

Final Project: XNA Video Game (v1.0)

Assigned: Monday, Nov 9, 2009

Gold release due: Wednesday, Dec. 9, 2009 at 11:59pm

Video Game Competition: Thursday, Dec. 10, 2009

SECTION I. Introduction

1. Introduction

Welcome to **AGL Games Division!** As a new hire, your first task will be to work on a short game to demonstrate your skills to see if you are ready to move up to some of the larger AAA titles we are currently working on. You will work in teams of up to two new hires and will have a little over one month to write a game that runs on a PC and an XBOX 360, and you will do so using Microsoft's XNA 3.0 framework to develop your code. You are free to write any kind of game you like, as long as it meets our internal requirements. This handout contains most of the information you will need to get started on the project and to complete it successfully. After all, we just hired you, the last thing we want to do is let you go after a month on the job! Read the handout thoroughly before you begin to make sure that you understand the project requirements. Also, we will set up a website off the course webpage that will have more information regarding this project (useful links, etc). Please be sure to check it regularly for updates. Good luck and enjoy your first project here at **AGL Games Division!**

2. Teams

For this assignment, you may work in **teams of one or two people**. We limit the size of the team for this project because it is difficult to divide the work fairly in groups of larger size. Although most of the grade for the project will be shared by all team members, there is an individual grade component that will be assessed based on individual effort. Therefore, it is important that *everyone* on the team contribute as much as possible since it will affect their final grade. Individual team members are required to maintain a work log where they keep track of both the hours that they work and what they accomplished in the process.

3. Game Requirements

Although here at **AGL Games Division** we do not tell our employees what kind of games to write, we do require that your game meet the following specifications:

1. First of all, **your project must be a game**, meaning that it must have a goal or purpose and there should be winning and losing conditions. Just building an environment to walk around in and check things out might be fun, but it is not technically a game. If you instead have to finish the path in a certain amount of time, or collect the most gold coins along the way, or solve puzzles to open the final door, then it's a game. Adding the winning condition is often left to the last minute because people are too busy working on

the graphics, the game engine, etc, to actually check if the player has won or lost. Be sure to incorporate this into your code early on in the development process.

2. It must be a 3-D game. Although 2-D games are fun, this class is primarily about 3-D graphics and so one of the requirements is that the game is a 3-D game. Therefore the objects in your game should consist primarily of 3-D models, although you are free to use 2-D sprites for effects and billboarding. Of course, the game screen could be looking down at the world and therefore have most of the “game play” happen on a 2-D surface. However, as long as it still has a 3-D engine we would still consider this to be a 3-D game. Ask me if you have questions about this.

3. The game must play on the Windows machines in ECE 211 and on the XBOX 360's in the AGL lab. The game should run in a “full-screen” mode on a PC.

4. The game must take user input. On the XBOX 360 implementation, the user can use the game pad controller, but for the PC version of the game the user should have the option of using the keyboard or mouse. Of course, this should only happen when it makes sense to use a keyboard or mouse. For example, a multiplayer with split-screen will have to use other inputs on top of the keyboard/mouse in order for several people to play at the same time.

5. Your game should have sound effects and music. You can record these yourself with a microphone, use digital musical instruments to create them if you are so inclined, or borrow these from internet sources where appropriate (please acknowledge the content of all sources). Also, we have access to some local artists you can request music/artwork from.

6. When the game is first started, the game must feature the required company splash screens and then have an introduction screen for the game itself. There are three company splash screens you will need to show: the UNM screen, the ECE/CS department screen, and the **AGL Games Division** screen. The code to run these splash screens will be provided to you through a file on the project webpage.

7. The game must have a “credit” screen that plays at the conclusion of the game or when it is on idle mode. This screen (or screens) should contain the following information: 1) the names of everyone on the team (you can include photographs if you want), 2) the names of other people who helped you with the project, 3) acknowledgements about any sources of content that you might have used in the game. You should also list your project manager (Pradeep Sen) as the “Producer” in the credits. He is responsible for working with you through out the game development process to ensure that your game is a success.

