Level 100 Pikachus

Josh Jacobs

Jon Churchill

Chris Knight

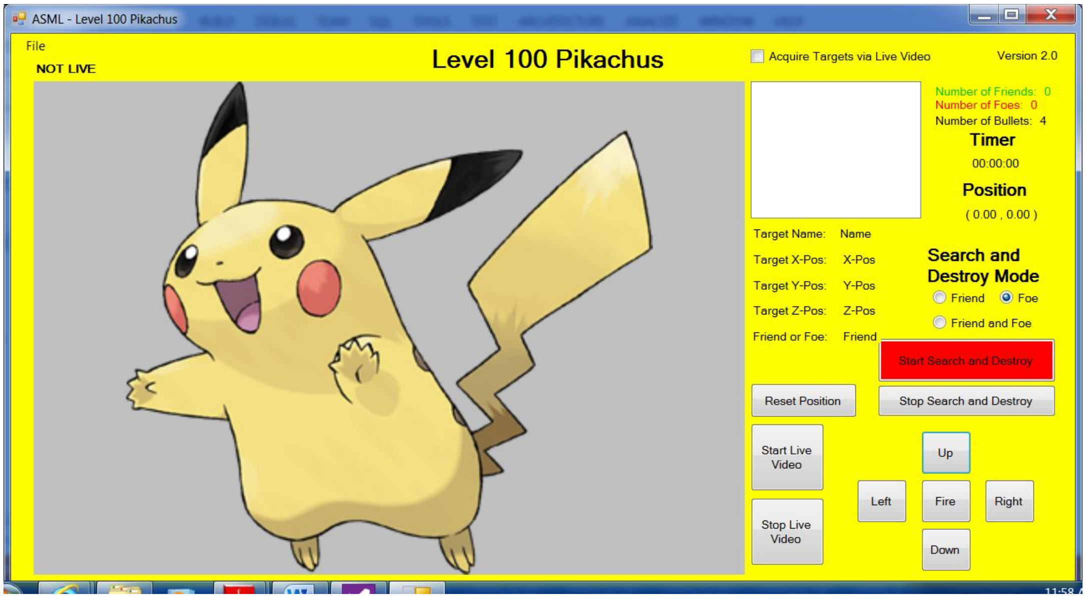
Table of Contents

Write Up

1) Project Description

This project is to create a graphical user interface (GUI) to control a dream cheeky Thunder missile launcher. The GUI will be able to load targets from an ".xml" or ".ini" file if the format is valid. These targets will then be loaded into the GUI and displayed in a list box. The GUI will have clickable buttons for directional movement of the missile launcher, and a button for firing missiles. When either the up, down, left, or right button is pressed, it can be held down to increase acceleration of movement. When the button is released, the movement will stop. When the fire button is pressed, a missile is fired from the launcher, and a new missile is put into the correct firing position, so when fire is pressed again the next missile is launched. When the user selects a new file from within the GUI, the previous target list will be removed, and a new target list will be loaded in. The GUI also contains buttons for search and destroy, and a box for video display. This project will implement the video display, allowing video to be sent to our GUI through a webcam. Search and destroy will be added to the project as well. It will be started by a button, and stopped by another one. There are radio buttons for selecting which set of targets to search and destroy: enemies, friends or enemies, and friends. The list of targets will be cycled through, telling the launcher to shoot at the targets determined by the radio buttons. A timer will also be kept when search and destroy is started, and the time will be stopped when search and destroy is stopped. The accuracy of the launcher at this stage is limited do to inconsistencies within the hardware, and the application is not multithreaded.

2) GUI storyboards



ASML - Level 100 Pikachus

NOT LIVE

Level 100 Pikachus

□ Acquire Targets via Live Video

Version 2.0

Number of Friends: 0
Number of Foes: 0
Number of Bullets: 4

Timer

00:00:00

Position

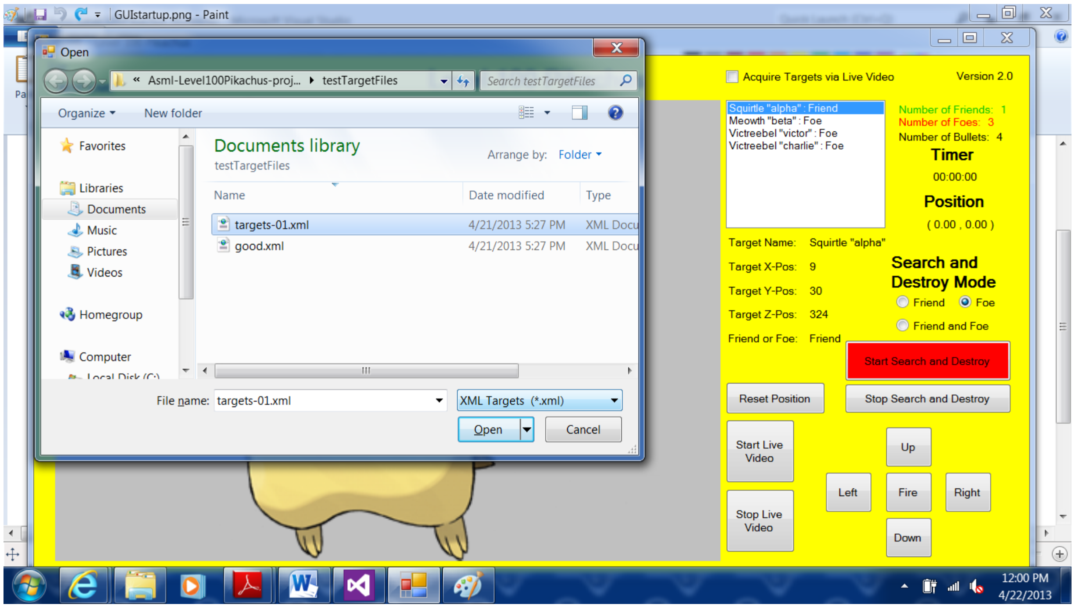
(0.00 , 0.00)

Target Name: Name
Target X-Pos: X-Pos
Target Y-Pos: Y-Pos
Target Z-Pos: Z-Pos
Friend or Foe: Friend

Search and Destroy Mode

○ Friend ● Foe
○ Friend and Foe

When the GUI is started up, this will be the first screen that the user will see.



GUIstartup.png - Paint

Open

Asml-Level100Pikachus-proj... ▸ testTargetFiles

Documents library
testTargetFiles

Name	Date modified	Type
targets-01.xml	4/21/2013 5:27 PM	XML Docu
good.xml	4/21/2013 5:27 PM	XML Docu

File name: targets-01.xml
XML Targets (*.xml)

☐ Acquire Targets via Live Video
Version 2.0

Squirtle "alpha" : Friend
Meowth "beta" : Foe
Victreebel "victor" : Foe
Victreebel "charlie" : Foe

