

Midway Math

Game Design Document

RELEASE
December 18, 2003

© 2003 Austin Krauss

Table of Contents

Disclaimer.....	- 1 -
Game Overview	- 1 -
How should the game be unique?.....	- 1 -
How is it different from other games?.....	- 1 -
What sort of control should the player have over the world?	- 1 -
Development Specification	- 1 -
Development Schedule	- 2 -
Production Team Description	- 3 -
Production Tools.....	- 3 -
Target Audience	- 4 -
Game Mechanics	- 5 -
Camera	- 5 -
User Interface Design.....	- 5 -
Control Summary.....	- 5 -
Game Play Details.....	- 6 -
Storytelling.....	- 6 -
Physics	- 6 -
Game Elements	- 7 -
Items (data dictionary).....	- 7 -
Character Bibles	- 7 -
Story Overview.....	- 8 -
Storyboard	- 9 -
Game Progression	- 10 -
Flowchart.....	- 10 -
Level and Scene Details.....	- 11 -
Carnival Game Level: “Duck Shoot”.....	- 11 -
Carnival Game Level: “Crate Knockdown”.....	- 12 -
Bibliography	- 12 -
Appendix: Using 3D Studio MAX 4	- 13 -
Exporting	- 13 -
Model Transformations.....	- 13 -
Rotation and Scaling Transformations.....	- 13 -
Translation Transformations	- 14 -

Modification Log

Name	Date	Rev #	Change
Austin	10/3/2003	01	Created
Austin	11/15/2003	02	Incorporate ideas from Game Treatment document
Austin	11/18/2003	03	Remove titles that aren't incorporated in game
Austin	11/22/2003	04	Added data dictionary items
Austin	11/29/2003	05	Remove game elements that won't be implemented
Austin	11/31/2003	06	Remove “Duck Shoot” game level
Austin	12/03/2003	07	Remove cut scene details
Austin	12/09/2003	08	Miscellaneous cleanup
Austin	12/16/2003	09	Final review, more cleanup
Austin	12/18/2003	10	Create .pdf format, mark for release

Disclaimer

This document has been created as part of CIS 587 – Computer Game Design and Implementation, taken Fall Term, 2003. This class is intended to partially fulfill the requirements for a Master’s Degree in Software Engineering at the University of Michigan – Dearborn. All portions of this document are hereby owned and copyright 2003 by Austin Krauss, unless otherwise noted.

Game Overview

This document contains the essential design components of a 3D game that teaches children basic mathematics. The game titled *Midway Math*, is intended to have a carnival-like atmosphere to incorporate mathematics education with carnival games commonly found on the midway.

The purpose of *Midway Math* is to exercise knowledge of the software engineering process and object oriented development as well as demonstrate game design principals learned in CIS 587.

How should the game be unique?

Midway Math will be unique from other games of the same genre by providing the player with an arcade/action type of game play at the same time as working as a math learning tool. *Midway Math* will attempt to eliminate the “boredom” factor of simply solving math problems by providing a game with additional game play and story goals (such as winning prizes).

How is it different from other games?

Other edutainment games often stick to a 2D sidescroller type of interface. *Midway Math* provides a first person point of view in a 3D setting, allowing the player to associate the game world with many popular console games that are currently on the market. With this form, the child may not dismiss the game as “childish”.

What sort of control should the player have over the world?

The player’s movement is limited to the current carnival game that is being played. The player can view a limited range, and interact with the game world in a limited fashion, according to the particular carnival game.

Development Specification

As mentioned above, the game will be played in a 3D world environment. In order to accelerate development, the existing 3D engine developed for the *Speedy Delivery* project will be reused and adapted for *Midway Math*. Below are the core technical specification details that *Midway Math* will be built around:

- Created for Win32 platforms, DirectX 8 or better
- Coded using C++
- Proprietary 3D Engine
- Freeware 3rd party Physics Engine (*Tokamak* from Tokamak Limited)
- 2D pre-rendered backgrounds
- Freeware 3D objects from the web
- Possible professional 3D models purchased from Turbo Squid (www.turbosquid.com)

Development Schedule

The software engineering methodology to be followed will be one that loosely resembles rapid prototyping. With an existing engine already providing basic 3D rendering, scene management, and input functionality, work can begin towards the beginning of the development cycle on content creation.