8. Your game should contain an instruction screen on how to play. By hitting a certain key or a button on the controller, a screen should pop up that describes the basic game play and controls.

9. The game you develop should be rated “E (Everyone)” through “T (Teen),” as defined by the Entertainment Software Ratings Board (ESRB). **AGL Games Division** is a family-friendly company and we want to preserve that image. The complete description of the ratings is available on the ESRB website at http://www.esrb.org/ratings/ratings_guide.jsp

10. Finally, your manager at **AGL Games Division** will need to approve your game based on your proposal pitch. In addition, you will have progress report meetings through the process where specific suggestions will be made to add to game play and graphics. Your project manager is Dr. Pradeep Sen, and he will be working closely with you in developing your game to ensure that it is a success.

In addition to these game requirements, your game must fulfill the following graphics requirements:

1. Your game should contain objects that are lit and shaded from light sources in the game. The lights can be either static or dynamic and you can use simple models (diffuse, Phong, etc.) for shading the surfaces. For a little eye-candy, you might want to later make these models a little more sophisticated.

2. Some of the models in your game should have texture-mapped surfaces. You will need to make your own texture maps or find them on the web. In addition, we will also provide you with specific artwork which you will be required to include as textures in your game where appropriate.

3. You should use vertex and fragment shaders to create effects in your game. It is probably easiest to implement these in Microsoft’s HLSL, examples of which you can find off the project webpage. You will be graded on the complexity and sophistication of your games.

4. You will need to implement “eye-candy” effects to improve the look of the game. A significant portion of your final score will come from the visual-look of the game. A list of the kinds of graphics effects you can use is in Section 4.

4. Graphics Effects

Because **AGL Games Division** is known for the quality of our real-time graphics, you will be required to implement several graphics effects to improve the look of your game. Your management team will decide your score based on your effort and the quality of your effects, as compared to the other teams. Some of the effects you can do include:

Shadows – Shadows can add a lot to the look of a game. If you choose to implement shadows, choose an algorithm that will result in the desired effect.

Lens Flares – These effects can give your game the look as if it was filmed with a camera.

Bump-mapping/Texture effects – Adding texture-based effects such as bump-mapping can significantly add realism to objects in your game.

Level-of-detail control – Objects that are far away need not be modeled in as much detail as the objects that are near the viewer. Implementing LOD control will increase the speed and performance of your game engine.

2-D postprocessing effects – These effects allow you to do blurs, dept-of-field and other effects that make the game look more realistic or to add to the game play. For example, when your character gets hit by a poison dart the screen can go blurry to emulate what happens to the character.

Configuration screens – It’s always cool when players can modify the look of their characters by changing the skins, modifying the level (weather, course, etc). Create a configuration screen that allows gamers to personalize their experience.

Articulated characters – Although XNA does not support classes for skinned animation of articulated characters, there are ways to do this using the current framework. Implementing articulated characters in your game could earn you extra credit.

5. Game Add-ons

Of course, your game is likely to require more than just fancy graphics in order to be fun. Some of the kinds of things you can add are:

AI – Some games will require enemy/friendly AI

Level editor – Depending on the game, you might need to build a level-editor to allow you to rapidly create content for your game.

Networking – Although the XNA framework does not explicitly support networking, it is still possible to implement this using the networking classes.

Physics – Your game might require implementation of a physics engine to keep track of objects in the game

SECTION II. Implementation

1. XNA Framework

In order to develop your game, you will be using Microsoft’s XNA 3.1 Game Studio framework. You can download the latest version of XNA at Microsoft’s webpage at <http://creators.xna.com/>. You will need to install a version of Visual Studio to run XNA Game Studio. However, everything that you will need to use is freely available, since you can download the Visual C# Express for free. XNA requires graphics hardware that supports DirectX9.0c with Shader Model 1.1. Microsoft further recommends Shader Model 2.0 compatibility if you want to be able to run all the demos.

