the business has expanded into Canada,

China, Colombia, Denmark, Finland, Germany, Korea, Japan, Lithuania, Singapore, Sweden, Ukraine and The United Kingdom. [9] In just five years, Unity Technologies grew from a 3 person company to having over 60 employees. Approximately 20% of Unity Technologies revenue is currently coming from the United States. Another 50% is generated from other countries, and the remaining 30% comes from non-gaming applications. [39] The list of countries where Unity is used the most frequently includes: USA, China, South Korea, Japan, the UK, Canada, Russia, Germany, Brazil and France.

Currently, Unity Technologies consists of more than 300 employees working in 27 locations. They have 2.5 million registered developers, 500,000 monthly active users, and 347 million web player installs. [40] These statistics further cement Unity's place as one of fastest growing and largest developer communities in the world.

Scripting Languages

Unity offers its developers three ways to program their application built into the IDE: Unity JavaScript (commonly known as UnityScript), C# and Boo. Each one of these scripting languages is supposed to allow the developers to implement the same content independent of their language preferences. In reality, this proves to not be 100% accurate. These languages are run on Mono and have fast compile times. Mono is an open-source version of Microsoft's .NET platform that allows cross compatibility on different platforms. [26]

One scripting characteristic that is unique to Unity is visual properties. Visual properties allow public variables in scripts to be modified directly in the editor without touching the code. [26] This is useful because it allows for quick tuning in game to get just the right value. This tool also allows game designers, who are not familiar with code, to modify the feel of the project.

UnityScript

UnityScript is a Javascript-like language, and the better choice for novice users. The majority of Unity developers use UnityScript, so there are more examples, and it is easier to get assistance on IRC or the Unity forum. UnityScript is easy to learn and fast to type. It handles a lot of type casting behind the scenes and allows the user to switch between dynamic or strict typing. Javascript is a prototypal language, whereas UnityScript is a classical one, and therefore receives the benefit of classes and inheritance. [47] Even though UnityScript supports dynamic typing, it can be more inefficient than C# because the compiler has to take into account that an object's type may change. Privacy and private variable creation work the same way as in C#: all the programmer has to do is put the keyword "private" before they declare something. Although generally considered the beginners language, there are a few well-known developers who prefer UnityScript even after learning the other available languages.

C#

C# (pronounced C-Sharp) is moderately more challenging to learn (and a lot more to type) than UnityScript, but allows the programmer to have complete and precise control. C# does

very little “behind the scenes” for you, and requires the developer to manage most of the components manually.

Boo

Boo has a Python-like syntax and is structured similarly to UnityScript. Very few people use Boo, so if you are troubleshooting code, it may be hard to find somebody that can actually assist you.

The following is the same code written in all three languages:

UnityScript [10]:

```
function Update() {  
    // Move the object forward along its z axis 1 unit/sec  
    transform.Translate(Vector3.forward * Time.deltaTime);  
  
    // Move the object upward in world space 1 unit/sec  
    transform.Translate(Vector3.up * Time.deltaTime, Space.World);  
}
```

C# [10]:

```
using UnityEngine;  
using System.Collections;  
  
public class Example : MonoBehaviour {  
    void Update() {  
        // Move the object forward along its z axis 1 unit/sec  
        transform.Translate(Vector3.forward * Time.deltaTime);  
  
        // Move the object upward in world space 1 unit/sec  
        transform.Translate(Vector3.up * Time.deltaTime, Space.World);  
    }  
}
```

Boo [10]:

```
import UnityEngine
import System.Collections

public class Example(MonoBehaviour):
def Update() as void:
    // Move the object forward along its z axis 1 unit/second.
    transform.Translate((Vector3.forward * Time.deltaTime))

    // Move the object upward in world space 1 unit/second.
    transform.Translate((Vector3.up * Time.deltaTime), Space.World)
```