Midway Math has a firm deadline date of December 19th, 2003. Major milestones (red) and minor milestones (black) are outlined in the schedule below:

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
26	27	28	29	30	31	1
November						
2	3	4	5 Design Work	6	7 Design Work	8 Design Work
9 Design Work	10	11 Game Treatment Due	12	13	14 Engine Analysis	15 Core Engine Work
16 Core Engine Work	17 Content Creation	18	19	20	21 Content Creation	22 Content Creation
23 Physics Engine	24	25	26	27	28 Physics Engine	29 Physics Engine

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
30	1	2	3	4 Physics Engine	5 Physics Engine	6 Physics Engine
December						
7 Physics Engine	8 User Interface	9	10	11 User Interface	12 User Interface	13 Sound Subsystem
14 Sound Subsystem	15 Sound Subsystem	16 Debug	17 Test and Debug	18 Test and Debug	19 Project Due	20

Production Team Description

The production team will consist of only Austin Krauss to do the design, implementation, and testing of *Midway Math*. Additional play testing may be conducted by friends and relatives of the author. Credit will be given to individuals or companies that provide content for *Midway Math* and all third-party licensing agreements will be fully complied with. Please see the section titled *Bibliography* for reference materials used in project development.

Production Tools

Microsoft's Visual C++ 6 and Autodesk's 3D Studio MAX 4 will be the two primary production and content creation tools. The author possesses expertise with Visual C++ 6 and familiarity with 3D Studio MAX. The game will be designed and implemented using object oriented development techniques in the C++ programming language. Primary and secondary production tools are enumerated below along with their intended purpose:

Visual C++ 6

- Source code compiler to be used in conjunction with the DirectX 8 API

3D Studio MAX 4

- Used for designing game levels and objects

Adobe Photoshop Elements 2

- Used to create and modify background art, as well as edit model textures

Tokamak Physics API

- A collection of C++ objects to incorporate dynamic, real-time physics into the game

Pandasoft DirectX Exporter

- A 3D Studio MAX 4 plugin to export game levels and objects to Microsoft's DirectX model format

CodeWright 6

- A source code editor providing more functionality than the Visual Studio IDE

Concurrent Versions System

- Version control system used for controlling game source code and content

Microsoft Word XP

- Document creation application

Microsoft Visio XP

- Technical design application (flowchart modeling)

Adobe Acrobat 5

- Document standardization formatting

Target Audience

The target audience for *Midway Math* will be children in the 2nd through 4th grades. This roughly equates to children in the age range of 6 to 9 years old.

Because the game will have a carnival-theme, with the central story focusing around the player trying to win prizes for his or her little sister, the game may be more geared to young boys rather than girls. However, the game story will be gender-neutral so as to avoid alienating young girls that are interested in playing *Midway Math*.

Game Mechanics

The heart of *Midway Math* will focus on the player trying to solve as many mathematics problems as possible in the given amount of time. To keep the game interesting and the child entertained, the player will play carnival games to solve the math problems.

Core game play will consist of the player trying to solve a wide range of basic math problems. Each level will be presented in a 3D world, through a first-person perspective. Menu selections and splash screens will be 2D bitmap animations and menus.

As each level is completed, the player will be given positive feedback for good performance, or encouraging feedback and hints for less than average performance. Negative feedback will be avoided in order to keep the learning entertaining and to avoid player frustration.

Camera

The camera will be from a first person perspective at a fixed location in the game world. The play will have a limited range of viewing.