Number of Friends: 1
Number of Foes: 3
Number of Bullets: 4

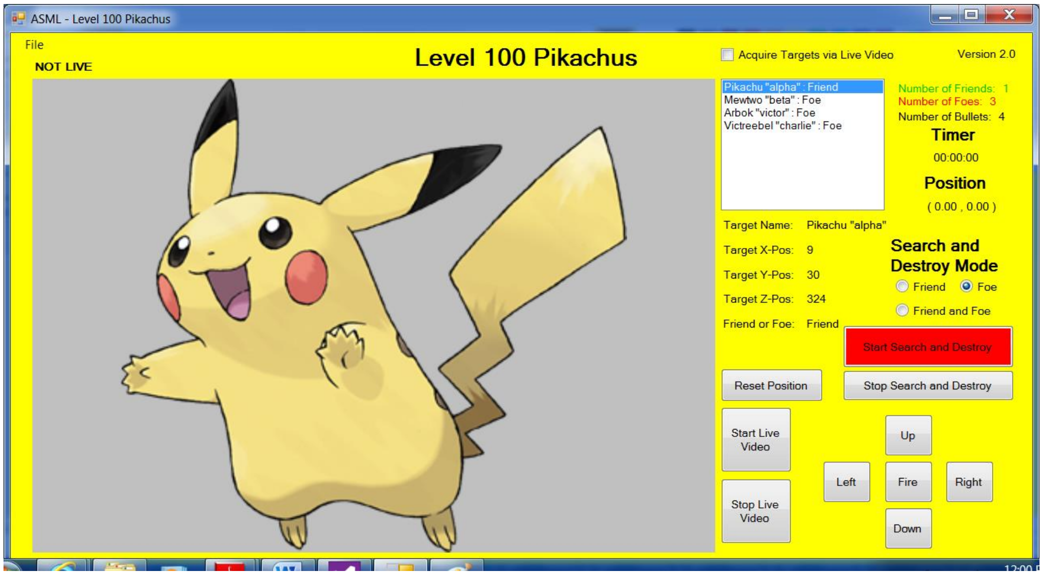
Timer
00:00:00

Position
(0.00 , 0.00)

Target Name: Squirtle "alpha"
Target X-Pos: 9
Target Y-Pos: 30
Target Z-Pos: 324
Friend or Foe: Friend

Search and Destroy Mode
○ Friend ● Foe
○ Friend and Foe

When choosing to open a file, the user will be presented with a pop up menu that will allow them to choose either xml or ini files.



ASML - Level 100 Pikachus

Level 100 Pikachus

NOT LIVE

☐ Acquire Targets via Live Video
Version 2.0

Pikachu "alpha" : Friend
Mewtwo "beta" : Foe
Arbok "victor" : Foe
Victreebel "charlie" : Foe

Number of Friends: 1
Number of Foes: 3
Number of Bullets: 4

Timer
00:00:00

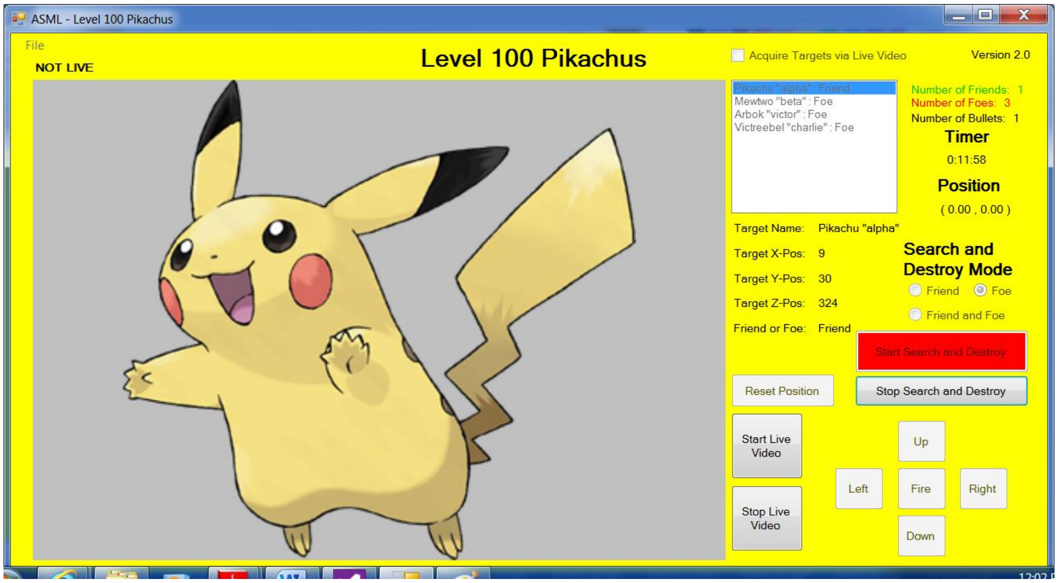
Position
(0.00 , 0.00)

Target Name: Pikachu "alpha"
Target X-Pos: 9
Target Y-Pos: 30
Target Z-Pos: 324
Friend or Foe: Friend

Search and Destroy Mode
○ Friend ● Foe
○ Friend and Foe

Once a file is loaded, the target list box will be updates, and the target information will be displayed just to the right of the list box. Also the Number of friends and foes will update.

When a fire or directional button is pressed, nothing on the GUI will pop up, but the launcher will follow those commands correctly.



ASML - Level 100 Pikachus

File

NOT LIVE

Level 100 Pikachus

□ Acquire Targets via Live Video

Version 2.0

Pikachu "alpha" : Friend
Mewtwo "beta" : Foe
Arbok "victor" : Foe
Victreebel "charlie" : Foe

Number of Friends: 1
Number of Foes: 3
Number of Bullets: 1

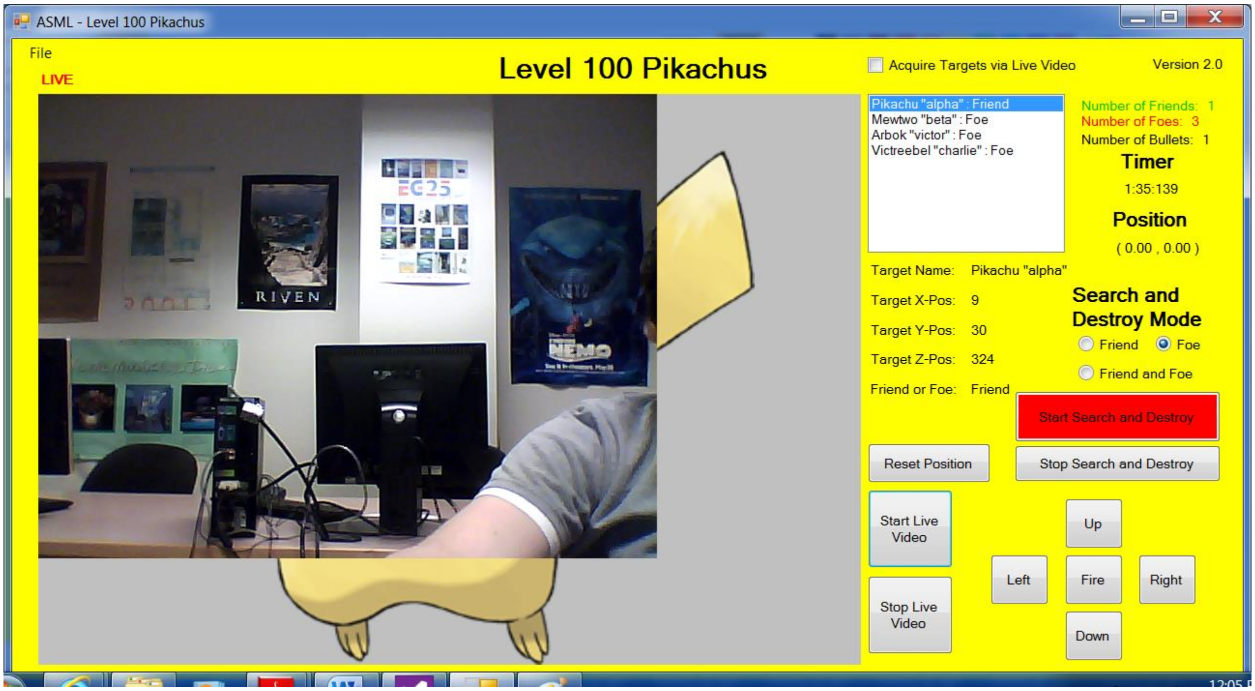
Timer
0:11:58

Position
(0.00 , 0.00)

Target Name: Pikachu "alpha"
Target X-Pos: 9
Target Y-Pos: 30
Target Z-Pos: 324
Friend or Foe: Friend

Search and Destroy Mode
○ Friend ● Foe
○ Friend and Foe

When search and destroy is started, the application will start the timer as seen above, and the launcher will cycle through the list of targets fire at the ones it is supposed to depending on the mode selected. The timer will stop when search and destroy is done.



