GAM150S14-D

Spring 2014

Neko Means Cat

|  |  |
| --- | --- |
| Tai Der Hui - Producer | Adam Rezich - Technical Director |
| Eduardo Gorinstein - Lead Designer | Samuel Valdez - Product Manager |

Game School Simulator 2015

Technical Design Document

Table of Contents

[Overview 2](#_Toc379780340)

[Graphics Implementation 3](#_Toc379780341)

[Coding Methods 4](#_Toc379780342)

[Project Layout 4](#_Toc379780343)

[Assets Directory Layout 5](#_Toc379780344)

[Source Control 5](#_Toc379780345)

[Style Guide 5](#_Toc379780346)

[Debugging 6](#_Toc379780347)

[Tools 7](#_Toc379780348)

[Visual Studio 2012 7](#_Toc379780349)

[NekoPak 7](#_Toc379780350)

[Technical Risks 8](#_Toc379780351)

[Appendix A: Interface Flow 9](#_Toc379780352)

[Appendix B: Art Requirements 10](#_Toc379780353)

[Appendix C: Audio Requirements 11](#_Toc379780354)

# Overview

*Game School Simulator 2015* is a graphical interactive game for Windows.

# Graphics Implementation

The graphics of the game are entirely sprite-based, loading images from PNG files and displaying them on the screen using the Sprite component.

# Coding Methods

## Project Layout

The project directory for *Game School Simulator 2015* is laid out as follows:

|  |  |
| --- | --- |
| / | Root directory. Contains Visual Studio solution file, README.md, and any scripts that might need to be frequently run. |
| AlphaEngine/ | Contains Alpha Engine header files. |
| bin/ | Visual Studio project is set to output game binaries here. |
| etc/ | Any assorted files related to the project such as this document are stored here. |
| fmod/ | Contains FMOD header files. |
| Game School Simulator 2015/ | Project folder for the game itself. Contains all C source and header files, as well as Visual Studio project files. |
| data/ | Contains all assets for the game. (See “Assets Directory Layout” on page 4 for more information.) |
| lib/ | Contains all dynamic library files to be included in the build of the game (copied over automatically to output directory), as well as the static object library files for the libraries used in the project. NekoEngine outputs to this directory. |
| NekoEngine/ | Project folder for the game engine. Contains all C source and header files, as well as Visual Studio project files. |
| NekoPak/ | Project folder for NekoPak[[1]](#footnote-1). Contains all C# source files and Visual Studio project files. |
| tmp/ | Temporary directory used by Visual Studio to build the various projects. |
| tools/ | Contains binaries of all custom tools used in development. Output directory for NekoPak. |

## Assets Directory Layout

Game assets for *Game School Simulator 2015* are stored in a folder called “data” in the “Game School Simulator 2015” subdirectory of the project root directory. Assets are then further organized into subdirectories by type, as follows:

|  |  |
| --- | --- |
| bgm | Background music files (.MP3 or .WAV) |
| cfg | User configuration files (.CFG) |
| sfx | Sound effect files (.MP3 or .WAV) |
| spr | Sprite image files (.PNG) and sprite definition files[[2]](#footnote-2) (.SPR) |
| tex | Sprite map texture files (.PNG) |
| txt | Text files (.TXT) |

These directories may contain any number of subdirectories as the designers see fit; the asset pipeline will load them in automatically[[3]](#footnote-3). When a debug build of the project is compiled, the entire “data” directory is copied into its output folder. For release builds of the project, the assets are instead packed into a single “data.pak” file using NekoPak. (See page 7 for more information on NekoPak and how it works.)

## Source Control

Source control is implemented using Git. Repository hosting is provided by Bitbucket, hosted in a private repository at <https://bitbucket.org/adamrezich/neko>. Backups are kept on a local server, as well as on DigiPen version control servers at <https://git.digipen.edu/projects/nekomeanscat>.

## Style Guide

Fill me in.

# Debugging

Fill me in.

# Tools

## Visual Studio 2012

Fill me in.

## NekoPak

Fill me in.

# Technical Risks

Fill me in.

# Appendix A: Interface Flow

Fill me in.

# Appendix B: Art Requirements

Fill me in.

# Appendix C: Audio Requirements

Fill me in.

1. See “NekoPak” on page 7 for more information. [↑](#footnote-ref-1)
2. See TODO: FILL IN [↑](#footnote-ref-2)
3. See TODO: FILL IN [↑](#footnote-ref-3)