User Interface Design

The main game screen will consist of the player's view of the game world, the math problem to be solved, and the possible answers to the problem. Concept art showing the main game interface is available in a subsequent section in this document.



Main Menu Mockup Screen

Control Summary

The player will navigate the menu screens using the arrow keys on the keyboard and the Enter key to make a menu selection. The mouse will be the primary input device used for in-game interactions. The ESC will also be used in game to pause the game and bring up a menu screen.

The left mouse button will throw the ball, and the mouse will be used to aim the ball.

Game Play Details

Game play will consist of a single game level. The objective of the game is to knock down a pyramid of crates onto the ground. Crates were decided to be used instead of bottles to due the availability of model artwork.

The player must hit the crate that displays the correct answer in order to make it move. If the player hits a crate with an incorrect answer, and buzzer sound will be played and the ball will disappear. The number of incorrect and correct answers will be displayed on a score screen at the conclusion of the game.

Storytelling

The game story will play a lesser role in *Midway Math* than other games. The basic story line will follow the trials of a child playing games on the midway to win prizes for his or her kid sister. The story line will not be present in the game, although it may be outlined in the game's manual to present an overview of the game.

Physics

Physics in the game world will be developed with the third-party *Tokamak* physics engine. *Tokamak* provides a physics and collision-detection API that realistically models gravity, mass, velocity, collision, and acceleration.

Game play will be affected directly by the physics engine. For example, the player can play upon the physics of the real-world by specifically targeting points on the crates that will result in a collapse in the pyramid in a quicker amount of time.

Game Elements

Items (data dictionary)

Game Item	Purpose
Crates	A variety of crates for the player to throw the ball at
Ball	A 3D sphere the player will use to interact with the game world
Skybox	A set of high-resolution textures used to surround the game world with an out-doors like atmosphere
Background Image	A low-resolution texture used to provide the image when the main menu is displayed
Menu Items	A set of 2D textures made programmatically to be displayed when the player needs to interact with the game menus
Ground Geometry	A 3D geometric box textured with a "grass-like" texture to serve as the ground in the game world
Sound Effects	Various sound effects to be played during game events and at random intervals

Character Bibles

The character set is limited to the player and the kid sister. The player is never represented as an on screen figure. This is because the game is intended to be gender-neutral and played in the first person.

The kid sister will be shown in the game manual if appropriate. The kid sister is intended to have a cartoon-like look as per the concept art below:



Kid Sister Concept Art

Story Overview

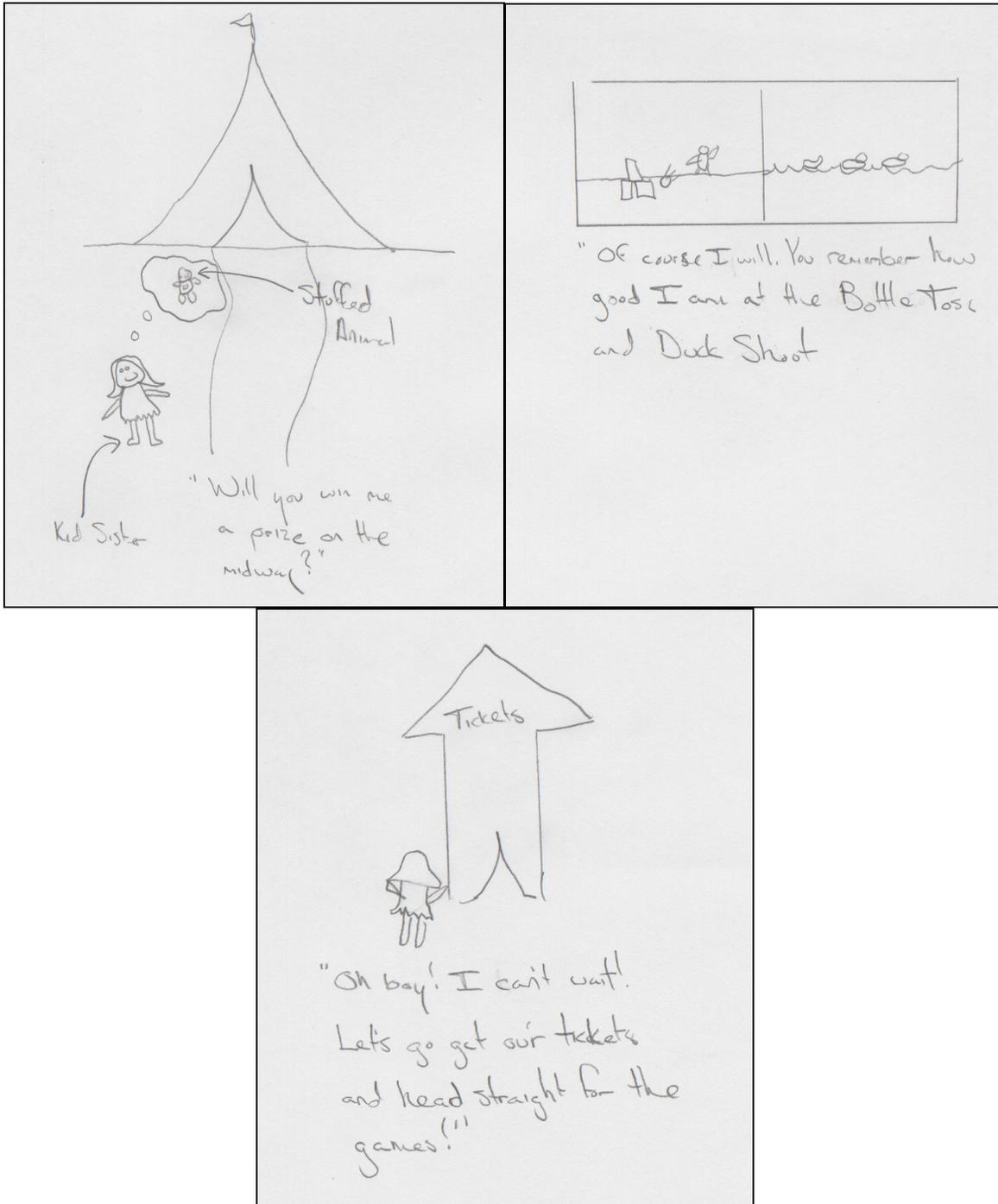
The story line is intended to be basic and, more importantly, gender neutral. To accommodate gender neutrality, a picture of the main game player will never be shown during game play and the player will never be referred to in the third person as “he” or “she”.

The game story is very simple: You and your kid sister have decided to spend the day at the local carnival. You’ve come to the carnival’s main midway and test your skills at some of the many games to try and win prizes for your sister.

Storyboard

Note: In-game cut scenes and animation are currently unimplemented. The storyboard presented here may represent what will go into the game manual and/or a future version the game

The introduction animation is intended to set the game story for the in-game reward sequences. Below is concept art detailing the introduction animation:



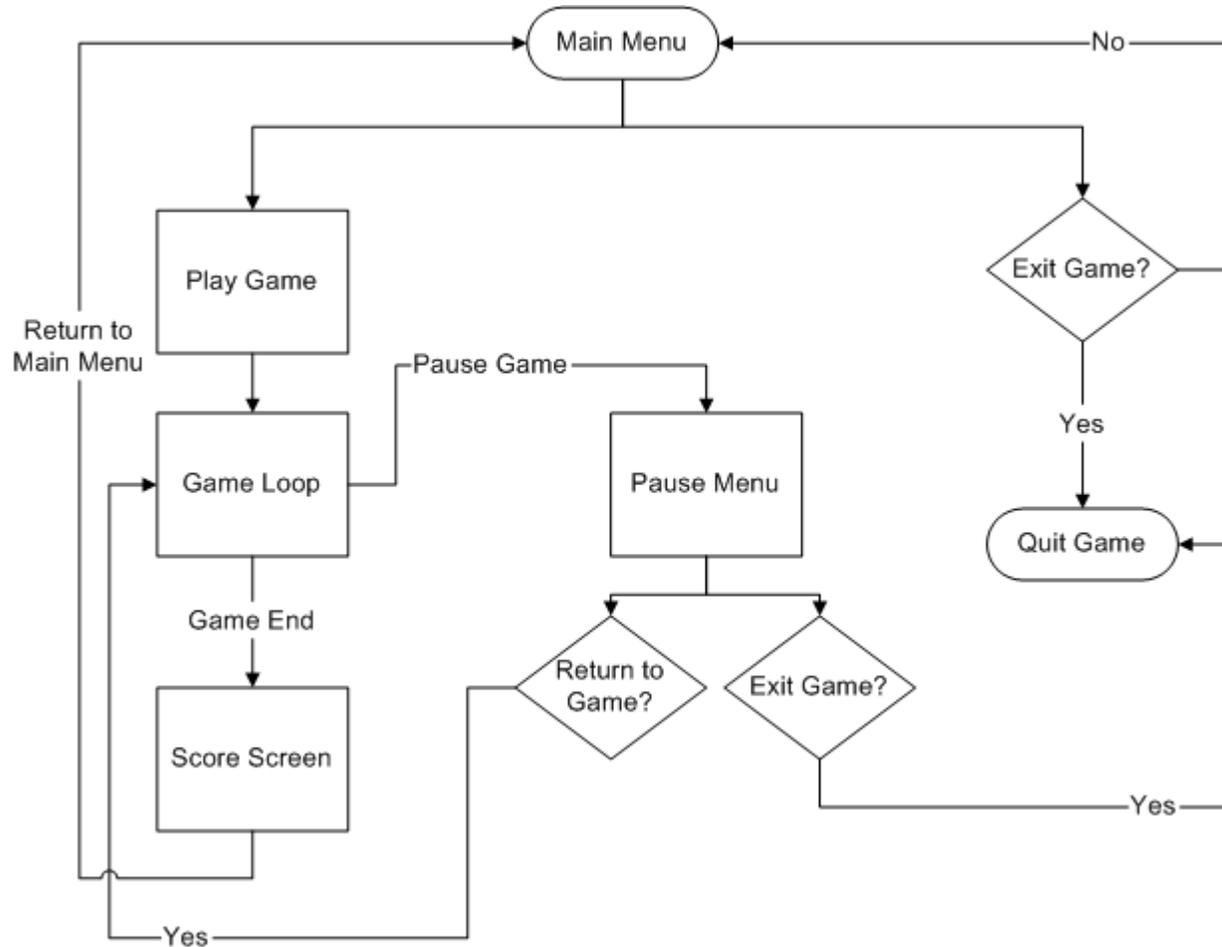
Intro Animation Story Board

Game Progression

Success in *Midway Math* will be measured by the player's ability to solve the math problems with the fewest incorrect answers as possible, and with the fewest number of balls thrown.

Flowchart

The flowchart below represents the main flow of the game. It should be noted that the process titled "Game Loop" is a standard representation of updating the game state, rendering the world, and checking for user input.



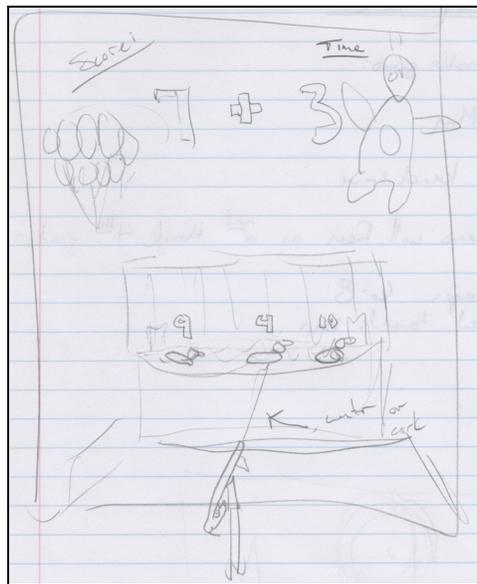
Level and Scene Details

Carnival Game Level: "Duck Shoot"