ASML - Level 100 Pikachus

File

LIVE

Level 100 Pikachus

□ Acquire Targets via Live Video

Version 2.0

Pikachu "alpha" : Friend
Mewtwo "beta" : Foe
Arbok "victor" : Foe
Victreebel "charlie" : Foe

Number of Friends: 1
Number of Foes: 3
Number of Bullets: 1

Timer
1:35:139

Position
(0.00 , 0.00)

Target Name: Pikachu "alpha"
Target X-Pos: 9
Target Y-Pos: 30
Target Z-Pos: 324
Friend or Foe: Friend

Search and Destroy Mode
○ Friend ● Foe
○ Friend and Foe

When the start live video button was pressed, video is displayed in the box. Stopping live video will end the live feed, but keep the last captured picture on the screen.

3) User Narratives

Judge is a long time programming professor that wants to see some real world applications with some of this pet projects. With some code snippets he gave to all of his students he wants to see how well they can implement his rocket launching software to read targets from a file and automatically shoot at them.

Chris, Josh, and Jon are preparing to get the ASML ready for the competition. They need to make sure that the search and destroy mode for the launcher works. They want to be able to start search and destroy, start a timer, and allow the launcher to cycle through targets. This test is for functionality.

At the day of the launcher competition, the Judge wants to be able to choose which set of targets to shoot at, whether they are friends, enemies, or enemies and friends. He wants this chose to be easy to do and fast.

While testing the ASML aiming mechanisms, Chris, Josh, and Jon want to make sure that the launcher is firing accurately at the targets from the file. To do this they create a test field to set targets up on, and put targets on the location from the file.

The judge would like to see a live video feed of the area the launcher is looking at. He would like to be able to start the feed and stop it at any time. The video should be accurate, and non-choppy.

4) Formal Use Cases

ID:	UC-7
Title:	Search and destroy
Description:	When the search and destroy button is pressed on the GUI, the ASML should aim at the targets listed in the file, and fire at them.
Primary Actor:	Judge
Preconditions:	The GUI has been started and the search and destroy button has been selected.
Post conditions:	The ASML exits search and destroy mode after firing all its missiles
Main Success Scenario:	When the search and destroy button is pressed, the system walks through the list of targets, and fires at the targets location. A timer is started, and stopped when all the missiles have been fired. Search and destroy mode is then ended.
Extensions/Alt paths:	1: When the search and destroy button is pressed, the launcher does not go into a search and destroy mode 2: When the launcher fires at the first target, it stops and does not move on to the next target.

	3: When the launcher fires all of its missiles, it does not end search and destroy mode
Frequency of use:	Used during the competition.
Owner:	Josh Jacobs
Priority:	Infinite, which is our first priority
Risk:	Mit_04

ID:	UC-8
Title:	Cycling through targets in search and destroy
Description:	Once in search and destroy mode, the program needs to move through the list of targets, firing at each.
Primary Actor:	Judge
Preconditions:	The GUI has been loaded, and is in search and destroy mode
Post conditions:	Search and destroy has ended, with the last target still displayed.
Main Success Scenario:	When the application goes through the list of targets in search and destroy mode, it cycles through the targets, telling the launcher the correct ones to shoot at.
Extensions/Alt paths:	1: The program only locks on to one target, and doesn’t cycle through them 2: The program cycles through all the targets, even ones we are not supposed to shoot at. 3: The application while in search and destroy mode fails to recognize a target
Frequency of use:	Every time search and destroy mode is started.
Owner:	Josh Jacobs
Priority:	Infinite, highest priority
Risk:	Mit_04

ID:	UC-9
Title:	Live Video feed
Description:	On the GUI there should be a button to start and to stop live video feed.
Primary Actor:	Chris, Josh, Jon, judge
Preconditions:	The GUI has been loaded, and the user is able to turn on the video

	feed when he wants to.
Post conditions:	The button has been pressed, and video is being sent to the GUI.
Main Success Scenario:	When the user presses the start button, video feed streams to the GUI without being choppy. It can be stopped at any time by pressing the video feed stop button.
Extensions/Alt paths:	1: When the video feed button is pushed, no video is streamed to the GUI. 2: When the video feed button is pressed, only one image is sent, instead of a continuous stream of video. 3: When the video is streamed to the GUI, the feed is not smooth, but choppy.
Frequency of use:	Every time the GUI is ran, video should be available
Owner:	Josh Jacobs
Priority:	Infinite, which is our first priority
Risk:	Mit_04

ID:	UC-10
Title:	Setting which group of targets to shoot at
Description:	The user has a choice when search and destroy is selected to shoot at friends, enemies, or friends and enemies.
Primary Actor:	Josh, Judge
Preconditions:	The GUI has been loaded, and is ready to enter search and destroy mode
Post conditions:	The correct group of targets was shot at.
Main Success Scenario:	Depending on which group of targets to shoot, that the user selected, search and destroy only fires at those targets.
Extensions/Alt paths:	1: All targets are shot at, even when search and destroy is told not to. 2: No targets are shot at; the system doesn’t acquire any targets. 3: The launcher shoots at targets that it is not supposed to.
Frequency of use:	Often, any time search and destroy mode is entered, and during the competition.
Owner:	Josh Jacobs
Priority:	infinite, it’s our highest priority.