2. Content creation

The successful games will have rich content full of 3-D models, textures, shaders, graphics effects, and so on. You will be expected to develop some of your own content as well as use some available on the internet. There are several ways to create your content:

3-D models: Models can be designed in a 3-D modeling package such as Blender (free), or other commercial modeling packages. You will then need to figure out a way to import these into your XNA project. You can also sketch out the models on graph paper and enter the coordinates manually into your program. Another good option is to generate your models procedurally, which can be done with things like terrain and vegetation. Finally, you can find a wide-variety of content on the internet. Make sure that you are allowed to re-use the content and acknowledge the source of the in your documentation and in your credits screen(s).

Textures: You can design 2-D textures using bitmap software such as Adobe Photoshop. You should also implement some of these as procedural shaders.

Sound/music: You can record sound/music yourself with a microphone, use digital musical instruments to create them if you are so inclined, or borrow these from internet sources where appropriate (please acknowledge the content of all sources).

Artists: At AGL Games Division we employ several artists who can help you with the assets for your game. Communicate via the class forum and ask them for specific things you want them to make.

3. Use of existing code

While we highly encourage you to use existing content (models, textures, sounds, etc.) in your game, you need to be more careful in using existing code. After all, you have been hired for your programming skills, and we need to make sure that you have what it takes to be the lead programmer for one of our AAA titles. For example, you are allowed to use code to implement low-level libraries (matrix classes, model loaders, etc) as long as you carefully document where you got the code. You can also draw inspiration from the XNA projects available for download, but you want to be sure to rewrite the code yourself rather than simply copy-and-paste (we will frown upon code that looks too much like the original demo application). Of course, you will still need to cite where you got your examples from in both your code and in your write-up. Finally, we want you to implement the code to do the graphics effects (such as the shadows, bump-mapping, etc) yourself. Again, you are free to look at existing code for inspiration but do not copy and paste.

3. Distribution of Games

The objective of this project is to have a high-quality game that we can distribute to the world through the AGL Games Division. Unfortunately, based on our past experience, not all students will make a proper, high-quality game. Furthermore, no matter how hard you work it is impossible to make a top game in just a month. So we have developed a two-semester system that allows students to develop working games. Here is how it works. After the games are completed this semester, we will have a public game session and private judging for the games (see below). At this point, the AGL management will choose the best games to release to the outside world. These students will be given the opportunity to take an advanced video game development course in which they will

continue to work and polish their video games and continue working with artists one-on-one while they improve their game. Once finished, these games will be released by the AGL Dev Club, a non-profit student group which is part of the Advanced Graphics Lab. The AGL Dev Club has already purchased a developer's club premium license and is producing a couple of games this semester as a trial run.

Finally, we must mention that before we post any game under the AGL brand for public download, we want to make sure that it lives up to our high standards for quality. Therefore, we will have judges and the AGL management decide which games are good enough to release to the public. The reason for this is that we want to build a brand as one of the better publishers in XBOX Live, which will benefit future students as we post new games.

4. Getting Started

This is a large project, so it would be wise to get started as soon as possible. I would suggest going through the XNA tutorials *individually* before you start working as a team to ensure that every team member understands the XNA framework. Spend some time brainstorming game ideas and looking through some of the sample games to make sure that your game is at the right level.

SECTION III. Deadlines, deliverables, and details

1. Your manager

One of our project managers, Dr. Pradeep Sen, will be overseeing your progress on this project. You will need to schedule three progress-report meetings with him during the course of the project, where the majority of team members must attend. These will be 15 minute meetings and should be scheduled through the website in a first-come first-serve basis. The goal of these meetings is to help make your game a success, since after all the future of **AGL Games Division** depends on having successful games.

2. Deliverables

This project requires that you submit various deliverables, both presentation materials as well as your executable of your game files.

1) Project proposal - You must submit a two-page description of the game you are going to work on for approval by **AGL Games Division** management. On the first page, you will need to address the following questions:

- What is the game about?
- What is the game play like?
- What will the 3-D environment of the game look like?
- Where will you be getting your assets and content (textures, models, sounds, etc)?
- Who are the members on your team who will be working on the project?

You should also come up with a team name, and include at least two images showing sample screen shots of the game. You can draw these by hand, on Photoshop, etc. These are meant to give us an idea of what the game will look like.

