A VISUALIZATION AND GAME DEVELOPMENT PIPELINE FOR STUDENTS

Andrew Roncin, P. Eng.
Red River College
aroncin@rrc.ca

Abstract *Modelling, visualization, and simulation can be important aspects of a research project. For students and faculty outside the game development industry, understanding what is required can be a serious hurdle.*

The purpose of this paper is to give non-programmers an overview of the skill sets, software tools, and resources required to begin 3D simulation and game development. Because they are easily acquired tools that cover source control, asset creation, and game design; Git, Blender, and Unity® be examined

Keywords: Unity, Blender, Serious Games, Development tools

1. INTRODUCTION

This paper is aimed at students who are interested in using 3d game techniques in their research and the advisors who support them. This paper gives an overview of the skill sets required for 3d simulation and game development, one set of tools available, and a selection of resources that support those tools.

An interactive game environment, which features animated characters, dialog, and scenes, is going to require a lot of work. Modern video games have teams of developers, graphic artists, and business people to support their development. As a student it is very important that you have a well defined project and focus on the design aspects that reinforce the core concept of your game [1].

For this paper, Git, Blender, and Unity® will be the tools examined. This is because both Git and Blender are free open source programs, while Unity® has indie(free), student (\$159), and pro (\$5000) price points (as of January, 14, 2014). Focusing on Unity and open source software allows students to acquire the software for themselves and continue to use it after graduation.

1.1 Prerequisite Skills

The underlying skills required for developing a 3d application are visualization, design, and problem solving.

You will need to be able to create a unique idea, flush it out, and create the design specifications. From there the implementation will require programming, graphic arts, animation, narration, and game theory. Finally with all these pieces in place, you will need to troubleshoot your project until it is complete.

1.2 Three General Suggestions For Students

Here are three suggestions that will help you with the whole design process:

- Work through several small projects from start to finish. That way you learn about all aspects of the design process from creation, to release of a finished project.
- Be very specific about what you are creating and why. Find the minimum features you need to implement to make your game or simulation work. Once these are built and positively received you can build on them.
- Test out your ideas on others often. Get feedback from your users about what is working and what is not.

1.3 Three General Suggestions For Advisors

Game development is a broad area to which many students will find themselves swimming in unfamiliar waters. Your students will probably need help with the planning, technical, or legal aspects of their design:

- Focus on the project management techniques that result in finished projects. Things like, creating a clear design document, limiting the scope of deliverables. And good coding practices.
- Develop a training and resource library.
- Coach your students on how to document their sources and manage copyright issues

2. DEVELOPMENT PIPELINE

The remainder of this document looks at the source control, asset creation, and game engine elements of putting a 3d simulation or game together. Version control

is examined first, because it lays a foundation for easier development in the asset creation and game engine stages. Asset creation (modelling, skinning, rigging, and animating) are considered second because they are the tools required to make animated 3d objects that will be used by the game engine. Finally, the game engine will be considered, because it is here that you program how the game objects interact to create your simulation or 3d world.

2.1 Technical Skills - Source Control

Version control is an important aspect of software development. It makes your project more robust in three critical ways. 1) It allows for restoring files that have been deleted or changed beyond recognition. 2) It facilitates working in parallel either with other people, or on multiple computers. And 3) It allows you to branch off your project and focus on solving one problem at a time.

Version control is essential for Unity® projects because the projects are quite complex and easy to break by:

- Accidentally deleting a required asset or component.
- Updating a source file or importing assets that have compilation errors.
- Changing a prefab without consideration for how the child instances are used.

Version control is also very valuable in maintaining and versioning assets downloaded from the Unity Asset Store. Apps in this store are maintained by hundreds of independent developers and sometimes packages develop problems or even get pulled from the store. Downloaded assets are stored locally on Windows systems in the hidden folder %appdata%\Unity\Asset Store.

Simply put, having the ability to roll back a project, undo an import, or just plain see the differences between versions of your project will save you time and frustration.

3. GIT

Git is a popular open source distributed version control software that allows each user to keep a full version of the project and its history. Git uses a two stage commit process in which the user scans for changes in the file structure, chooses which files to stage, and then commits them to the repository. This allows the user greater control by allowing comments to be targeted to specific files and allowing changes to be spread over multiple commits. Git is very easy to use and well worth the effort.

