# Computer Science 474 – Computer Graphics

**Spring 2014**

**Final Project - 400 Points**

**Design Due: Lesson 35, Friday, 25 April, 2014 @ 2300**

**Code Implementation Due: Lesson 40, Friday, 9 May, 2014 @ 2300**

**Demo and Pizza Party: Friday, 16 May, 2014 @ 1200**

# Help Policy

|  |
| --- |
| **AUTHORIZED RESOURCES:**   Any, except another cadet’s program.  **NOTES:**   * Never copy another person’s work and submit it as your own. * Do not jointly create a program. * You must document all help received from sources other than your instructor or instructor-provided course materials (including your textbook). * DFCS will recommend a course grade of F for any cadet who egregiously violates this Help Policy or contributes to a violation by others. |

# Documentation Policy

|  |
| --- |
| * You must document all help received from any source.  Unless quoting directly or paraphrasing, you do not need to document your course text, lectures, or any other course materials provided by your instructor. * Each documentation statement must be specific enough that it explicitly describes **what** assistance was provided, **how** it was used in completing the assignment, and **who** provided the assistance. * If no help was received on this assignment, the documentation statement must state “NONE.” * If you checked answers with anyone, you must document with whom on which problems.  You must document whether or not you made any changes, and if you did make changes you must document the problems you changed and the reasons why. * Vague documentation statements must be corrected before the assignment will be graded, and will result in a 5% deduction on the assignment. |

# Learning Objectives:

# To apply your knowledge of computer graphics and WebGL programming to an independently developed software system.

# Project Description

Using your knowledge of computer graphics and WebGL, develop a computer game that uses 3D graphics. Due to time constraints, the game will not be complex, but it should demonstrate all of the basic elements of 3D computer graphics. See the grading criteria for more details on the requirements.

You are free to develop almost any type of game -- as long as it does not include obscene, vulgar, or sexual content. If you are struggling to come up with a good idea for a game, consider the list of games genres at <http://en.wikipedia.org/wiki/Video_game_genres>.

Some possible game ideas include:

* 3D Pong (3D ping-pong) (Requires some math to calculate the intersection of a ball with a paddle)
* 3D Asteroids (Requires some math to calculate the intersection of a “ray gun” with a 3D object.)
* 3D Pac-man (e.g., <http://www.flashpacman.info/pacman-3d-pacman.php> )
* 3D Lego Builder (users build things out of Legos) (e.g., <https://www.buildwithchrome.com/builder> )
* 3D Artillery (have cannons that shoot at things or each other) (requires math to calculate when a cannon shell hits a target.)
* Bloxorz (See <http://www.miniclip.com/games/bloxorz/en/> )
* Any other idea that is approved by your instructor.

# Project Description - Step-By-Step

**Step 1**: Design a computer graphics 3D game. The game design includes the user interface, the 3D objects in the game, and the game logic.

**Step 2**: Create the 3D models for your game using Blender. Add texture maps where appropriate.

**Step 3**: Animate your models according to your game's logic.

**Step 4**: Add camera motion appropriate for your game.

**Step 5**: Add lighting effects and texture maps appropriate for your game.

**Details**

**Step 1**: Design a computer graphics 3D game. The game design includes the user interface, the 3D objects in the game, and the game logic. The game can be an implementation of an existing game, or it can be a totally new game from your imagination.

Since we have a limited amount of time for this project, your game needs to be scoped to a manageable size. Create a list of all the functionality you would like your game to have. Then prioritized your list to create a subset of the game logic that would allow a user to play a basic version of your game. Your list of functionality should contain 2 lists:

1. the functionality your game will have when you submit it for grading, and
2. the functionality that you would add to the game later to make it totally finished.

**The prioritized list of functionality is due by Friday, 25 April @ 2300**. Your instructor will provide you with a critique of your game proposal as quickly as possible. Your instructor may remove functionality from your list to make your project more realistic in scope, or may add functionality to your list to make your project meet the expected project scope.

**Step 2**: Create the 3D models for your game using Blender. Add texture maps where appropriate. Export your Blender models as OBJ files.

You are not allowed to use models create by 3rd parties unless you are given explicit permission by your instructor.

Remember that any objects that "move" must be distinct objects. For example, if you create a fish, and you want the fish's fins to move, the fins need to be distinct objects separate from the fish's body. And if you want the fish's body to bend, you must make the fish's body out of several distinct pieces.

**Step 3**: Animate your models according to your game's logic.

Remember that in the browser you do not have control over the refresh rate of the canvas window. You simply ask the browser to call your “animation function” as soon as it can. Make sure your movement and rotations are based off of the elapsed clock time between re-draws.

**Step 4**: Add camera motion appropriate for your game.

Your code should include at least four appropriate camera movements. In some games it would be appropriate to animate the camera and have it follow the action or character of the game.

**Step 5**: Add lighting effects and texture maps appropriate for your game.

You must have at least one light in your scene that changes its position over time. All other lighting effects are optional.

You must have at least one object in your game that is texture mapped.

**Notes and Suggestions**

* Please let me help you **when** you get stuck! Please don’t hesitate to get help.
* Your game will most likely **not** turn out as "slick" or as "finished" as you initially envisioned. THIS IS OK! I am expecting a reasonable effort on your part. I am **not** expecting a commercially viable game!
* When I grade your work I will play your game AND I will read your code. Your code should be documented and adhere to good coding practices (i.e., good variable names, correct code indention, modular, etc.)

# Turn-in

Electronic turn-in: (at <http://dfcs-moodle> on the course web site). Please submit all files needed to execute your game web page. If the number of your files exceeds the upload file limit, please submit a single zip file that contains all of your files.

**Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**CS474 – Final Project - Cut Sheet**

**400 Points**

|  |  |  |  |
| --- | --- | --- | --- |
| **Grading Criteria** | **%** | **Pts** | **Comments** |
| Step 1: Game design | 15% | 60 |  |
| Step 2: Modeling | 20% | 80 |  |
| Step 3: Animation and game logic | 20% | 80 |  |
| Step 4: Cameras | 20% | 80 |  |
| Step 5: Lighting and Texturing | 20% | 80 |  |
| Software Design and Good Coding Style | 5% | 20 |  |
| **Total** |  | **400** |  |
|  |  |  |  |
| Poor or missing documentation (-5%) |  | -20 |  |
| Late Penalty Cap  (25% per 24 hour period) (300, 200, 100, 0) |  |  |  |
| **Final Grade** |  |  |  |