Spring
2013

Automated Sentry Missile Launcher

PROJECT REQUIREMENTS

Table of Contents

Introduction

Welcome comrades! Your services are required to design, document, build, and release an **Autonomous Sentry Missile Launcher (ASML)** that can acquire foe targets, eliminate them, while keeping friend targets safe. Your team (consisting of one to two other classmates) will define a software engineering process, specification, and implementation and will be graded on the quality of the design, implementation, and documentation.

Your team will be competing against other teams in the class with a large sum of extra credit to be awarded to the winning team. A competition will be held to reward the team that can acquire and eliminate targets on a battlefield the fastest.

Software Requirements

This document details the general requirements for the target detection system. You'll create an **ASML** that analyzes images captured through a web-camera, detects targets, displays the video on a graphical user interface (GUI), and fires foam missiles at foe targets.

Your **ASML** System will be written in C# using Visual Studio 2012 Ultimate. The GUI should be written using Windows Forms or Windows Presentation Foundation (WPF), your choice. The user interface should have the following: display for live video, start button, stop button, display of time elapsed since start, targets visible with the number of friends and the number of foes, while displaying the position of your sentry gun in degrees (Euler Angles - http://mathworld.wolfram.com/EulerAngles.html).

In the live video view, the **ASML** should display the targets that have been acquired. These should include a red outline for foes, and a green outline for allies. Live video should not hiccup or be unresponsive while the program is moving the sentry, unless the system is put into frame averaging mode. Frame averaging mode will average the last N video frames together. The live video should also support other image processing plugins for display purposes only (e.g. outline, black and white, pixelation, image subtraction etc.). It is because of this, that **ASML** must be multi-threaded. When the sentry is in operation, the stop button should kill any missile firing or sentry movement operations. The start button, when pressed and when the **ASML** system is idle, should reset the timer, and put the **ASML** into search and destroy mode. During the idle period, the **ASML** should still be identifying targets it just may not take action on them.

The display of time elapsed since searching should be displayed in total fractional seconds (56.7) where .7 is 700 milliseconds. If the total number of seconds exceeds or is equal to 60, then the system should display (01:00:12) where this displays units are 1 minute, 0 seconds, and 120 milliseconds.

Targets should be displayed in the live video, i.e. an enemy target should be outlined on the image displayed in the live video display. Targets should be detected using image processing techniques. The EMGU Image Processing Package (http://www.emgu.com/wiki/index.php/Main_Page) describes a framework you will have to write an adapter for to acquire and process images. This means your program should do some basic image processing routines.

As a fall back, your system should also be able to read target files, which contain the coordinates of targets on the competition course. These target files will be given in INI and/or XML formats. Your program should defer to using these coordinates if your imaging system is occluded.

Finally, your program should also be able to read and integrate plug-ins for the video acquisition, and software interface to any missile launcher hardware.

The sentry gun system should interact with a instructor supplied sentry gun from ThinkGeek.com (http://www.thinkgeek.com/product/8a0f/?srp=3). You may rent the sentry gun free of charge from the instructor. At the end of class all rented materials must be returned to the instructor or your team will fail the project. You may purchase your own sentry gun, however, it must be safe, legal, and comply with all requirements of the university, i.e. no firearms or compressed air. In fact, it is restricted to low power foam missile launchers.

The imaging system should interact with a cheap standard USB web-camera. You are required to supply your own web-camera for the project. Your imaging system of choice is up to you, e.g. Kinect or USB web-camera. However, it must be deployable to other systems (e.g. the instructor's computer).

Your system should be autonomous, that is no manual control should be available during competition. However, when idle, your sentry should be controllable manually through the GUI through four buttons (up, down, left, right). When any of these buttons are pressed, the sentry should move accordingly. The position display should be updated as the sentry moves. Holding any button should increase the speed of the sentry (i.e. accelerate the movement). When the system is eliminating targets the timer should be running. The timer should stop when the last missile is fired. Failing to stop the timer when this happens will result in time penalties.