The three images of Git BASH, Git GUI, and GitK comprise the "msysGit" interface for Windows.

Git BASH is the command line environment for GIT, and allows users full control of the GIT system. It uses a Linux style syntax and is easily recognizable in documentation because all the commands start with a dollar sign \$ prompt. This is important because much of the documentation on GIT assumes a command line interface.

"Git GUI" is a graphical add on which makes interfacing and using Git intuitively easy. Major GIT functions are listed in the menus, while the main window allows you to easily understand which files have been changed, which files are being committed, and what comments you are associating with your commit.

"GitK" is a graphical history viewer which allows you to easily see how the project has been branched. It also includes an inspector which makes it easy to see the differences between files. For stability reasons, do not branch the project in GitK, while controlling the repository in Git-GUI.

Figure 1: Git BASH is the command line editor for msysGit

Figure 2: Git GUI is the graphical interface for msysGit

Figure 3: GitK is the graphical history viewer for msyGit

3.1 Three Things Students Should Know About Git

- Git allows the user to selectively back up (commit) every file in the project folder that has changed since the last commit.
- Because you maintain a copy of each file revision, repositories can easily become several times your project size. Using a .gitignore file is an effective way to avoid backing up temporary files that are easily reconstructed.
- Although it is easy to store (push) your project onto a USB drive or online repository. You need to be careful about where you store your work. USB drives are easily stolen, and storing information on a public server could result in you violating licence agreements, intellectual property, or privacy laws, thus putting you and your intellectual property at risk.

3.2 Three Things Advisors Should Know About Git

- Git is an industry standard distributed source control program. It allows you to take snapshots of your project over time and restore them easily.
- Git facilitates team work and prevents students from writing over each other's work. Each GIT repository is a full copy of its parent.
- Git repositories can be stored both locally and on a server.

3.3 Resources

- Git is available for free at: git-scm.com
- Easy to understand resources: <http://www.gitguys.com/>
<http://www-cs-students.stanford.edu/~blynn/gitmagic/>
<http://stackoverflow.com/questions/tagged/git>

3.4 Using GIT With Unity®

Unity offers a “Unity Asset Server” to people with team licences. Alternatively, Unity® 4 also supports external version control in both its pro and indie licences’.

To have GIT work properly with Unity® 4, each project must be configured such that the meta files used to track changes within Unity are stored as text files in the same folders as the assets they are monitoring. This is done by adjusting the project settings as follows[2]:

Edit -> Project Settings -> Editor
mode -> visible meta files
asset serialization mode -> force text

Since GIT is designed to detect differences in files from one version to the next, even very small changes to the source or meta files will cause a new backup copy to be created during a commit. In order to minimize the amount of data that is being backed up, a .gitignore file is used to exclude binary files whose contents frequently change and are easily reconstructed. Only the “Assets” and “ProjectSettings” folders need to be kept. Table 1 shows a “.gitignore” file that eliminates many of the common temporary files found during development[3][4].

4. TECHNICAL SKILLS - ASSET CREATION

The Unity® interface allows you to draw basic shapes like cubes, spheres, capsules and particle effects. However making more complicated shapes like buildings or characters requires a 3d modelling package like those listed in table 2. Once created, the contents of a Maya, 3DS Max, or Blender file can be easily added to any project[5][6].

As image processing, normal maps, and ambient occlusion maps are critical to asset development, you should also consider investing in a good image editor such as Substance Designer, Photoshop, or Gimp listed in table 3.

4.1 Blender [7]

Blender is an open source, full featured program with tools for modelling, animation, and motion capture. It is designed to support a fast workflow over many diverse aspects of 3d modelling. But unfortunately for new users, the sheer amount of information available on screen, the number of tools and keyboard driven design, can be overwhelming. However, once mastered, Blender is a fast and powerful 3d modelling tool.

