Spring
2013

Project 2 – Search and Destroy
Due 4-18-2013

MEDIATOR, STRATEGY, OBSERVER, SEQUENCE, COMPONENT

COMPUTER SCIENCE 323 | WSU

Table of Contents

Administration

Points – 300 points

Due Date

The deliverables are due at the start of class on April 18th, 2013. Please **NOTE:**
ELECTRONIC VERSIONS OF THE SOFTWARE ARE ALSO DUE AT THE START OF CLASS.

NO LATE SUBMISSIONS, NO EXCEPTIONS, NO POST DEADLINE FIXES

Submission and Naming

Submit your project to the instructor by e-mail. Use the following file name format: *cpts323-2013-spring-project-02-groupName.pdf* Use your group's name. Failure to use this format will result in a deduction of 50%, and 10% for every submission after. You must submit a hard copy at the beginning of class, electronic version may be submitted after class. No handwritten assignments will be accepted. No late homework is allowed. This is a group assignment. Please only submit one hardcopy and one electronic copy. One person from your group should submit this homework. This email must contain the names of each student in the group. Hardcopies should be turned in on the nearest lecture day.

Code submissions should not include the bin and obj folders. You must include any dll references that your project uses to build.

Any code that does not build will not be graded. No post-submission (after start of class) fixes will be allowed for this project.

Collaboration

Group project, you may divvy up the project and responsibilities how you want. You are expected to work together.

Assignment Details

Software Requirements

You are expected to modify your graphical user interface (GUI) that can control and fire the Thunder Dream Cheeky missile launcher. The previous requirements are still required.

Previous Requirements

1. The software package must be able to also read a target description file in XML or INI format.
2. The GUI should display these targets with the following attributes
   a. Position X, Y, Z
   b. Friend / Foe
   c. Target Name
3. The GUI should display the version of the software at the top of the program
4. The missile launcher should be controllable from the GUI
   a. Move Up/Down
   b. Move Left/Right
   c. Fire a missile
5. The GUI should display your group's name.
6. Your program should be called "Asml-GroupName.exe" where Group Name is your group's name.

New Functional Requirements

7. Modes
   a. The GUI should allow the user to select a mode for search and destroy.
   b. The GUI should have a start button, that when pressed, the program should start in search and destroy mode.
   c. The GUI should have a stop button, that when pressed, will stop the search and destroy mode.
   d. Search and destroy mode is automatic, the turret should move automatically, the turret should fire automatically.
   e. When in Search and Destroy mode, the time elapsed since the start of this mode should be displayed on the GUI.
      i. When search and destroy is stopped, the timer should also stop.
      ii. When search and destroy is started, the timer should start at 0.
   f. Supported Modes
      i. Automatic Mode – search and fire at all targets
      ii. Automatic Mode – search and destroy only enemy targets
      iii. Automatic Mode – search and destroy only friend targets
8. Video
   a. The GUI should display a live video stream.
   b. The GUI should have a button to stop the live video stream

c. The GUI should have a button to start the live video stream

9. Missile Launcher Position Display
   a. The missile launcher should display the Phi and Theta angles of the missile launcher in the GUI
   b. The launchers orientation should be given in spherical coordinates (degrees) by phi and theta. An angle of (0, 0) should be pointing horizontal to the playing field. At a direction (x = 0, y = 0) to (x = 0, y = 1). Negative phi angles should be to the left of the reference point (x < 0). Negative theta angles should be referenced downward from where the launcher stops after the *command_reset* call in the object provided from class discussions.

Design Patterns

You must use the following design patterns:

1. Singleton
2. Adapter
3. Factory
4. Mediator
5. Observer
6. Strategy

Other Required Design Elements

You must also include the following:

1. An object to manage the list of targets
2. A target object that encapsulates target data

Required Interfaces

Your design should use the design principles discussed in class: abstraction and polymorphism where appropriate. Your missile launcher should realize the IMissileLauncher interface that is provided in the Cpts323 GIT repository:

https://github.com/brianlamarche/Cpts323/tree/master/ProjectSDK

GUI Screenshots

You must create GUI storyboards for each screen of the software. Each storyboard should show the different states of the GUI if the software enters different modes, e.g. search and destroy versus idle. You can design your GUI to look however, you want. You just have to create storyboard screenshots beforehand. You must update your screenshots as appropriate for this project. If your screenshots do not include updates, you will lose points.

Supplied Data

Example XML and INI files are provided in the Git repository in a subdirectory of the lib folder.

https://github.com/brianlamarche/Cpts323/tree/master/lib

User Narratives

Write a user narrative for each user in your system.

Use Cases

Use cases are required to model each user and system interaction. You should use the following format:

ID:	**UC-6**
Title:	Fire Missiles
Description:	The AMSL should fire missiles when the fire button is pressed.
Primary Actor:	Judge
Preconditions:	Software has started an in idle mode
Postconditions:	Missile is fired
Main Success Scenario:	1. <fill in> 2. <fill in> 3. <fill in>
Extensions:	2a. <fill in>
Frequency of Use:	For testing purposes
Owner:	John Smith
Priority:	P3 – Medium
Risk	MIT23

Each Use Case should be associated to an owner (a task) and also related to a risk from your software management plan.

UML Modeling

You must model your software with UML Class Diagrams. All classes need to be included in these models. You must create UML component and sequence diagrams. You must include a UML use case diagram.

Tasks and Issue Tracking

You must enter each development task into GitHub as an issue per direction of your software management plan that you turned in as a group. Tasks must be closed before submission of your electronic version. Tasks must be printed and delivered as part of the hardcopy. Each task for this submission must be managed under a milestone that describes the assignment. **WARNING: Failure to use issue and task tracking will result in a zero on the assignment.**

Write-up

You must include a write-up with the following sections:

1. Description
2. GUI Storyboards
3. User Narratives
4. Formal Use Cases
5. UML Modeling Diagrams
    a. Class
    b. Use Case
    c. Component
    d. Sequence
6. Design patterns used
    a. Pros/Cons
7. Design considerations
8. Issues

Deliverables

The following must be turned in:

1. Hardcopy
    a. Write-up
    b. Code
2. Electronic Copy
    a. Write-up (PDF)
    b. Code (per assignment turn-in requirements)
        i. Project & Solution
    c. Modeling Project

Grading Breakdown

The grading breakdown will be as follows

1. Software Functions Per Specification – 40%
    a. If it doesn't build, then no credit is awarded here.
2. Modeling – 30%
3. Design Pattern Implementation – 15%
4. Use Cases - 5%
    a. Must include user narratives
    b. Updated use cases per functional requirement
    c. Updated storyboards
5. Design Consideration Description – 5%
    a. You must describe each of your design patterns
    b. You must describe why and how you used each one
    c. You must include any other considerations you gave
6. Bug Tracking – 5%