Union / Unity Games

Union is a business model and game publisher created by Unity Technologies and launched November 10, 2010. [12] Their office is located in San Francisco, CA, but they accept applications worldwide. [27] On July 1, 2013 the name of this program was changed to ‘Unity Games’ to “better reflect our mission to democratize games distribution”. [11] It was launched to help small-time developers improve publicity and bring their creations to otherwise unavailable platforms. The driving goal of Unity Games is to allow independent developers to “focus more energy on making great games instead of worrying about how to make money,” according to Seyler, General Manager of Unity Games. [12] Unity Games was primarily established as a tool for helping developers with promising game concepts to target platforms that would be normally unavailable to them due to high licensing costs, or other prerequisites. [27]

After a game is submitted to Unity Games and passes the initial review and screening process, porting begins to the unavailable platform. If a game is chosen and successfully ported,

Unity Technologies takes 20 percent of the revenue / royalties. The original developer who submitted their game gets the remaining 80 percent. [12]

Publications / Tutorials About Unity

Before Unity 3.0, tutorials, and publications for Unity were scarce and hard to come by, the Unify Community Wiki being one of the very few places to get information. This changed with Will Goldstone's *Unity Game Development Essentials*. This book was the first published that enabled users to learn Unity by following the provided tutorials. Before this, developers had to learn through trial and error and a lot of web searching.

With the rise of Unity's popularity there became a greater need for publications and tutorials to satisfy the growing community. Making the base Unity product free also increased the interest in Unity-related publications. As of this writing there are at least 23 regularly suggested books to get started. [45] See Appendix for a list of commonly cited resources.

Unity Cloud

Unity Cloud is an upcoming set of tools designed for Unity developers "to build, market and operate compelling multiplatform games without an extensive investment in cloud infrastructure." [13] Developers will be able to select as many or as few of the cloud tools as

they desire. The first version of the Unity Cloud service will reportedly support: player locating, title promotion, and cross-platform advertisements. All of these features will be built-in, without the need for a third party library (e.g. Admob, Mobclix). [13] Developers will be able to use the Unity Cloud ad service to earn extra revenue, or to cross-promote their other applications. “In short, we are working to unlock the reach of over half a billion Unity mobile game installs to create the largest game promotion network on the planet”. [13]

Licensing Unity

Currently, there are ten different licenses available to licensing the Unity engine: [14]

- Unity (Free)
- Unity Pro (\$1,500)
- iOS Basic (Free)
- iOS Professional (\$1,500)
- Android Basic (Free)
- Android Professional (\$1,500)
- BlackBerry Basic (Free)
- BlackBerry Professional (\$1,500)
- Windows Store Basic (Included in Unity Pro)
- Windows Store Professional (Included in Unity Pro)

In order to take advantage of a professional license (add-on), game developers must purchase a Unity Pro base license. On October 28, 2009, Unity Technologies abandoned the \$200 Unity Indie license and replaced it with a completely free version simply called Unity, or Unity Free. [15] “The thinking was that Unity Indie [was not] generating a significant portion of our revenue ... so we figured, let's take Indie and just give it to everyone ... What's important

[was] to get [Unity] in the hands of as many people as possible,” David Helgason said. He also hoped that by making the basic version of Unity completely free they could “further position [Unity] within the hobbyist and indie community.” [15]

Four years later, on May 21st, 2013, Unity Technologies issued a press release stating that all basic mobile add-on licenses were now free. [16] “We were able to make Unity free for the web and for desktop computers a while ago, but have been dreaming of doing the same for mobile for what seems like forever,” Helgason explained. [16] Two days later, on May 23, 2013, Unity Technologies introduced a new license type: subscription based. [38] This allows developers to pay a monthly fee instead of forking over the full cost of a license up front. The price of this new service is \$75/month for Unity Pro and an additional \$75/month for each add-on. [38] There is a 12-month minimum purchase. One of the benefits to this method is that you automatically get the newest releases of Unity without having to upgrade an existing license.