Risk:	Mit_04

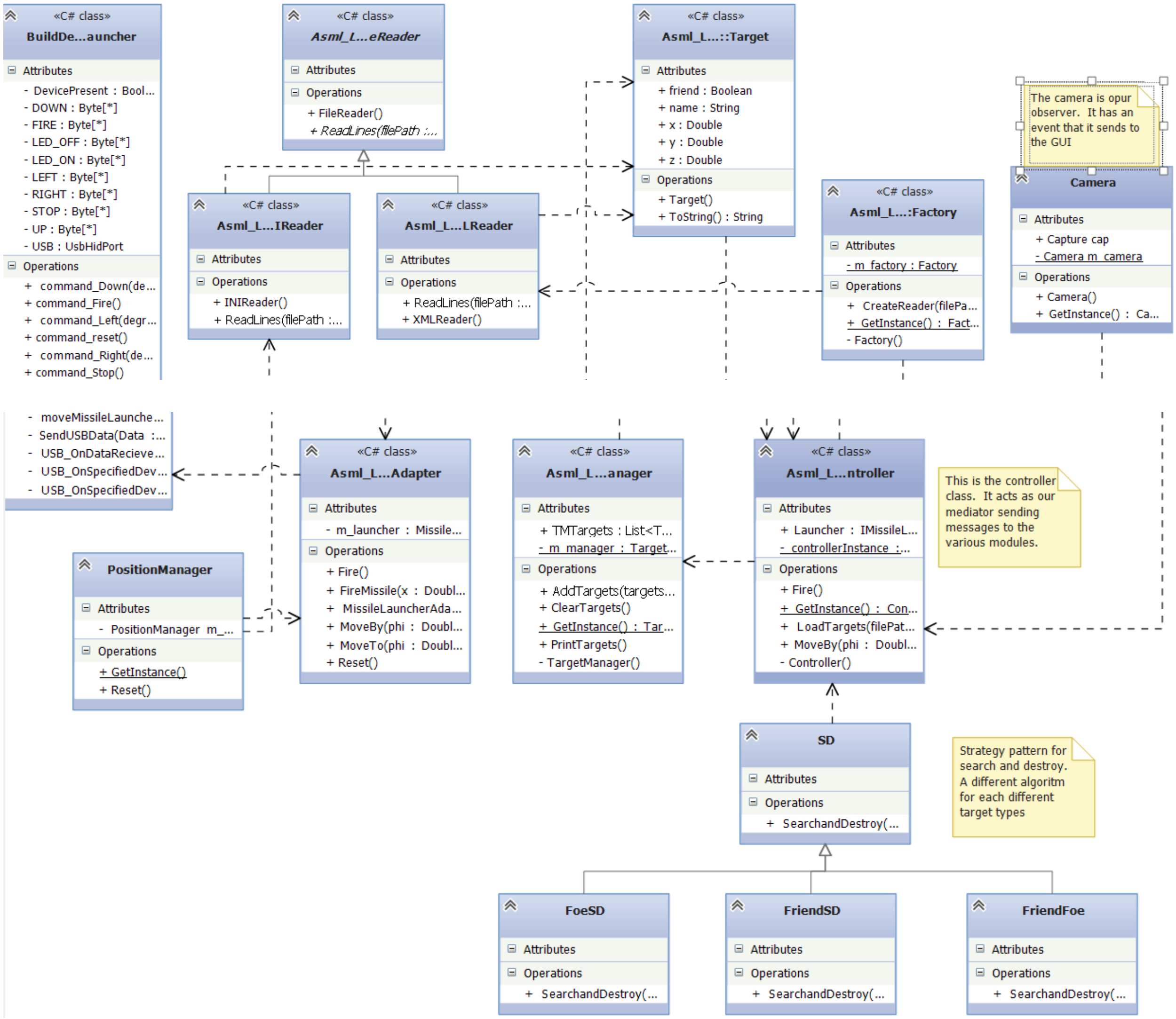
ID:	UC-11
Title:	Aim of launcher for move to.
Description:	The GUI is running, and search and destroy is moving to fire on a target.
Primary Actor:	Chris, Josh, Jon
Preconditions:	The GUI has been loaded, and is in search and destroy mode
Post conditions:	The ASML is in idle mode, and the targets have been fired at.
Main Success Scenario:	The launcher moves to the correct coordinates to fire at the target, and hits the target.
Extensions/Alt paths:	1: The launcher doesn't move anywhere when told to 2: The launcher moves, but is inaccurate and misses the target.
Frequency of use:	Every time search and destroy is ran
Owner:	Jon Churchill, Christopher Knight, Josh Jacobs
Priority:	4, so it's our highest priority
Risk:	Mit_04

ID:	UC-12
Title:	The timer for search and destroy
Description:	The GUI is running, and search and destroy is entered, displaying the time it has spent in search and destroy.
Primary Actor:	Judge
Preconditions:	The GUI has been loaded, and search and destroy is started
Post conditions:	The GUI displays on the timer, the amount of time the system spent in search and destroy mode.
Main Success Scenario:	When search and destroy mode is started, the timer starts, and when it ends, the timer stops, keeping the time where it was stopped displayed on the GUI.
Extensions/Alt paths:	1: Search and destroy is started, but the timer doesn't start 2: Search and destroy ends, but the timer does not stop 3: The timer stops correctly, but doesn't keep the time displayed on the GUI.
Frequency of use:	Every time Search and destroy is started.
Owner:	Josh Jacobs
Priority:	1, this is only the third highest because there will be a backup timer