Note: This level is currently unimplemented due to time constraints

In this level the player is presented with a 3D scene of duck decoys swimming around a circle. The player wields a water pistol or cork gun and shoots the ducks which have the correct answer to the particular math problem which is displayed above.

The goal is to shoot all the duck decoys within the time period to win the main prize.

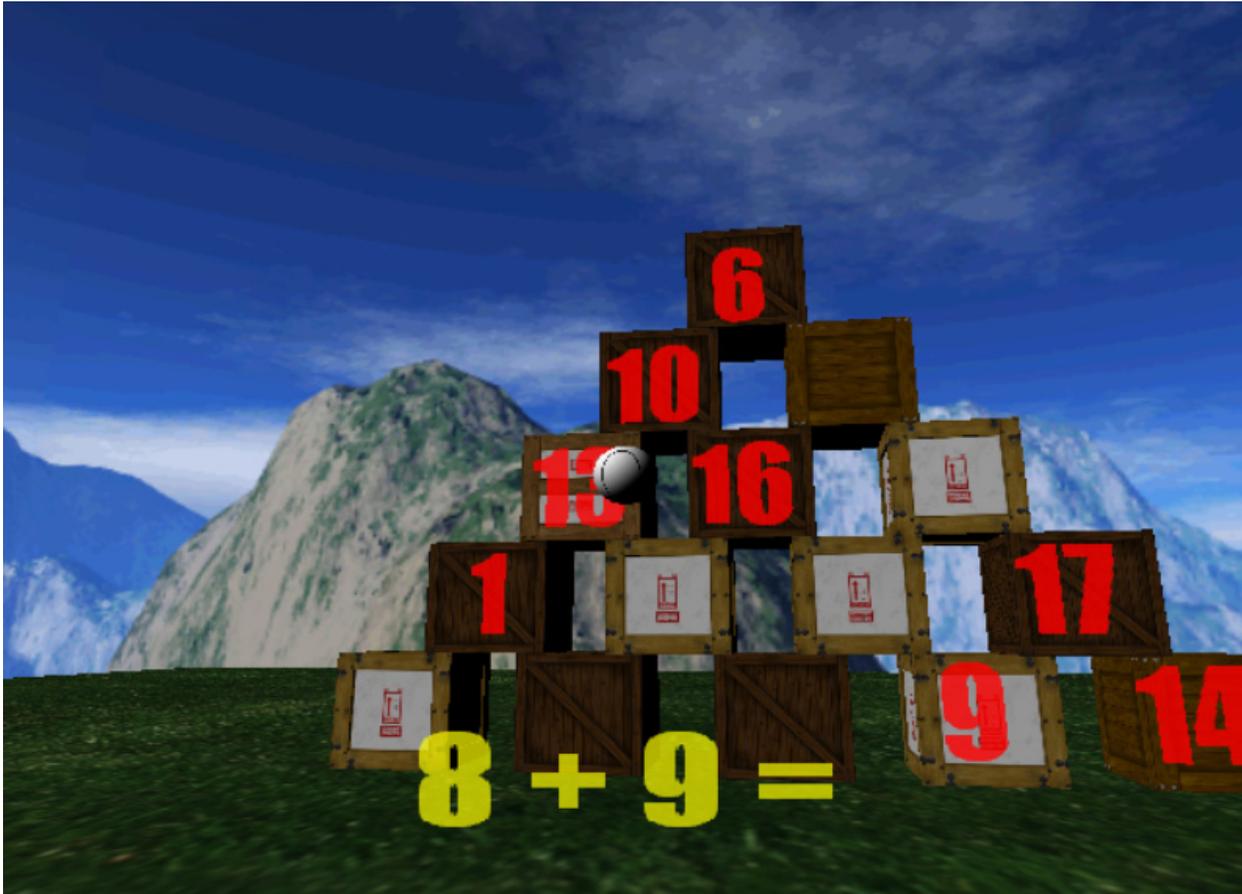


Duck Shoot Concept Art

Carnival Game Level: "Crate Knockdown"