Unite

Unite is an annual event hosted for developers, publishers and connoisseurs alike. These conferences provide people with the opportunity to connect with other community members, share knowledge, and utilize resources. The first Unite took place in San Francisco, CA and lasted for three days. Unite “offer[s] attendees a chance to learn more about Unity and the Unity Web Player, to expand their horizons and explore what's possible”. [5] The second Unite conference took place from October 22-24, 2008 at the Tycho Brahe Planetarium in Copenhagen, Denmark. The Unity team went back to the United States for the 3rd Annual Unite,

again hosting it in San Francisco. In 2010, the event moved to Montreal Canada briefly, but returned to San Francisco in 2011. For the 2012 event, the Unity Community went to Amsterdam. Due to Unity's growth and expansion, the most recent event (2013) brought them to both Vancouver and Nordic regions.

At the Unite conventions, developers get to meet and chat with Unity engineers, receive early access to the next generation of Unity, and have the opportunity network with other Unity developers. [42] Training sessions delivered by the makers of Unity help the attendees learn more about the aspects of Unity that interest them. [42] Unity engineers will also sit down and help you with your own projects to polish out bugs, implement new ideas, or provide general feedback. Unite is the perfect opportunity for Unity developers around the world to get together and learn from the experts and each other.

Asset Store

On November 10, 2010, Unity Technologies announced and launched a third-party marketplace for Unity developers called the Unity Asset Store. [17] This market is built right into the IDE, so that developers can browse and purchase assets for their application without ever leaving their project. Individual developers can also post assets and packages. This provides the Unity community with an entirely new method to monetize their skills and talents. Unity Technologies keeps 30% of revenue and the asset designer receives the remaining 70%. [43] Top selling assets produce enough money for a person to comfortably live on. Calle Lundgren, from

the Indie game company VisionPunk, has made enough money selling his assets that he has lived for more than two years on nothing but these earnings. [44] Many developers also use the Asset Store to fund their more advantageous projects on the side. [44] Currently, there are thousands of asset packages available in the store, with more being added every day.

Appendix I: Notable Publications

- *GooBall* (Ambrosia Software, 2005)
- *Off-Road Velociraptor Safari* (Blurst/Flashbang Studios, 2008)



- *Cartoon Network Universe: FusionFall* (Cartoon Network, 2009)



- *Battlestar Galactica Online* (Bigpoint, 2011)
- *Family Guy Online* (20th Century Fox, 2011)



[37]

- *Shadowgun* (Madfinger Games, 2011)
- *Temple Run* (Imangi Studios, 2011)
- *Temple Run 2* (Imangi Studios, 2013)
- *Tiger Woods PGA Tour Online* (EA Sports, 2011)



[36]

- *Zombiville USA* (mikamobile, 2010)



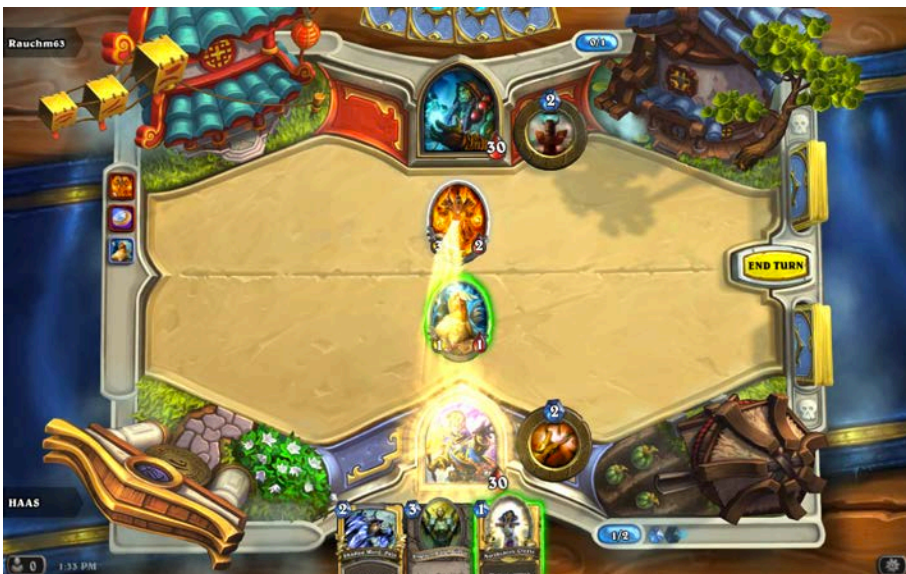
[35]