	system during the competition.
Risk:	Mit_03

5) UML Diagrams

Class Diagram

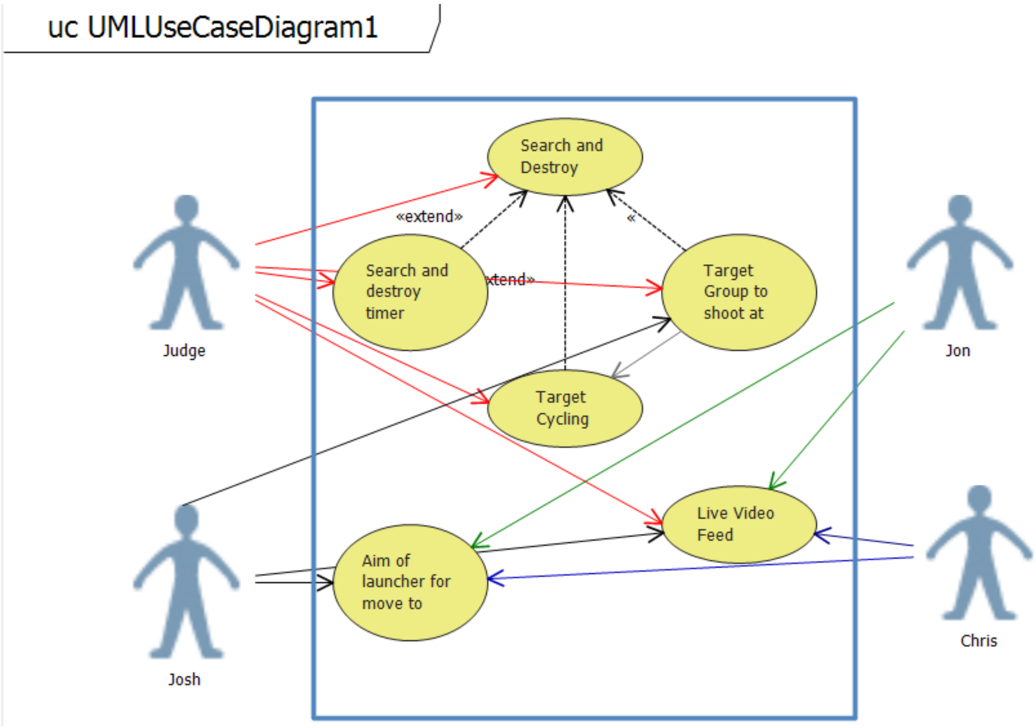


The camera is opur observer. It has an event that it sends to the GUI

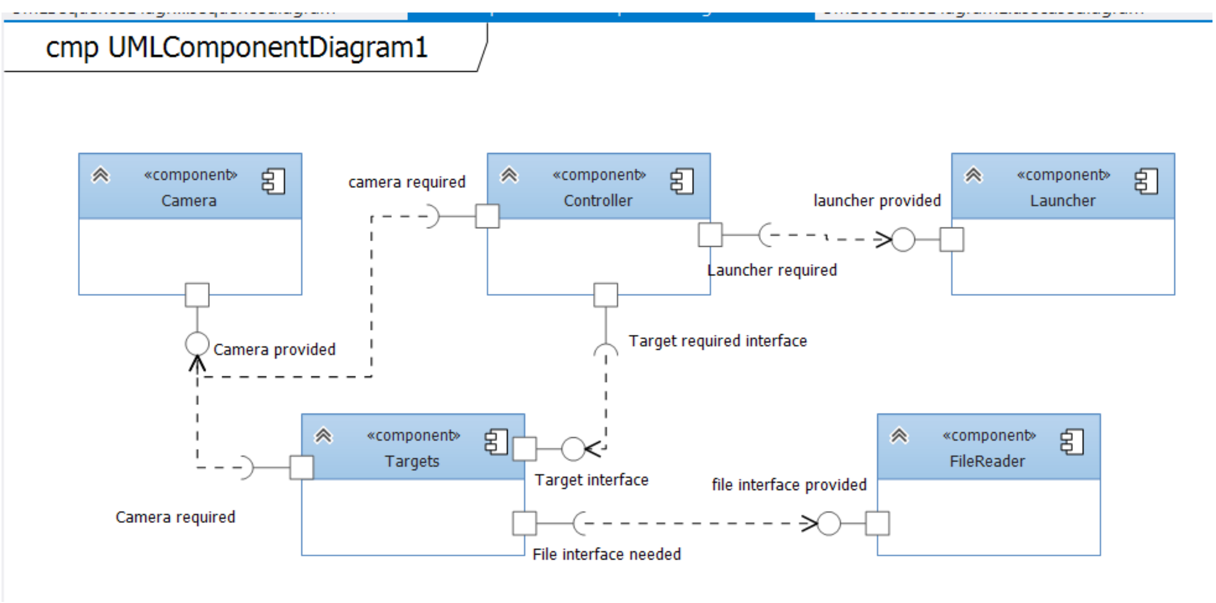
This is the controller class. It acts as our mediator sending messages to the various modules.

Strategy pattern for search and destroy. A different algoritm for each different target types

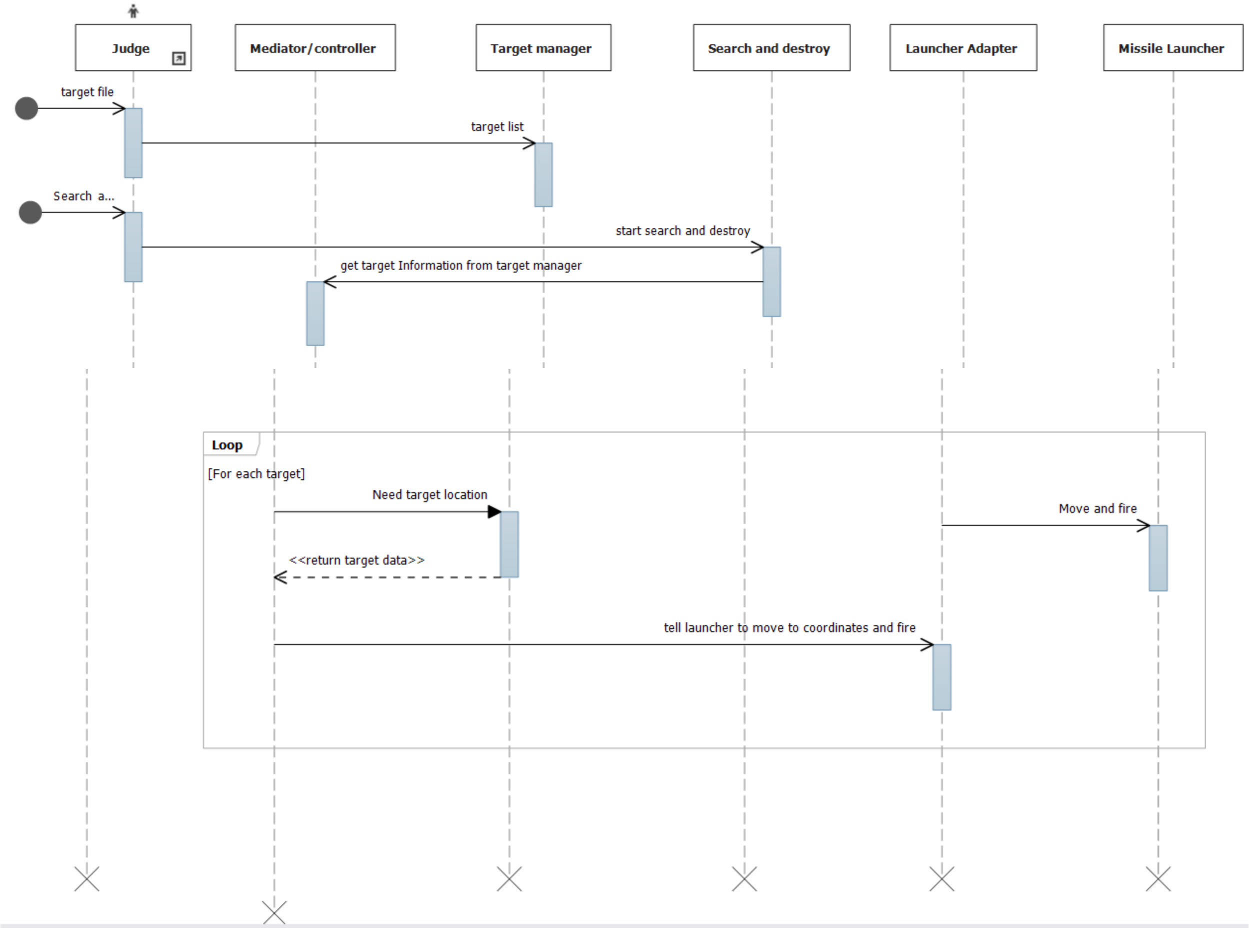
Use Case Diagram



Component Diagram



Sequence Diagram



6) Design Patterns used

The design patterns that we used were adapter, singleton, factory, mediator, observer, and strategy. A Benefit of using the factory design pattern is that the file readers are not coupled. If later in the project, the requirements change, and we need to allow for another type of target file, the factory pattern allows for the easy insertion of another type of file reader. A good thing about using a singleton is when targets are read in from a file; only one set of targets can be active. When another file is read from, the original targets must be deleted to allow for a new set of targets. This ensures that we are not getting extra targets. The adapter pattern is very good for using the original launcher code, but not having to access it directly. The program that we use to fire, does not directly fire using the provided code. The adapter translates our fire to the original fire. This is good in case we get a new missile launcher class. We would just need to change the adapter to be able to talk to the new launcher file. A downside to using the singleton pattern for a target manager is if we wanted to read multiple files and keep appending targets onto the list of targets. This is not allowed because only one set of targets can be loaded at a time with a singleton. A singleton launcher control is a good thing though so only one module can control the launcher. We do not want multiple modules trying to control the launcher at the same time. The mediator pattern we are using to control the logic of the application. It will know which information to send to the GUI, and when to update it. It will also know which modules to use when buttons on the GUI are clicked. This is useful because it will control which module needs to do its work, without being coupled to all the other modules. The observer pattern will be our target manager and will update our targets list. It will see when a new target is selected, and tell the GUI to update the list box, and tell the launcher where to shoot at. We will use the strategy pattern for which targets to shoot at, whether it is friends and foes, just foes, or just friends. This pattern is useful because most of the steps needed for which targets to shoot as are the same, but each setting will change the last final steps when shooting at the correct target. It will also allow the same section of code to work until one of the three different firing types needs to alter it.

