

Programmers Guide

Introduction

This Programmers Guide is designed to be read alongside the Technical Manual, Style Guide, Glossary of Terms and the source code (which will be referenced herein). This document assumes the reader is somewhat comfortable with programming, and has an understanding of software design paradigms. It is also highly recommend the reader familiarise themselves with Professor Albertons Algorithmic Adventures (herin refered to as “the application”).

SFML and C++

The application is programmed in C++, using the SFML API. If you’re reading this, you should already have access to the source code and are familiar enough with C++ to get started.

SFML is well a well documented API which includes an OpenGL wrapper (for drawing geometric primitives), and an interface for: sound, window management, action listeners and graphics.

For instructions on installing the SFML library, or a detailed guide to the interface see <http://www.sfml-dev.org/>

If you’re approaching this document with the intention of adding to the code base, check the following list for the most useful interfaces:

[MainLoop.h](#)
[MainMenu.h](#)

The code that drives the creates the render window can be found in:

[MainLoop.cpp](#)

State Driven Design

The application uses a state driven design methodology to separate the program data into discrete structures.

Game States Explanation and Implementation

The aforementioned design methodology is achieved using enumerated types. Take for example, the interface the class MainLoop in the interface file MainLoop.h:

```
enum ProgramState {    Uninitialized,
                      ShowingSplash,
                      ...    };
```

Combined with the relevant function definitions:

```
static void ShowSplashScreen();
```

And

```
// Should never happen
if(programState != Uninitialized){
    return;
}
```

The above clearly demonstrates two game states. As per the Coding Style Guide, the entire interface to the class MainLoop is self commenting, as are the enumerated ProgramStates above.

The Main Loop

```
void MainLoop::RunLoop(){
    sf::Event currentEvent;
    while(MainWindow.pollEvent(currentEvent)){

        switch(programState){
            case MainLoop::Uninitialized:{
                break;
            }
            case MainLoop::ShowingSplash:{
                ShowSplashScreen();
                break;
            }case ...
        }
```

The main program loop consists of: An action listener (currentEvent), a render window, (MainWindow), and a switch statement of program states.

The Menu System

ResourcePath.hpp

The ResourcePath.hpp file stores the header information for all the .h files, as well as the C++ library files that are required for compilation.

ResourcePath.mm

The ResourcePath.mm file copies the contents of the applicationpath/resource folder into the resource path of the application bundle upon compilation. This ensures the sound and graphics assets are properly contained, and locatable by the application once it has been compiled into a binary.