- *Skee-Ball* (Freeverse, 2009)
- *Hearthstone: Heroes of Warcraft* (Blizzard Entertainment, 2014)

HEARTHSTONE

HEROES OF WARCRAFT™

[34]



- *World Series of Poker: Full House Pro* (Microsoft, 2013)



[31]

Appendix II: Awards and Accolades

The list below was compiled from [9].

2006:

- Unity was named an Apple Design Award Runner-up for "Best Use of Mac OS X Graphics" at Apple's WWDC - this is the first time that a game development tool reached the level of quality and ease of use required to win this prestigious award

2008:

- Unity was named a Front Line Award Finalist by Game Developer

2009:

- Unity Technologies was named "One of the Top Five Game Companies of the Year" by Gamasutra
- Unity Technologies was credited with starting a "Tech Revolution" by Develop Magazine
- Unity Technologies was named a Front Line Award Finalist by Game Developer
- Unity Technologies CTO Joachim Ante made Develop Magazine's "30 under 30" List

2010:

- Unity Technologies won the prestigious Wall Street Journal Technology Innovation Award
- Unity Technologies won Develop Magazine's coveted Grand Prix Award and Technical Innovation Award Unity Technologies was named the Most Innovative Company at the SIME Media Momentum Award Ceremony
- Unity Technologies was named a Tech and Innovation Entrepreneur Award Finalist by San Francisco Business Times
- Unity Technologies was named "One of 20 Companies to Watch" for 2010-2011 by Game Developer
- Unity Technologies was named an "Interesting Company to Watch" by InformationWeek
- Unity Technologies was named a Front Line Award Finalist by Game Developer
- Unity was named a Best Engine Finalist by Develop Magazine
- Unity Technologies won GigaOM's Mobilize 2010 LaunchPad
- Unity was named a Mashable Best Mobile Platform Finalist

- Unity Technologies was named the Best Games Service Provider by Mobile Entertainment
- Unity Technologies CTO Joachim Ante made the Game Developer 50 List
- Unity Technologies CTO Joachim Ante received an honorable mention on Develop Magazine's "30 under 30" List

2011:

- Unity was named the #1 game engine that Game Developer readers use for traditional and casual games
- Unity won the Best Programming/Development Environment Award at the Computer Entertainment Developers Conference (CEDEC) in Japan
- Unity Technologies was named a Media Momentum “One to Watch”
- Unity Technologies was named a Telegraph Tech Start-Up 100
- Unity Technologies was named an AlwaysOn's OnMobile Top 100 Private Company
- Unity Technologies was named Game Developers Conference (GDC) Best of Show
- Unity Technologies was named a TiEcon 50 Finalist
- Unity Technologies was named a Red Herring 100 North America Finalist

2012:

- Unity ranks #1 for mobile game developer technology survey, topping the list with 53.1% of developers reporting using Unity.
- Unity named to AlwaysOn 2012 OnMobile 100 Top Private Companies in Enabling technologies category