Table 1: A .gitignore file for use with Unity

```
# Adapted from www.gitignore.io
# # signs are comments
# * are wildcards
# / denotes a folder
# [ ] accepts any one character within the brackets

#### MonoDevelop ####
#User Specific
*.userprefs
*.usertasks
#Mono Project Files
*.pidb
*.resources
test-results/

#### Unity ####
[Ll]ibrary/
[Tt]emp/
[Oo]bj/
# Autogenerated Visual Studio and MonoDevelop files
/*.csproj
/*.unityproj
/*.sln
/*.suo
/*.user
/*.pidb

#### Windows ####
Thumbs.db
ehthumbs.db
Desktop.ini
$RECYCLE.BIN/
*.cab
*.msi
*.msm
*.msp

#### Blender ####
*.blend[0-9]
```

Table 2: 3d modelling programs

Maya (\$3675)
<http://www.autodesk.com/products/autodesk-maya>

3ds Max (\$3675)
<http://www.autodesk.com/store/3ds-max>

Blender (Free)
<http://www.blender.org/download>

Table 3: Image Processing Programs

Substance Designer (\$590)
<http://www.allegorithmic.com>

Bitmap2Material (\$149)
<http://www.allegorithmic.com>

Photoshop (\$79.99/month)
<http://www.adobe.com/ca/downloads.html>

Gimp (Free)
<http://www.gimp.org/downloads>

4.2 Three Things Students Should Know About Blender

- Start with modelling tutorials from Andrew Price, CGCookie, and CGBoorman.
- Memorizing the hot keys is critical.
<http://waldobronchart.be/blenderkeyboard/>
http://www.blenderguru.com/wp-content/uploads/Exclusive/Blender_Cheat_Sheet.pdf
- Try learning Blender in these stages, box modelling, modifiers, UV mapping, and rigging.

4.3 Three Things Advisors Should Know About Blender

- Blender has a challenging interface. Professionals often choose Maya over Blender.
- An Nvidia graphics card and multi-core system is recommended for rendering[8] [9].
- Building a library of well done training materials will really help your students. Training DVD's from Blender.org [10] [11] and CGCookie.com [12] are quite good.

4.4 Blender Resources

- Online manual, wiki, and tutorials
<http://wiki.blender.org/>
<http://www.blender.org/support/tutorials/>
- High quality training materials
<http://www.cgmasters.net/category/free-tutorials/>
http://www.blendtuts.com/blender_tutorials
<http://cgcookie.com/blender/>
<http://www.blenderguru.com/>
- Millions of videos on Vimeo and You-Tube
<http://www.youtube.com/user/tutor4u>
<http://www.youtube.com/user/AndrewPPrice>
<http://www.youtube.com/user/blendercookie>
<http://www.youtube.com/user/BlenderFoundation>
<http://www.youtube.com/user/cgboorman>

4.5 Using Blender With Unity®

Unity supports blender files "natively" within its project structure but handles these files through Blender's fbx exporter. When using Blender 2.69 and Unity 4, the following points should be considered:

- Blender files have to be saved in object mode for the fbx exporter to work properly. Otherwise Unity thinks the files are broken.
- Blender automatically creates backup files ".blendx" where x is a number that gets incremented. These automatic backups cannot be used by Unity (or Blender) unless their filenames are changed back to .blend. However, with version control, this could result in a lot of unnecessary files being saved.
- Material assignments are not preserved between Blender and Unity. UV maps should be clearly named and stored with the project so that they can be reassigned inside Unity.
- Shaders are slightly different between Blender and Unity.
- Depending on the target platform[13], you can be seriously constrained by the number of triangles being rendered or the number of draw calls being made. Keep your models simple. Reuse texture maps and atlases wherever possible.
- Investigate using linked and appended files within Blender to construct models. These features allow Blender to bring merge assets from other files and allow you a greater ability to modify and reuse design components.

5. GAME ENGINE

Game engines manage the transformations, collisions, animations, and triggering of events for all the game objects in your scene. And it is at this stage, where you enable the user to control what is happening in the game.

5.1 Unity® [14]

Although many game engines do exist, Unity® was selected because it has a reasonable student price, supports c# code, contains a comprehensive game engine, and has extensive documentation.

Unity® projects can be developed on Windows and Mac environments, for multiple platforms. Once written, Unity® allows you to configure the game controls to match the target platform. Thus allowing you to deliver applications to desktop, console, mobile and web based platforms.

5.2 Unity® Licensing

Unity Technologies has released the Unity® engine at several different price points. The commercial "Unity®Pro with mobile support" set development tools retails for \$6500 USD[15]. With upgrades costing half as much. Academic pricing is available either through the Studica website or contacting Unity directly. Further, students can purchase one year commercial and non-commercial student licences from Studica as well [16].

