Object Design Description
for the
Track and Control System

Version 1.0

Robert Moss, Aaron Periera, Matthew Shrago

March 5, 2014

Contents

1 Introduction

1.1 Object Design Trade-offs

1.1.1 Buy vs. Build

The decision to Buy vs Build can be made by:

- If it is possible for the software to be built within a reasonable time frame using little resources, it is better to build than buy.

1.1.2 Space vs. Speed

In the Track and Control System, space is not a major priority. With only settings and some other menial data being kept, speed is the most crucial; as a fluid motion experience is what it is striving for.

1.1.3 Delivery Time vs. Functionality

If the development of TACS is behind schedule, the more in-depth functionalities can be sacrificed at little expense to the overall project.

1.1.4 Delivery Time vs. Quality

If testing runs behind schedule, the software can still be released and updates can be released to fix bugs at a later date.

1.1.5 Files vs. Databases

Files would not be as beneficial as a database because the data for TACS is not voluminous. A database would provide a sufficient structure to organize users and their settings. In using a file, users can theoretically edit and thus corrupt the file and the data inside. An "off site" database would be more secure and less likely to be modified.

1.2 Interface Documentation Guidelines

Classes

- Class names should be in Pascal Case.
- i.e.: MoveWindow

Constants

- All constants should be entirely in Upper Case.
- i.e.: RESOLUTION

Identifiers

- Identifier names should be in Camel Case.
- i.e.: xPosition, yPosition, userSpeed

Local Variables

- Variables should be all lowercase, less than 8 characters long.
- i.e.: speed

1.3 Definitions, Acronyms, and Abbreviations

- Application Specific Definitions
  - TACS - Track and Control System
  - TM - Tracking Module
    - OT - Object Tracker
    - FRT - Facial Recognition Tracker
  - WCM - Windows Control Module
    - WGO - Windows Grid Organizer
    - WP - Windows Perspective
  - SM - Settings Module
- Industry Definitions
  - SRS - Software Requirements Specification
  - OpenCV - Open Computer Vision: An open source library for object tracking via the camera.
  - SQLite - A lightweight, low maintenance, self contained local database.
  - DB - Database
  - RGB - Red, Green, Blue color values.
  - HSV - Hue, Saturation, Value.
  - API - Application Programming Interface
  - C++ - An object oriented programming language.
  - GUI - Graphical User Interface
  - QT - An API for building GUIs

2 Packages

2.1 Package Diagram

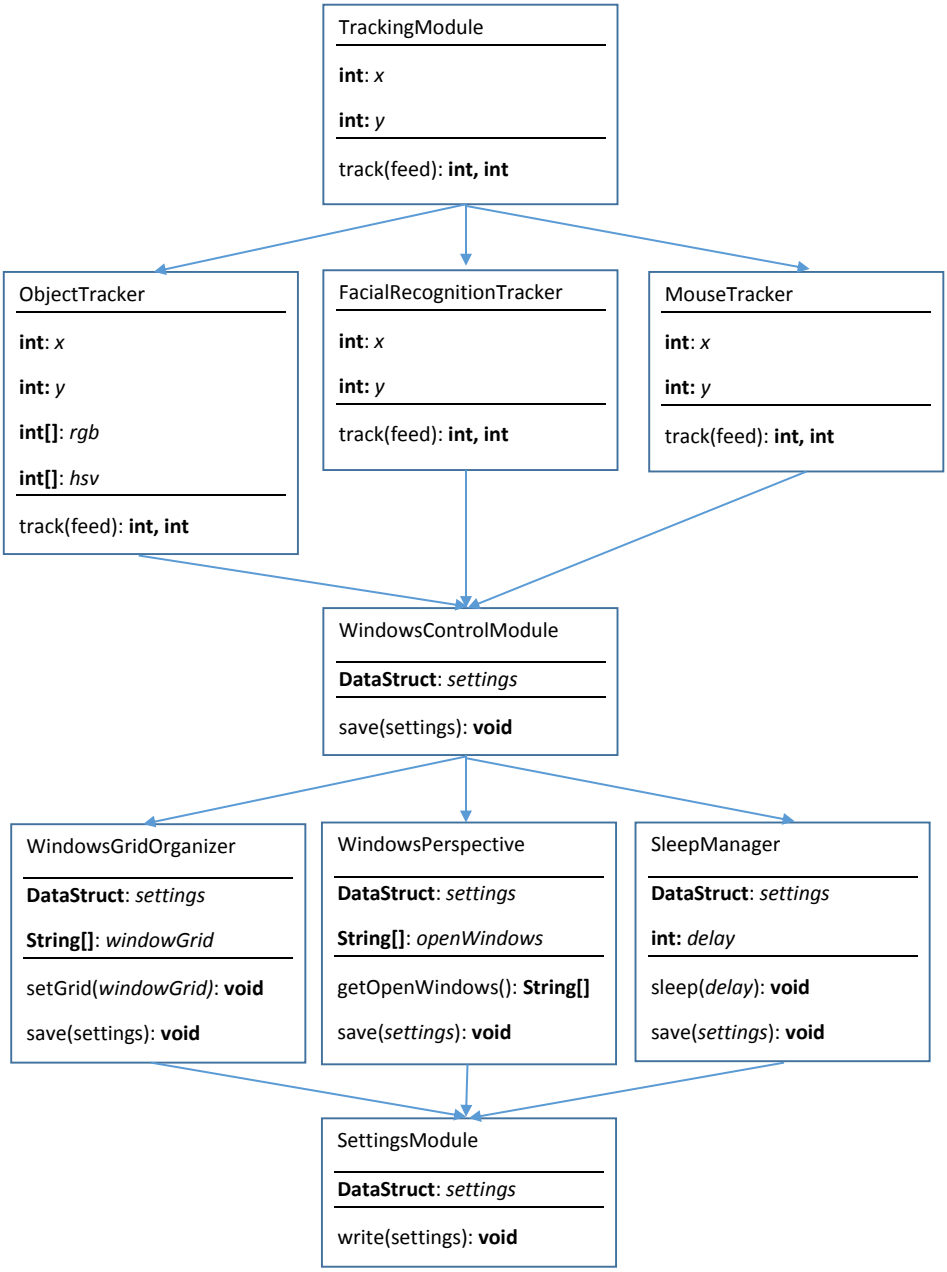
2.2 Package Definition

This software only has one package. This package is going to contain 1 main class and 8 subclasses.

3 Class Interface

3.1 Class Diagram

3.2 Decomposition Description



3.3 Class Definition