On the second page, your team should provide a timeline that indicates what your milestones so that you can submit the gold version of your code by 11:59pm **December 9**.

2) Early publicity materials – Shortly before the video game competition, you will be required to submit three screenshots and a short paragraph about your game. These will be used to promote your game and the video game competition.

3) Game code and executables – You must submit your final code and executables the night before the competition so that we can get it loaded onto the demo computers and XBOX systems. At the end, you will need to have two main deliverables, a .ccgame package that will be uploaded to XBOX Live and a “ClickOnce” executable that will be set up in a UNM portal for game download onto Windows machines.

4) Project webpage – You will need to make a simple webpage that describes your game, provides instructions and shows some screen shots.

5) Post-mortem report – You will have to write a post-mortem report as is convention for game developers. In particular, you will be writing this in the style of Game Developer magazine, also known as the “5 rights, 5 wrongs” concept. Check out their website at <http://www.gdmag.com/postmort.htm> for details on how to write the report. After listing the 5 rights and 5 wrongs, the post-mortem should conclude with two things. First, should be a list of the features you implemented in your game (e.g. shadows, physics, etc) and a few sentences for each describing how they were implemented and how they worked in the end. The second should be a list of citations for the sources for code and content you have used in your project. I will post last year’s post-mortems off the course website for you to read. Submit these to me via WebCT.

6) Work log – Each member of the group will be required to keep a work log of the hours they worked and what was accomplished during those hours. This individual work log should be submitted at the end of the project along with the post-mortem write-up. When submitting the work logs, all team members must review everyone’s work hours and sign off on it to show that they concur with the work performed by every team member.

7) Final publicity materials – Similar to the early publicity materials, we will need updated screen shots and a short paragraph that describes your game after it has been finished. This will be used in the final project website and used for promoting the class.

Note that several of the presentation materials (numbers 2, 4, and 7), require similar content. We would recommend that you re-use content wherever possible in order to avoid wasted effort when developing these presentation materials.

3. Deadlines

Here are the important dates for this project that you must keep in mind:

November 12, 2009 – Initial project proposal due by 11:59pm

Week of November 16, 2009 – Progress report meeting (schedule online)

Week of November 23, 2009 – Progress report meeting (schedule online)

Week of November 30, 2009 – Progress report meeting (schedule online)

December 6, 2009 – You should be playtesting your game

December 6, 2009 – Early publicity materials due by 11:59pm

December 9, 2009 – Finished projects should be submitted by 11:59pm, so that we can install it on the machines for the video game competition on December 11

December 10, 2009 – Public play session and video game competition

December 13, 2009 – Post-mortem report, work log, project webpage are due at 11:59pm

Spring 2010 – After the semester is over, selected students will be invited to take an advanced video game class where the games will be published on XBOX Live.

4. Progress Report Meetings

The purpose of the progress report meetings is for the AGL Games Division management to see your progress first-hand and to give you important feedback on your games. Specifically, your manager will give you direct suggestions that you must implement. Part of your score will depend on how you implement these requests. These meetings should only be 15 minutes long and more than half the development team must be present. To give you a good benchmark, by the meeting of November 23 you should already have the basics of your 3-D environment, and there should be some rough game play implemented so that your project is starting to look like a real game.

5. Playtesting (December 6)

Aim to have finished all major parts of your game by Sunday Dec. 6, so that you can upload it to the XBOXes in the AGL for public playtesting. It is important that you get feedback from others to help improve your games.

5. Gold release (December 9, 11:59pm)

You must submit your finished game by 11:59pm on December 9 so that we can load it onto the computers (or XBOX) for the video game competition. Late submissions will not participate in the video game competition and will be penalized at 25% per late day. Remember that you are not allowed to use your late days on this project!

6. The Big Day (December 10)

The main day for this project will be **December 10**. Here is what will happen on that special day:

1) Public Game Play Session

All games will be on public display on December 10 from 9am to 5pm in the atrium of the ECE building with 2-hour time slots per game. You will need to make a poster describing your game to the public so that people walking by will have some idea how to play it and what it took to develop the game. Sign up for the public game session online through the project website. It is preferable that at least one member of each team be present during your time slot to answer questions and help with the game.