Appendix III: Supported File Types

3D Package Support				
	Meshes	Textures	Anims	Bones
Maya .mb & .ma ¹	✓	✓	✓	✓
3D Studio Max .max ¹	✓	✓	✓	✓
Cheetah 3D .jas ¹	✓	✓	✓	✓
Cinema 4D .c4d ^{1 3}	✓	✓	✓	✓
Blender .blend ¹	✓	✓	✓	✓
modo .lxo ²	✓	✓	✓	
Autodesk FBX	✓	✓	✓	✓
COLLADA	✓	✓	✓	✓
Carrara ¹	✓	✓	✓	✓
Lightwave ¹	✓	✓	✓	✓
XSI 5.x ¹	✓	✓	✓	✓
SketchUp Pro ¹	✓	✓		
Wings 3D ¹	✓	✓		
3D Studio .3ds	✓			
Wavefront .obj	✓			
Drawing Interchange Files .dxf	✓			

¹ Import uses the application's FBX exporter. Unity then reads the FBX file.

² Import uses the application's COLLADA exporter. Unity then reads the COLLADA file.

Image Formats

- Photoshop .psd and .tiff are imported with layers automatically flattened.
- JPEG, PNG, GIF, BMP, TGA, IFF, PICT and many other image formats are supported.

Supported Audio & Video Formats

- MP3 and Ogg Vorbis .ogg audio files are natively supported, depending on platform. On mobile platforms, audio is converted to MP3 to take full use of hardware decompression.
- AIFF, WAV and most other audio format are supported, ideally suited for sound effects. Optional compression can be configured in the Unity Editor.
- MOD, IT, S3M, XM tracker files are fully supported.
- Ogg Theora video is natively supported.
- Video MOV, AVI, ASF, MPG, MPEG, MP4VIDEO files are transcoded by Unity with a configurable bitrate.

Other File Formats

- XML and text files with .xml and .txt extensions can be referenced at runtime.
- Any other file types, such as RTF and DOC, can be used for project notes and to-do lists.

[24]

Appendix IV: Scripting Language Comparison

Language	Boo	C#	UnityScript/JavaScript
Paradigm(s)	<ul style="list-style-type: none"> ▪ imperative ▪ object-oriented ▪ functional ▪ generic ▪ reflective 	<ul style="list-style-type: none"> ▪ imperative ▪ object-oriented ▪ functional ▪ generic ▪ reflective 	<ul style="list-style-type: none"> ▪ imperative ▪ object-oriented ▪ functional ▪ reflective
Standardized Variation?	Yes	Yes, ECMA, ISO ECMA-334; ISO/IEC 23270:2006	No
Type Strength	strong, duck	strong	strong, duck
Type Safety	safe	safe	safe
Expression of Types	implicit with optional explicit typing	explicit	implicit with optional explicit typing
Compatibility Among Composite Types	?	name-based	?
Type Checking	static with optional dynamic typing	static with optional dynamic typing	static with optional dynamic typing

[28]

Appendix V: Unity Publication List

The list below was compiled from [45].

- Blackman, Sue. *Beginning 3D Game Development with Unity 4: All-in-one, Multi-Platform Game Development*. Apress, 2013.
- McDermott, Wes. *Creating 3D Game Art for the iPhone with Unity*. Focal Press, 2010.
- Watkins, Adam. *Creating Games with Unity and Maya: How to Develop Fun and Marketable Games*. Focal Press, 2011.
- Goldstone, Will. *Developpez des jeux 3D avec Unity*. Pearson Education, Pearson Education, 2010.
- Totten, Chris. *Game Character Creation with Blender and Unity*. Sybex, 2012.
- Menard, Michelle. *Game Development with Unity*. Cengage Learning PTR, 2011.
- de Byl, Penny. *Holistic Game Development: An All in One Guide to Implementing Game Mechanics, Art, Design and Programming*. Focal Press, 2011.
- Duggan, Michael. *iPad Multiplayer Magic*. Cengage Learning PTR. Cengage Learning PTR, 2011.
- Alessi, Jeremy. *iPhone 3D Game Programming All in One*. Cengage Learning PTR, 2010.
- Stevenson, Craig. *Unity 3 Blueprints - A Practical Guide to Indie Games Development*. Deep Pixel Publishing, 2011.
- Wittaybundit, Jate. *Unity 3 Game Development Hotshot*. Packt Publishing, 2011.
- Goldstone, Will. *Unity 3.x Game Development Essentials*. Packt Publishing, 2011.