This game starts by displaying a large pyramid of crates stacked on top of each other. Each crate has a number on it and the player is to "throw" a ball at the selected crate using the mouse. If the number on the crate is the correct answer to the math problem, the crate falls down.

The goal is to knock all of the crates using the least number of balls as possible and with the fewest incorrect answers.



In-Game Screenshot

Bibliography

The following reference materials were used during the development of *Midway Math*:

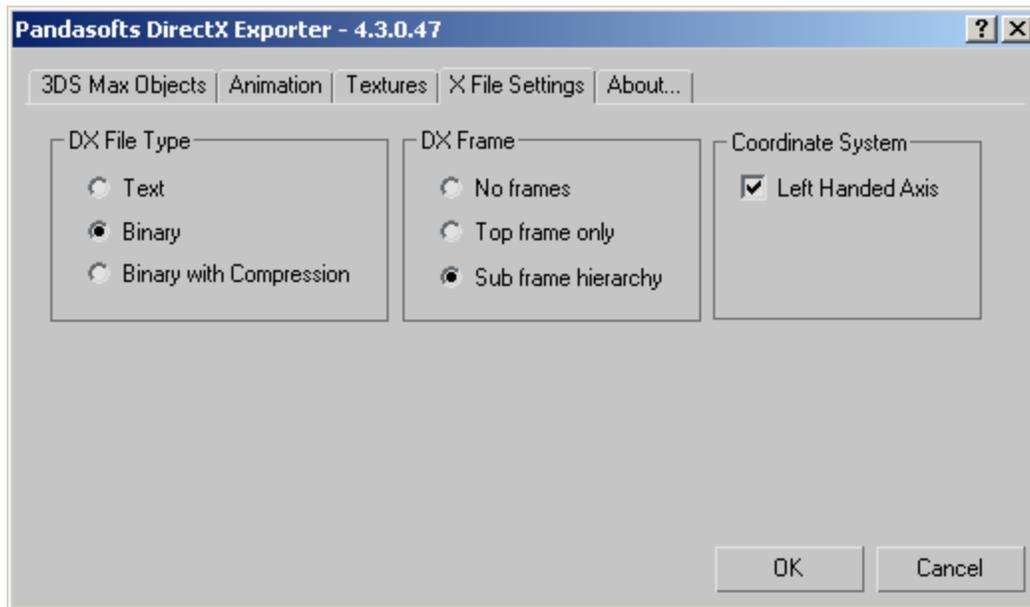
- Game Institute Course Material
- DirectX 8 SDK Samples and Demos
- David Eberly's 3D Game Engine Design book
- Turbosquid.com 3D assets
- Clipart.com 2D assets

Appendix: Using 3D Studio MAX 4

This section is intended to serve as a general guideline to importing geometry created in 3D Studio MAX into *Midway Math*. The following notes were taken during actual use.

Exporting

Models are intended to be exported as a hierarchy. Thus, a special exporter must be used to ensure the model hierarchy is not lost during the export process. The exporter to be used is the Panda Direct X Exporter Plugin. This exporter has various options for exporting the mesh. However, in order to keep the mesh hierarchy with the correct transformations matrices, it is recommended that all meshes be exported as to the screen below:



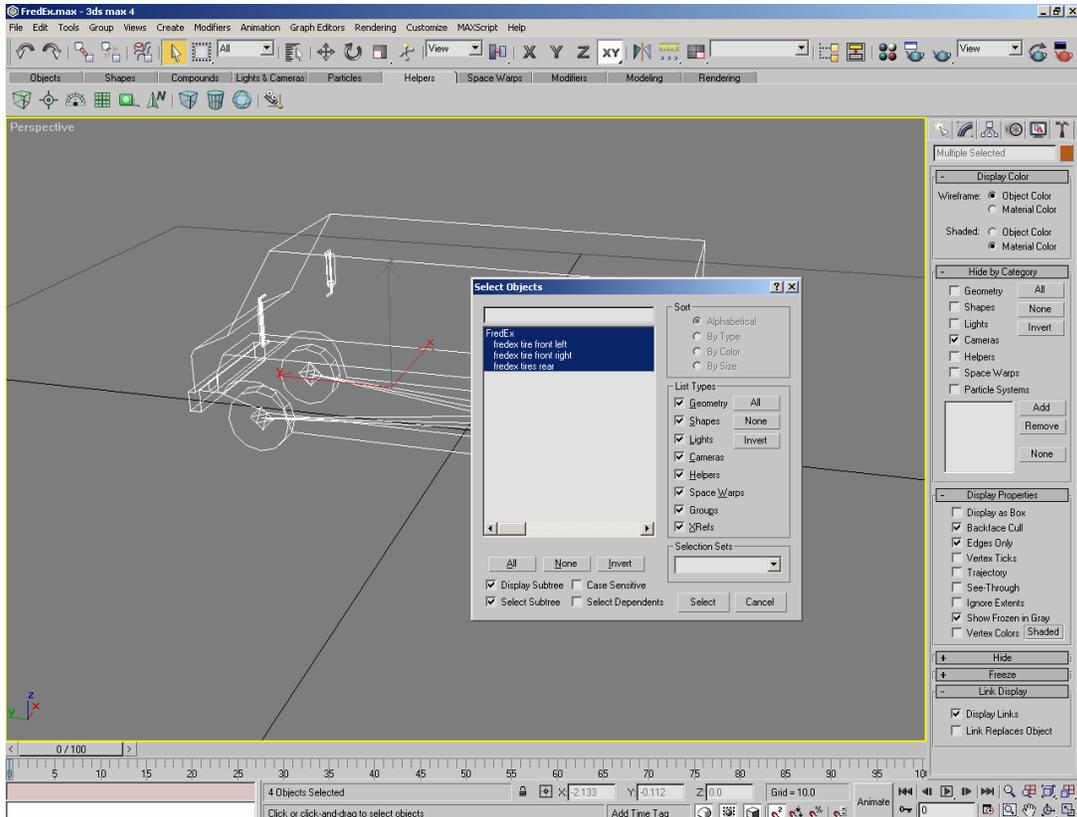
Recommend X File Export Settings

Model Transformations

When constructing the game world, it is only natural that geometry will undergo scaling, rotation, and translation transformations. Positioning objects close to one another and in specific starting places in the game world require this. Therefore, the following rules must be followed to correctly export the mesh hierarchy and transformation matrices:

- Rotation and Scaling Transformations
 1. Make a note of the model hierarchy, that is use the “Select By Name” command to view the model hierarchy.
 2. Select all objects that compose the model, and press the “Unlink” button. This will remove all parent/child relationships in the model.
 3. Perform the desired transformation.
 4. Perform a reset of the transformation matrices by selecting the entire model and pressing the “Reset XForm” button on the Utilities panel. This will set the rotation matrix to be the identity matrix and set the scale of each axis to be 1.0.

5. Finally, relink the model into its original hierarchy using the “Link” command.
- Translation Transformations
 1. Perform translations as needed; the transformation matrix exported will contain the coordinates of the parent mesh in world space. Transformation matrices of child meshes will contain coordinates in the space of their parent.
 2. The *Midway Math .x* file loader will correctly interpret these coordinates into parent/child relationships in the game engine.



Viewing the mesh hierarchy