Although students and faculty are required to use the pro version, Unity Technologies also offers a free version available to independent developers [17]. This free version is most noticeably limited in the areas of inverse kinematics, navigation, and advanced shading.

5.3 Unity® Packages And The Asset Store

Unity® has an exceptionally flexible framework that allows the development community to create development tools and asset packages that can be used to enhance the editor or as content within your own games. This community support is handled through the Unity Asset Store [18] which has a searchable database of thousands of licensed assets which can be purchased.

5.4 Three Things To Know About The Asset Store.

- Assets may be removed at any time.
- Purchased assets are tied to individual licences.
- Before purchasing, review the developer's website to ensure the asset is being actively supported and meets your needs.

5.5 Three Useful Assets To Acquire

NGUI [19]: NGUI allows developers to more intuitively create graphical user interfaces. It also optimizes the GUI to reduce draw calls and thereby increase the frame rate.

Props Animation [20]: Props animation is a library of animation sequences to get a game developer started. With over 590 retargetable humanoid animation sequences, this product is excellent for learning the "Mecanim" animation controller or developing a proof of concept. In addition to the animations, it also includes working Mecanim animation controllers to demonstrate how the animation sequences should be controlled. Of particular note is the excellent technical support this developer is putting into this asset. Upon request, this developer added custom animations to the package free of charge based on videos I took.

UMA [21]: The Unity Multipurpose Avatar is a freely available package that lets developers create humanoid

characters for crowds easily. The humanoids are configured with a second rig that controls the aspect ratio of body parts, allowing each generated character to have a distinct appearance. UMA also uses texture atlas's to minimize processor loading and allowing for crowds to be easily created and managed.

5.6 Three Things Students Should Know About Unity

- The Mecanim portion of Unity® allows you to retarget animations from one avatar to another. It also features an easy to understand controller for managing the transitions between animation states.
- Certain assets cause compiler errors because they are constructed using the .NET 4.0 Framework while the Unity® project settings force MonoDevelop to use the .NET 3.5 Framework. A work around is located on the Unity forum[22].
- Unity requires all the code within the project folder to compile, even if it not being called. Thus, it is easy to put your project in an uncompileable state. Use backups or different file extensions to your advantage.

5.7 Three Things Advisors Should Know About Unity

- Unity® requires a Unity®Pro licence for academic work.
- Unity® has a full set of development tools capable of creating AAA games.
- Assets undergo continual development and may be removed from the Asset store or significantly changed over time. Consider routinely backing up assets that you have purchased.

5.8 Unity® Resources

- Online manual, forum, and tutorials
<http://docs.unity3d.com/Documentation/Manual>
<http://answers.unity3d.com/index.html>
<http://unity3d.com/learn/tutorials/modules>
- 3rd party training materials
<http://www.unity3dstudent.com/category/modules/>
- You-Tube Channels
<http://www.youtube.com/user/Unity3D>
<http://www.youtube.com/user/TornadoTwins>
<http://www.youtube.com/user/4GameFree>
<http://www.youtube.com/user/Brackeys>
<http://www.youtube.com/user/BurgZergArcade>
- C# programming
<http://www.dotnetperls.com/>
http://en.wikibooks.org/wiki/C_Sharp_Programming
<http://www.csharp-station.com/tutorial.aspx>

<http://csharp.net-tutorials.com/>
<http://www.learncs.org/>
<http://stackoverflow.com/questions/tagged/c%23>
• Free textures available for commercial use
<http://www.cgtextures.com/>
<http://www.mayang.com/textures/>
<http://freestocktextures.com/>
<http://www.texturemate.com/>

6. CONCLUSION

Unity® is a good engine for doing games and visualization. However it does take some time to learn and develop expertise. To be effective in this process, a student should have a reasonable understanding of object orientated programming and 3d graphics. Given the overall complexities of these programs, mastery will take some time and effort. This paper has provided a rough overview of three areas needed to start game development using Unity®, along with useful websites that will serve as resources along the way.