- Henson Creighton, Ryan. *Unity 3.x Game Development by Example*. Packt Publishing, 2011.
- Wiebe, Robert. *Unity iOS Essentials*. Packt Publishing, 2011.
- Pierce, Gregory. *Unity iOS Game Development*. Packt Publishing, 2012.
- W. Murray, Jeff. *Game Development for iOS with Unity3D*. A K Peters/CRC Press, 2012.
- Blackman, Sue. *Beginning 3D Game Development with Unity 4*. Apress, 2013.
- Chu, Philip. *Learn Unity 4 for iOS Game Development*. Apress, 2013.

Works Cited

1. Jon Brodtkin, . N.p.. Web. 4 Dec 2014. <<http://slashdot.org/topic/cloud/how-unity3d-become-a-game-development-beast/>>.
2. David Helgason, N.p., 2005. <<http://forum.unity3d.com/threads/56-Unity-1-0-is-shipping>>.
3. Unity Technologies. Unity Technologies, 2011. <<http://www.marketwired.com/press-release/unity-technologies-lands-12-million-series-b-funding-led-westsummit-capital-iglobe-partners-1540593.htm>>.
4. Ed Fear, . n.d. 2009. <<http://www.develop-online.net/analysis/united-they-stand/0116643>>.
5. Unite 2007 Developer Conference “Unite 2007 - Keynote” Event. San Francisco Film Centre, San Francisco, CA. October 10th, 2007. Keynote Presentation. <<https://unity3d.com/unite/archive/2007>>.
6. Ambrosia Software Inc, . N.p.. Web. 9 Dec 2013. <<http://www.ambrosiasw.com/news/story/GooBall-100-released-Get-Gooey-get-GooBall>>.
7. Joachim Ante, . . N.p., 2006. <<http://forum.unity3d.com/threads/2864-Unity-1-5-1-unleashed>>.

8. Unity Technologies. "Effortlessly Unleash Your Game on the World's Hottest Platforms." Unity. N.p., n.d. Web. 02 Jan. 2014. <<http://unity3d.com/unity/multiplatform>>.
9. Unity Technologies. "Fast Facts." Unity - Fast Facts. N.p., n.d. Web. 21 Dec. 2013. <<http://unity3d.com/company/public-relations>>.
10. Unity Technologies. "Transform. Translate." Unity Script Reference: Transform.Translate. N.p., n.d. Web. 17 Dec. 2013. <<https://docs.unity3d.com/Documentation/ScriptReference/Transform.Translate.html>>.
11. Unity Technologies. "Introducing Unity Games." Unity Technologies Blog. N.p., n.d. Web. 08 Jan. 2014. <<http://blogs.unity3d.com/2013/07/01/introducing-unity-games-2/>>.
12. Unity Technologies. "Develop." Unity Announces New Business Model; Union. N.p., n.d. Web. 17 Jan. 2014. <<http://www.develop-online.net/news/unity-announces-new-business-model-union/0108155>>.
13. Unity Technologies. "Announcing Unity Cloud and Our Mobile Ads Service." Unity Technologies Blog. N.p., n.d. Web. 02 Jan. 2014. <<http://blogs.unity3d.com/2013/08/30/announcing-unity-cloud-and-our-mobile-ads-service/>>.
14. Unity Technologies. "License Comparisons." Unity. N.p., n.d. Web. 01 Jan. 2014. <<http://unity3d.com/unity/licenses>>.
15. Remo, Chros. "In-Depth: Unity Launches Free Option, Announces Xbox 360 Support." Gamasutra. N.p., n.d. Web. 11 Dec. 2013. <http://www.gamasutra.com/php-bin/news_index.php?story=25840>.