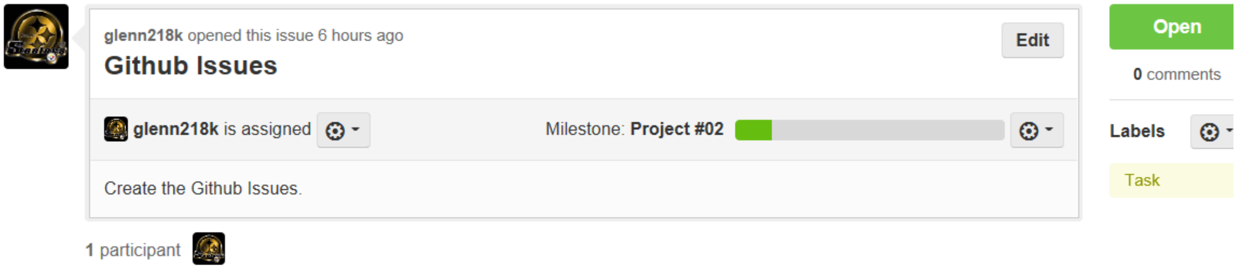
7) Design Considerations

The reason that we built the software the way we did was to keep the program from being tightly coupled, and adaptable to projects that are coming down the road. Using the adapter design pattern allows us to interact with the launcher without having to change any of the launchers code. When we say fire, the adapter knows how to talk to us, and then translate that to the missile launcher Google code without any problems. By doing this, if we got a new missile launcher source code file, we would only have to alter the adapter to be able to talk to it. The singleton design pattern makes sure that we only have one set of targets at a time. A consideration we had when using this pattern was when we were in the competition. We would be able to load a new set of target files for each event, clearing out the data from the last events file. The pattern also makes sure that there is only one missile launcher controller, so different commands can't be sent to the launcher at the same time. The mediator pattern was considered because we needed some way of controlling the logic throughout the program but still wanted to keep the modules decoupled. It allows for updates to and from the GUI, targets, and launcher. The observer pattern is a good consideration for our target list. It will view the event of a target change, and can notify the launcher, GUI, and camera through the mediator. This will be an easy way of moving to the next target in the list and updating our application. The strategy pattern fits well with how we have our radio buttons for which set of targets to shoot at. Most of the work for these three buttons is the same and only need to branch off to shoot at the specific set whether it is enemies, friends or enemies, or just friends. Each of these three requires the same steps leading up to them, but only slightly change what happens. This makes the consideration of a strategy very useful.

8) Issues

https://github.com/glenn218k/ASML/issues

(project 2 issues home)


glenn218k opened this issue 6 hours ago

Github Issues

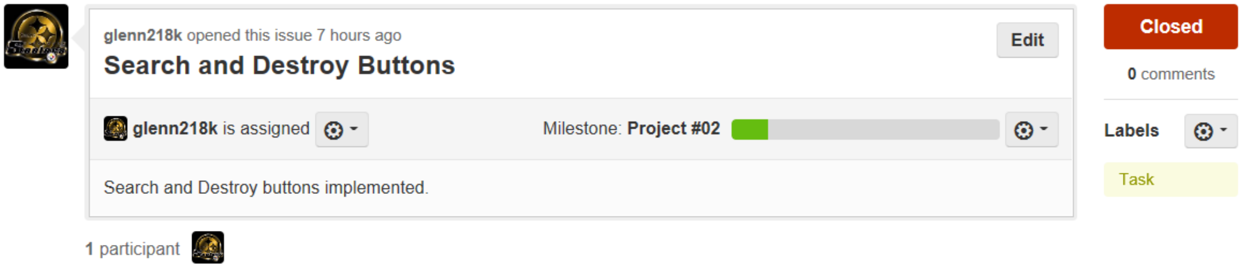
glenn218k is assigned

Milestone: Project #02

Create the Github Issues.

1 participant

https://github.com/glenn218k/ASML/issues/24



glenn218k opened this issue 7 hours ago

Search and Destroy Buttons

glenn218k is assigned

Milestone: Project #02

Search and Destroy buttons implemented.

1 participant

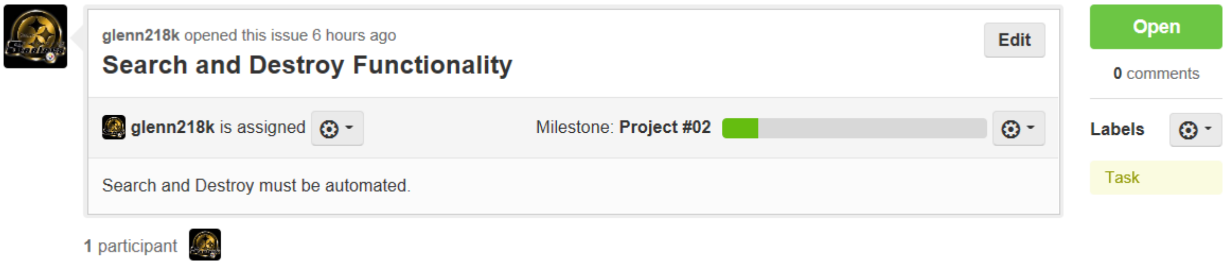
Closed

0 comments

Labels

Task

https://github.com/glenn218k/ASML/issues/25



glenn218k opened this issue 6 hours ago

Search and Destroy Functionality

glenn218k is assigned

Milestone: Project #02

Search and Destroy must be automated.

1 participant

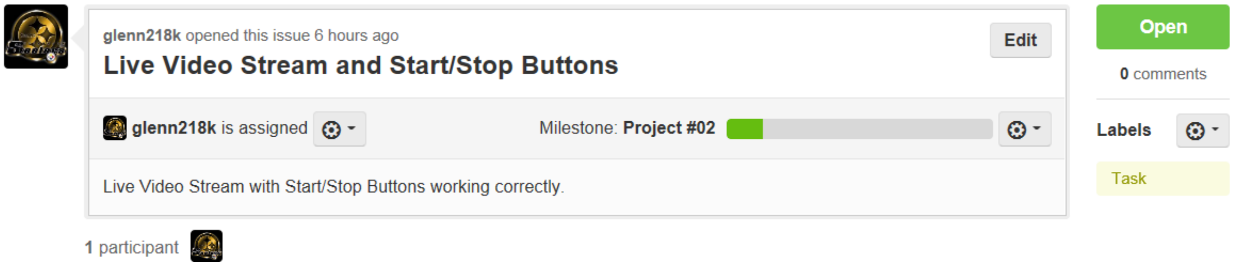
Open

0 comments

Labels

Task

https://github.com/glenn218k/ASML/issues/26



glenn218k opened this issue 6 hours ago

Live Video Stream and Start/Stop Buttons

glenn218k is assigned

Milestone: Project #02

Live Video Stream with Start/Stop Buttons working correctly.