7. BIBLIOGRAPHY

- [1] J. Schell, The art of game design: A book of lenses, Amsterdam: Morgan Kaufmann Publishers, 2008.
- [2] S. Pierre, "Some Git and some Unity," OriGamine, 21 March 2012. [Online]. Available: <http://www.shawn.origaminc.com/blog/?p=357>. [Accessed 24 May 2014].
- [3] "nik.shornikov", ".gitignore for Visual Studio Projects and Solutions," Stack Overflow, 27 October 2013. [Online]. Available: <http://stackoverflow.com/questions/2143956/gitignore-for-visual-studio-projects-and-solutions>. [Accessed 24 May 2014].
- [4] C. McKnight, "gitignore file syntax," Stack Overflow, 15 December 2011. [Online]. Available: <http://stackoverflow.com/questions/8527597/gitignore-file-syntax>. [Accessed 24 May 2014].
- [5] Brackeys, "Best Modeling Software for Unity," Brackeys, [Online]. Available: <http://www.youtube.com/watch?v=r1BRTNqCJ1A>. [Accessed 24 May 2014].
- [6] Unity Technologies, "The market-leading import pipeline," Unity Technologies, [Online]. Available: <http://unity3d.com/unity/workflow/asset-workflow>. [Accessed 24 May 2014].
- [7] Blender Foundation, "Blender," Blender Foundation, [Online]. Available: <http://www.blender.org/>. [Accessed 24 May 2014].
- [8] Blender Foundation, "Requirements," Blender Foundation, [Online]. Available: <http://www.blender.org/download/requirements/>.

- [Accessed 24 May 2014].
- [9] A. Price, "The Ultimate Guide to Buying a Computer for Blender," Blender Guru, 13 January 2013. [Online]. Available: <http://www.blenderguru.com/the-ultimate-guide-to-buying-a-computer-for-blender/>. [Accessed 24 May 2014].
- [10] N. Vegdahl, "DVD training 8: Humane Rigging," Blender Open Movie Workshop, [Online]. Available: http://www.blender3d.org/e-shop/product_info_n.php?products_id=146. [Accessed 24 May 2014].
- [11] blender Foundation, "DVDs," Blender Foundation/Institute, [Online]. Available: http://www.blender3d.org/e-shop/default_dvds.php. [Accessed 24 May 2014].
- [12] CG Cookie, "CG COOKIE Education Hub Site," CG Cookie, Inc., [Online]. Available: <http://cgcookie.com/>. [Accessed 24 May 2014].
- [13] "Chris 2", "How many polygons is too much?," Unity Technologies, 12 June 10. [Online]. Available: <http://answers.unity3d.com/questions/19379/how-many-polygons-is-too-much.html>. [Accessed 24 May 2014].
- [14] Unity Technologies, "Create the games you love with Unity," Unity Technologies, [Online]. Available: <http://unity3d.com/unity>. [Accessed 24 May 2014].
- [15] Unity Technologies, "Pricing," Unity Technologies, [Online]. Available: <https://store.unity3d.com/products/pricing>. [Accessed 1 March 2014].
- [16] Studica, "Students & Faculty: Create Amazing Games with Unity," Studica, [Online]. Available: <http://www.studica.com/unity>. [Accessed 24 May 2014].
- [17] U. Technologies, "License Comparisons," Unity Technologies, [Online]. Available: <http://unity3d.com/unity/licenses>. [Accessed 24 May 2014].
- [18] Unity Technologies, "Unity Asset Store," Unity Technologies, [Online]. Available: <https://www.assetstore.unity3d.com/>. [Accessed 24 May 2014].
- [19] Tasharen Entertainment, "NGUI: Next-Gen UI," Tasharen Entertainment, [Online]. Available: <https://www.assetstore.unity3d.com/#/content/2413>. [Accessed 24 May 2014].
- [20] Proportional Studios, "Props Animations," Proportional Studios, [Online]. Available: <https://www.assetstore.unity3d.com/en/#!/content/10214>. [Accessed 24 May 2014].
- [21] UMA Steering Group, "UMA - Unity Multipurpose Avatar," UMA Steering Group, [Online]. Available: <https://www.assetstore.unity3d.com/en/#!/content/13930>. [Accessed 24 May 2014].
- [22] "MattRix", "Unity is overwriting my MonoDevelop target framework setting every time it syncs.," Unity Technologies, 25 July 2012. [Online]. Available: <http://forum.unity3d.com/threads/145027-Unity-is-overwriting-my-MonoDevelop-target-framework-setting-every-time-it-syncs>. [Accessed 24 May 2014].