16. Rose, Mike. "Unity's Mobile Licenses Are Now Free." *Gamasutra*. N.p., n.d. Web. 11 Dec. 2013. <<http://www.gamasutra.com/view/news/192654/>>.
17. Unity Technologies. "Unity Technologies Launches 3rd Party Marketplace 'Unity Asset Store'" *Marketwire*. N.p., n.d. Web. 5 Jan. 2014. <<http://www.marketwired.com/press-release/Unity-Technologies-Launches-3rd-Party-Marketplace-Unity-Asset-Store-1350977.htm>>.
18. Unity Technologies. "Press Releases." Unity - Press Releases. N.p., n.d. Web. 18 Jan. 2014. <<http://unity3d.com/company/public-relations/press-releases>>.
19. Unity Technologies. "Unity Technologies Delivers Unity 3." *Marketwire*. N.p., n.d. Web. 19 Jan. 2014. <<http://www.marketwired.com/press-release/Unity-Technologies-Delivers-Unity-3-1325564.htm>>.
20. Unity Technologies. "Unity Technologies Begins Unity 3.5 Open Beta With Flash Player Deployment." *Marketwire*. N.p., n.d. Web. 29 Dec. 2013. <<http://www.marketwired.com/press-release/unity-technologies-begins-unity-35-open-beta-with-flash-player-deployment-1601525.htm>>.
21. Unity Technologies. "The Next Generation of the Unity Game Engine Unveiled." *Marketwire*. N.p., n.d. Web. 10 Feb. 2014. <<http://www.marketwired.com/press-release/the-next-generation-of-the-unity-game-engine-unveiled-1670512.htm>>.
22. Helgason, David. "Unity 4 Is Out!" Unity Technologies Blog. N.p., n.d. Web. 22 Jan. 2014. <<http://blogs.unity3d.com/2012/11/13/unity-4-is-out/>>.
23. Unity Technologies. "Releasing Unity 4.3." Unity Technologies Blog. N.p., n.d. Web. 18 Feb. 2014. <<http://blogs.unity3d.com/2013/11/12/releasing-unity-4-3/>>.

24. Unity Technologies. "The Market-leading Import Pipeline." Unity - Asset Workflow. N.p., n.d. Web. 03 Feb. 2013. <<http://unity3d.com/unity/workflow/asset-workflow>>.
25. Unity Technologies. "Unity 3.4 Released." Unity Technologies Blog. N.p., n.d. Web. 14 Jan. 2014. <<http://blogs.unity3d.com/2011/07/26/unity-3-4-released/>>.
26. Unity Technologies. "Unity Scripting." Unity. N.p., n.d. Web. 12 Feb. 2014. <<http://unity3d.com/unity/workflow/scripting>>.
27. Unity Games. "About Us | Unity Games." Unity Games. N.p., n.d. Web. 6 Jan. 2014. <<http://www.unitygames.com/about-us/>>.
28. Unity Technologies. "Comparison of Programming Languages." Unify Community. N.p., n.d. Web. 15 Feb. 2014. <http://wiki.unity3d.com/index.php/Comparison_of_Programming_Languages>.
29. Helgason, David. "Sunsetting Flash." Unity Technologies Blog. N.p., n.d. Web. 03 Mar. 2014. <<http://blogs.unity3d.com/2013/04/23/sunsetting-flash/>>.
30. Over The Edge Entertainment. N.p., n.d. Web. 7 Jan. 2014. <http://i.i.cbsi.com/cnwk.1d/i/tim//2009/05/07/32031cc7945b0a0f8beb3ad0d9881f1e8aa1_1gooball1_540x405.jpg>.
31. Microsoft. "WSOP: Full House Pro." Xbox.com. N.p., n.d. Web. 03 Feb. 2014. <<http://marketplace.xbox.com/en-US/Product/WSOP-Full-House-Pro/66acd000-77fe-1000-9115-d802584112b9>>.
32. Brothersoft. "Off-Road Velociraptor Safari for Mac Free Download." Off-Road Velociraptor Safari for Mac Free Download. N.p., n.d. Web. 13 Jan. 2014. <<http://mac.brothersoft.com/off-road-velociraptor-safari.html>>.