2) Video Game Competition

After the public play session, a group of judges will be judging the games from 5pm to 6:30pm. Each team will get a 10 to 12 minute timeslot (length of time will depend on the total number of games), where they get to demo the game to the judges and present their features. Please sign up for a slot on the website, and be ready to present when your time comes. This will be a private session and only team members, **AGL Games Division** management and the judges will be allowed. We recommend that your entire team be there to present the game to the judges, since they might have specific implementation questions (note that you only need to be there for your own judging session). Judges will be members of the game development community and will be judging the games based on their own criteria (fun, quality, etc). The results of the judging as well as AGL management will be used to decide which games to pursue for later release.

3) Awards Ceremony

The award ceremony will take place immediately after the judging of the games. For example, the top team will receive \$100 gift certificates to Best Buy while other prizes will be awarded at the discretion of the judges and **AGL Games Division** management.

7. Warnings

We cannot stress how important it is that you document where you borrow code, content, etc. You will need to credit the sources of all your external content in your credit screen and in your documentation. Employees found in violation of company policy will be fired immediately and receive a zero on the project.

8. Grading

Since graphics is important to us, a good part of your final grade will be decided by the quality of the graphics of your game. The following breakdown will be used to grade the project:

<u>Graphics</u>	30
Overall look of the game	20
Fragment/vertex shaders	10
“Eye candy” graphics Effects	5

Quality of the Game	60
Quality of game play	15
Incorporation of management requests	15
Game ranking (comparison with other games)	15
Fun factor	15
<hr/> Presentation & deliverables	10
Game proposal	1
Early publicity materials	1
Work log	1
Webpage to describe your game	1
Final publicity materials	1
XBOX 360 and Windows executables	5
Individual performance score	10
TOTAL:	110

This project will be worth 35% of your final grade.

9. Extra credit

For those teams that really want to impress upper-management, we encourage them to pursue the following extra credit to their projects:

Porting your game to Zune – It would be pretty cool if you can port your game to Microsoft’s handheld Zune player, something that XNA 3.0 will allow you to do. Of course, you will need to figure out the Read more about this on the XNA Creator’s Club website at <http://creators.xna.com/>. (30pts)

Other extra credit – We will look favorably on teams that put extra effort to make a top-quality game. Exampled include composing a cool music track for the game, making intro cut-scenes, or making a world that looks visually stunning. Extra credit will be assessed at the discretion of **AGL Games Division** management and will be based on the amount of effort and the final quality of the final submission.

10. Tips & Advice

This will be the biggest programming assignment you work on this semester, and for many of you this might be the biggest project you have ever worked on. We expect this project to take approximately 300 to 400 man-hours of work to pull off for an “A” quality project (for the entire team), so in the first week you should budget your time appropriately by taking the total amount of time and dividing by the number of team members and the number of weeks remaining until December 6 to determine how many hours you should be working on this per day or per week. Of course, this is just a rough

rule of thumb, since you might be an expert coder and can work extremely efficiently. Check out some of last year's projects for inspiration.

11. Keys to Success

Since this project is more “open-ended” than previous projects in the class, you will be required to do a lot of learning on your own. We will list useful resources on the project webpage to show you where you can find information. If you find other resources, please let us know so that we can add them to the website. In addition, we encourage you to look at games written by students at other universities under the same time constraints so that you can see what is achievable in a short amount of time. Don't be intimidated by what other people are doing or have done, but look at their games as a source of inspiration. After all, they demonstrate what a small group of talented people like you can do in a short amount of time if they are motivated enough. The game you will develop will be part of your portfolio you can show companies when you graduate.

To assist you with the project, we have asked our more senior developers (students who have taken this class before and then took the advanced class) to work with you and give you advice on your projects. Please communicate with them through the forum.

12. Final words

Get started soon, and feel free to speak to your manager or your mentor developers if you have questions or if you get stuck. Good luck!