Your system is allowed to recalibrate (fix the position and reset) when the start button is pressed. The timer should be running during this process. A sound should indicate when the system is finished recalibrating (e.g. "SEARCH AND DESTROY")

Project Requirements

Your team must define a software engineering process and complete a process template supplied by the user. Your team must define and agree on a coding standard. You must use a configuration management system such as SVN, git, Mercurial or Google Code. Software issues and tasks must be tracked using a project management system (TRAC, Redmine, Jira, Trello – https://trello.com/, or Google Code). You must supply your instructor access to your management tools of choice. You must also notify him of the location of your configuration management choice and instructions on how to interact with this system in your documentation.

Your final deliverable (code) should be compilable by the instructor. Do not include binaries (unless part of a 3rd library). You must not all licensing restrictions regarding these libraries and adhere to their specifications. Your electronic submission should comply with that stated in the

syllabus. Don't foul up by not following directions. You may always ask the instructor to validate compliance. A hardcopy of the source code (all including graphical user interface code) should be submitted before the start of the competition (this will happen on the final exam date). Screenshots of the GUI also need to be included in this documentation.

Collaboration

You may share code and collaborate across teams. However, you may not share entire parts of the project. For example, you may share how to interact with the sentry gun. Any collaboration should be documented in your project documentation. Keep in mind that you are competing with these other teams. So share at your own discretion.

NOTE: You may not use code without permission from another team. This is considered cheating and violates University rules. Violating these rules will result in an F in the course and reporting to the University.

Competition

Teams will compete against each other at the end of class. The competition will test for the best ASML system based on the system that can eliminate all foe targets the quickest. The competition will be broken up into several events, each of which increasing in difficulty (in terms of number of targets, target types, and target spacing).

Your team must compete in all events. Each event has a two minute time limit. Your system should stop after two minutes. You must fire all missiles by the end of this time limit. Failure to fire any missiles will result in a no score for the event and you will be penalized. If you are clearing your missiles from your sentry, your system must play a sound ("CLEARING MISSILES") loud and clear to avoid penalty. The sentry aim should be out to a void space away from the target area.

You can collect additional points by hitting foe targets. Missing a foe target will result in a small point deduction. Shooting at a friend target after acquiring it will result in a point deduction. Hitting friend targets will result in severe point deductions. Left over foe targets will result in point deductions, so don't just fire off missiles.

Foe targets will be red, friend targets will be green. Both sets of targets will be squares and circles. Note early that lighting will affect your ability to discern color. Test your system early to understand the restraints imposed on your design by these environmental factors.

Deployment

Your software should be submitted to the instructor as an installer prior to the competition. No testing will be allowed within an hour prior to the competition. You may submit your system early to make sure

it works on the test machine. Only one system will be used for testing. You may not run your code from any integrated development environments (IDE).

Details to be released at a later time
More information will be given at a later time on the competition.

Final Deliverable

1. Coding Standard
2. Requirements
3. Specifications
4. Project Management
    a. Software Process
    b. Risk Analysis
    c. Tasks, Issues, Stories
    d. Project Gantt Chart
    e. Configuration Management
5. Design
    a. User Narratives
    b. 4+1 Architecture
6. Design Pattern Analysis
7. GUI screenshots
8. Implementations and Code
9. Other documentation as discussed in class
    a. There will be no surprises to what these additional documentation are and will be readily discussed during the course if needed.

You will submit this as a single deliverable one per team.

Team Evaluations

You will be expected to evaluate your team members at the end of the project. You will be evaluating them on these areas:

1. Contribution
2. Technical Capabilities
3. Attendance to Team Meetings

Grading

You will be graded on the following:

1. Functionality
2. Quality and completeness of documentation
3. Meets Requirements
4. Execution of Process
    1. Measured by completeness of documentation
5. Has working software that competes in the competition

Schedule and Assignment

You will be given homework assignments throughout the semester that directly relate to the project. You are expected to submit these assignments individually. You will incorporate these assignments into your project as you go as a team. For example, while everyone will write an INI file parser, you will have to make a decision as a team which implementation gets integrated with the team's project.