1 participant

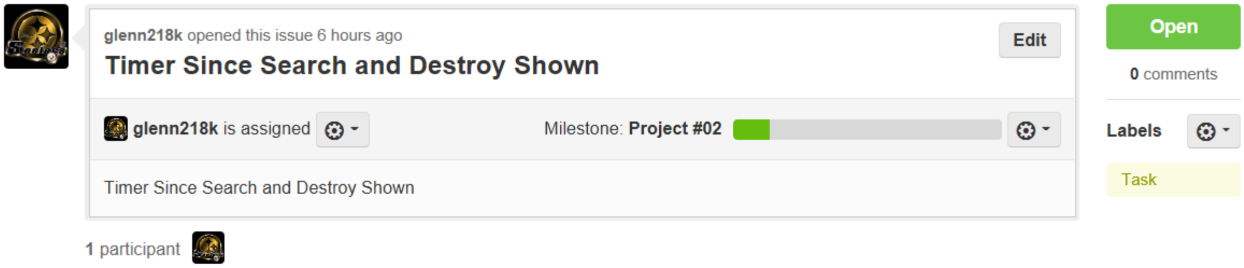
Open

0 comments

Labels

Task

https://github.com/glenn218k/ASML/issues/27



glenn218k opened this issue 6 hours ago

Timer Since Search and Destroy Shown

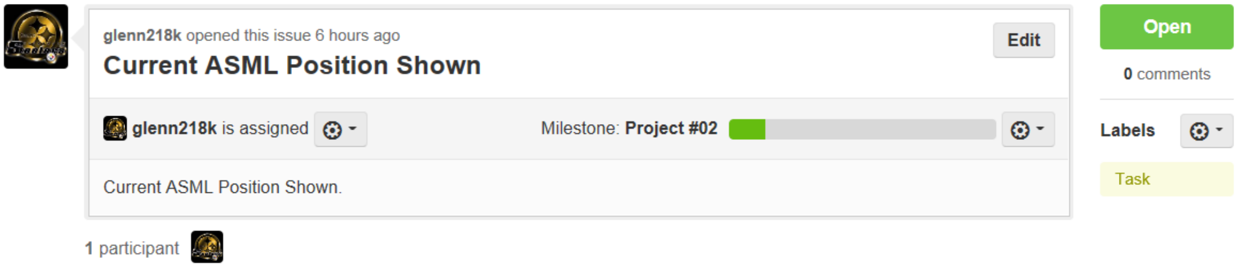
glenn218k is assigned

Milestone: **Project #02**

Timer Since Search and Destroy Shown

1 participant

https://github.com/glenn218k/ASML/issues/28



glenn218k opened this issue 6 hours ago

Current ASML Position Shown

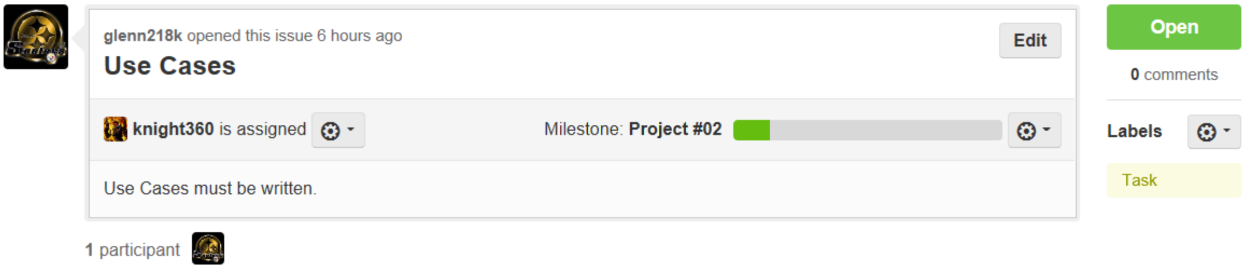
glenn218k is assigned

Milestone: **Project #02**

Current ASML Position Shown.

1 participant

https://github.com/glenn218k/ASML/issues/29



glenn218k opened this issue 6 hours ago

Use Cases

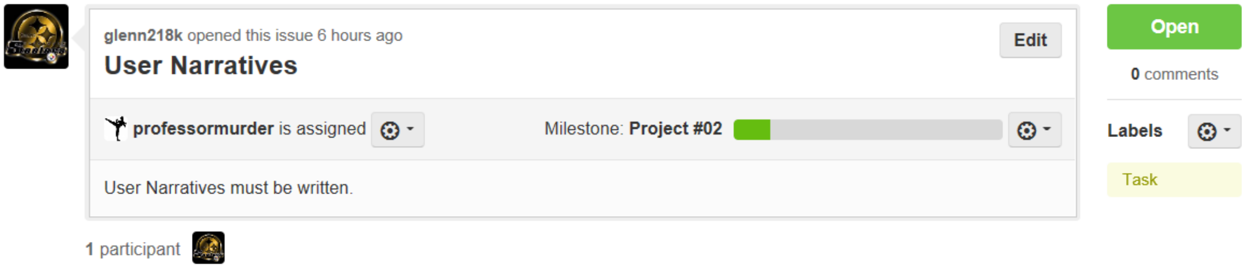
knight360 is assigned

Milestone: **Project #02**

Use Cases must be written.

1 participant

https://github.com/glenn218k/ASML/issues/30



glenn218k opened this issue 6 hours ago

User Narratives

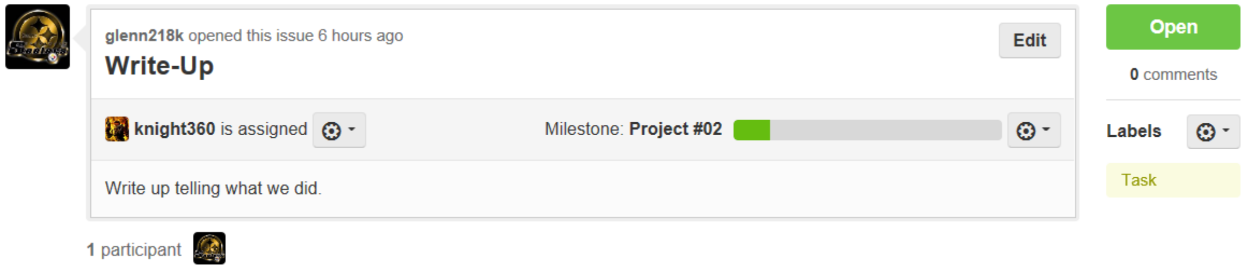
professormurder is assigned

Milestone: **Project #02**

User Narratives must be written.

1 participant

https://github.com/glenn218k/ASML/issues/31



glenn218k opened this issue 6 hours ago

Write-Up

knight360 is assigned

Milestone: Project #02

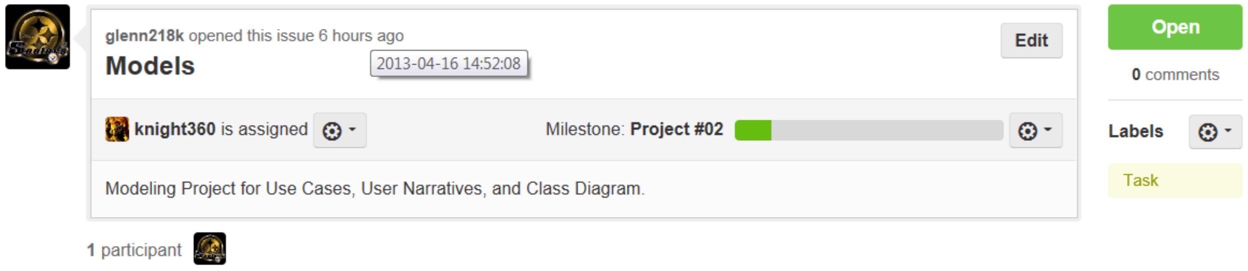
Write up telling what we did.