33. Cartoon Network. "FusionFall." FusionFall Heroes. N.p., n.d. Web. 1 Feb. 2014.
<<http://www.cartoonnetwork.com/games/fusionfall-heroes/index.html>>.
34. Blizzard Entertainment "Hearthstone." Press Kit N.p., n.d. Web. 1 Feb 2014.
<<http://us.blizzard.com/en-us/company/press/presskits.html>>.
35. Mika Mobile. "Zombierville USA." App Store. N.p., n.d. Web. 03 Feb. 2014.
<<https://itunes.apple.com/us/app/zombierville-usa/id304871622?mt=8>>.
36. IGN. "Pre-E3 2009: Tiger Woods PGA Tour Online First Look." Tiger Woods PGA Tour Online First Look. N.p., n.d. Web. 03 Mar. 2014.
<<http://www.ign.com/articles/2009/05/29/pre-e3-2009-tiger-woods-pga-tour-online-first-look>>.
37. "Family Guy Online MMPORG Is Here, In Glorious Unity3D. GIGGIDY, GIGGIDY! | Wired In." *Wired In*. N.p., n.d. Web. 03 Feb. 2014.
<<http://wireframe.co.za/wiredin/?p=725>>.
38. Unity Technologies. "Create with Unity Pro for Just \$75/month." Unity Pro Now Available via Subscription. N.p., n.d. Web. 15 Feb. 2014.
<<http://www.gamefromscratch.com/post/2013/05/23/Unity-Pro-now-available-via-subscription.aspx>>.
39. Wauters, Robin. "TNW Profile: Unity CEO David Helgason Talks about Trends in Gaming, and His Business." TNW Network All Stories. N.p., n.d. Web. 24 Feb. 2014.
<<http://thenextweb.com/insider/2012/03/17/tnw-profile-unity-ceo-david-helgason-talks-about-trends-in-gaming-and-his-business/#!x9uEb>>.

40. Unity Technologies. "Democratizing Development." Unity. N.p., n.d. Web. 03 Mar. 2014. <<https://unity3d.com/company/public-relations>>.
41. Francis, Nicholas. "Blast From the Past." Unity Technologies Blog. N.p., n.d. Web. 7 Dec. 2013. <<http://blogs.unity3d.com/2009/04/20/blast-from-the-past/>>.
42. Unity Technologies. "Power up Your Creative Development at Unite." Unity. N.p., n.d. Web. 19 Jan. 2014. <<https://unity3d.com/unite>>.
43. Unity Technologies. "Sell Assets." Submitting Your Content to the Asset Store. N.p., n.d. Web. 03 Jan. 2014. <<https://unity3d.com/asset-store/sell-assets>>.
44. Paskova, Kristyna. "Funding Indie Games with the Asset Store." Unity Technologies Blog. N.p., n.d. Web. 15 Jan. 2014. <<http://blogs.unity3d.com/2013/07/19/funding-indie-games-asset-store/>>.
45. Derp. "Unity Book Roundup." Game from Scratch. N.p., n.d. Web. 10 Jan. 2014. <<http://www.gamefromscratch.com/page/Unity-Book-Roundup.aspx>>.
46. Unity Technologies. "The Unity Editor." Unity. N.p., n.d. Web. 20 Feb. 2014. <<https://unity3d.com/unity/workflow/integrated-editor>>.
47. Unity Technologies. "UnityScript versus JavaScript." Unify Community. N.p., n.d. Web. 18 Feb. 2014. <http://wiki.unity3d.com/index.php/UnityScript_versus_JavaScript>.