0 comments

Task

1 participant

https://github.com/glenn218k/ASML/issues/32



glenn218k opened this issue 6 hours ago

Models

2013-04-16 14:52:08

knight360 is assigned

Milestone: Project #02

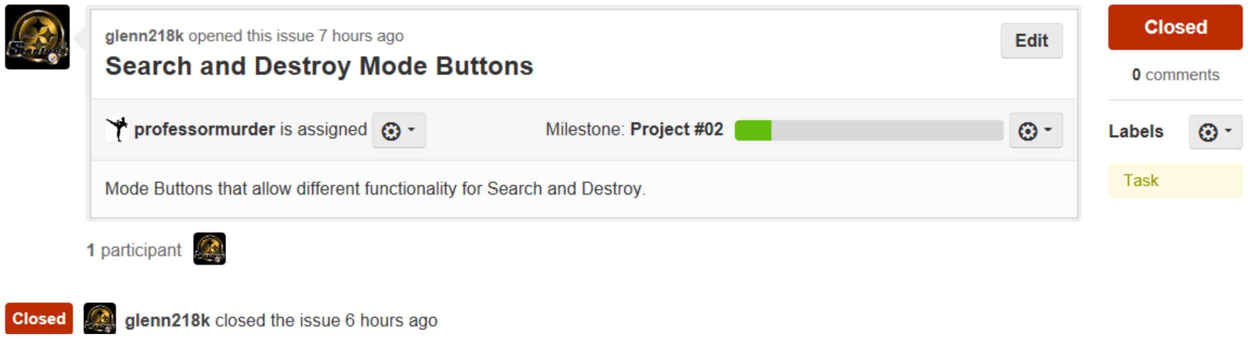
Modeling Project for Use Cases, User Narratives, and Class Diagram.

0 comments

Task

1 participant

https://github.com/glenn218k/ASML/issues/33



glenn218k opened this issue 7 hours ago

Search and Destroy Mode Buttons

professormurder is assigned

Milestone: Project #02

Mode Buttons that allow different functionality for Search and Destroy.

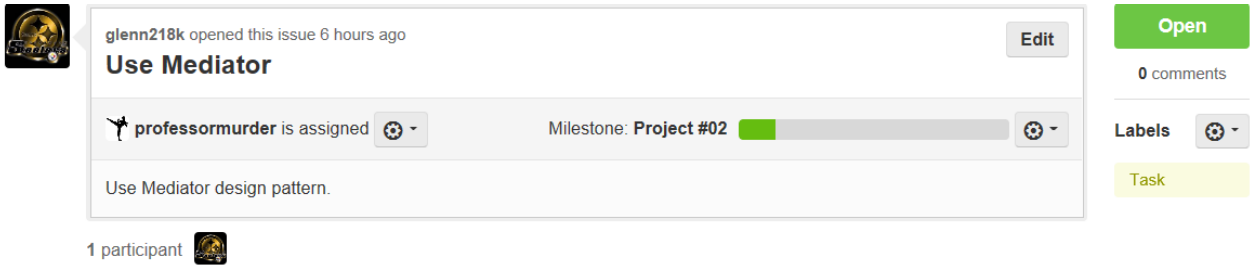
0 comments

Task

1 participant

Closed glenn218k closed the issue 6 hours ago

https://github.com/glenn218k/ASML/issues/34



glenn218k opened this issue 6 hours ago

Use Mediator

professormurder is assigned

Milestone: Project #02

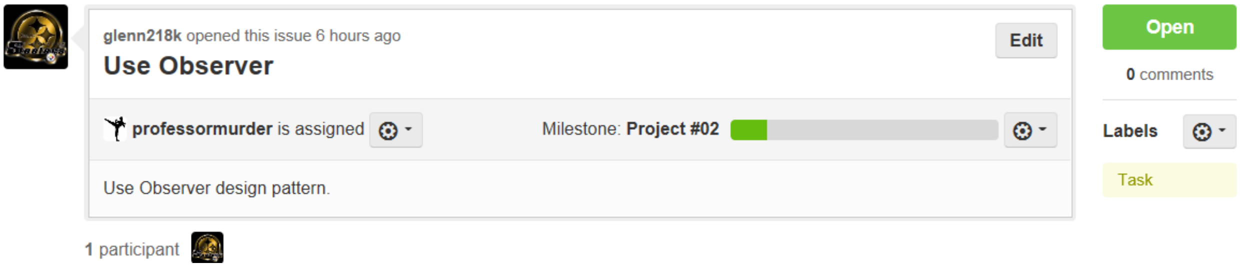
Use Mediator design pattern.

0 comments

Task

1 participant

https://github.com/glenn218k/ASML/issues/35



glenn218k opened this issue 6 hours ago

Use Observer

professormurder is assigned

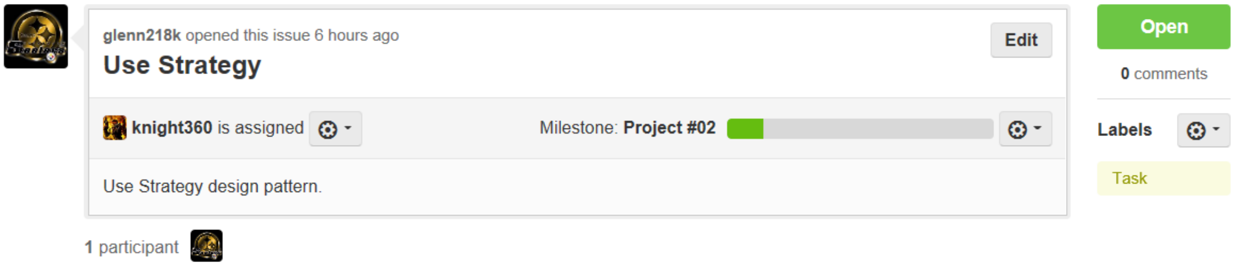
Milestone: Project #02

Use Observer design pattern.

1 participant

0 comments

https://github.com/glenn218k/ASML/issues/36



glenn218k opened this issue 6 hours ago

Use Strategy

knight360 is assigned

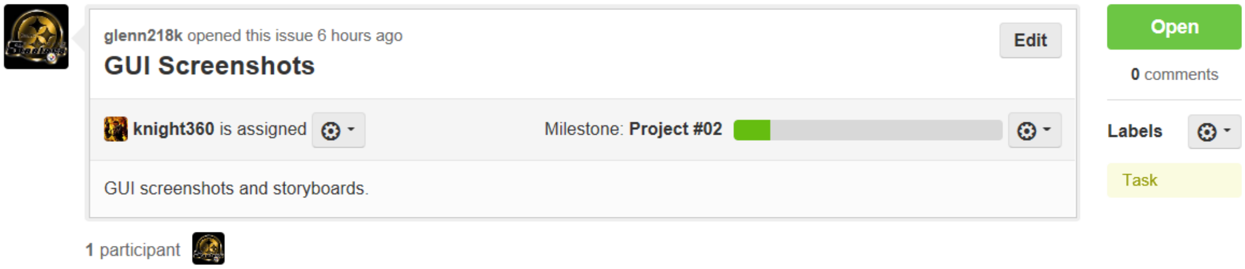
Milestone: Project #02

Use Strategy design pattern.

1 participant

0 comments

https://github.com/glenn218k/ASML/issues/37



glenn218k opened this issue 6 hours ago

GUI Screenshots

knight360 is assigned

Milestone: Project #02

GUI screenshots and storyboards.

1 participant

0 comments

https://github.com/glenn218k/ASML/issues/38