



UNIVERSITEIT VAN PRETORIA  
UNIVERSITY OF PRETORIA  
YUNIBESITHI YA PRETORIA

NEO TANDEM TECHNOLOGIES



---

## Eye Tracking

---

*Author:*

Duran Cole  
Michael Nunes  
Molefe Molefe  
Tebogo Seshibe  
Timothy Snayers

*Student number:*

u13329414  
u12104592  
u12260429  
u13181442  
u13397134

July 27, 2015

# ARCHITECTURE REQUIREMENTS

## EYE TRACKING

Github Link: [Neo Tandem Technology](#)

Version: 0.2 Alpha July 27, 2015

# Contents

1	Introduction . . . . .	5
1.1	Project Background . . . . .	5
1.2	Project Vision . . . . .	5
2	Use Case/Service Contracts . . . . .	6
2.1	Heat Maps . . . . .	6
2.2	Save Heat Maps . . . . .	8
2.3	Gaze Point Maps . . . . .	11
2.4	Save Gaze Point Maps . . . . .	13
2.5	Models . . . . .	16
2.6	Raw Information . . . . .	22
2.7	Statistics . . . . .	24
2.8	Record Eyes . . . . .	25
3	Required Functionality . . . . .	27
3.1	Render 3D scene . . . . .	27
3.2	Eye-tracking on a 3D scene . . . . .	27
3.3	Create heat mapped texture . . . . .	27
3.4	Map heat mapped texture back onto scene . . . . .	27
3.5	Real-time heat mapping . . . . .	27
3.6	Interactive Maps . . . . .	28
3.7	In-Video Eye Tracking . . . . .	28
3.8	Post-video Eye Tracking . . . . .	28
3.9	Suitable Video Output Formats . . . . .	28
4	Domain Model . . . . .	30

# List of Figures

1	Heat Map Use Case . . . . .	6
2	Create Heat Map Activity . . . . .	7
3	Read Heat Map Activity . . . . .	7
4	Save Heat Map Activity . . . . .	8
5	Save Heat Map Use Case . . . . .	9
6	Save on 2D Model Activity . . . . .	9
7	Save on Video Model Activity . . . . .	10
8	Save on 3D Model Activity . . . . .	11
9	Gaze Point Use Case . . . . .	11
10	Create Gaze Point Map Activity . . . . .	12
11	Read Gaze Point Map Activity . . . . .	13
12	Save Gaze Point Map Activity . . . . .	13
13	Save Gaze Point Map Use Case . . . . .	14
14	Save on 2D Model Activity . . . . .	14
15	Save on Video Model Activity . . . . .	15
16	Save on 3D Model Activity . . . . .	16
17	Models Use Case . . . . .	16
18	3D Models Use Case . . . . .	17
19	Import 3D Model Activity . . . . .	17
20	View 3D Model Activity . . . . .	18
21	Create 3D Model Images Activity . . . . .	18
22	2D Models Use Case . . . . .	19
23	Import 2D Model Activity . . . . .	19
24	View 2D Model Activity . . . . .	19
25	Export 2D Model Activity . . . . .	20
26	Video Models Use Case . . . . .	20
27	Import Video Model Activity . . . . .	21

28	View Video Model Activity . . . . .	21
29	Export 3D Model Activity . . . . .	22
30	Raw Information Use Case . . . . .	22
31	Save Raw Information Activity . . . . .	23
32	Read Raw Information Activity . . . . .	23
33	Statistics Use Case . . . . .	24
34	Create Statistics Activity . . . . .	24
35	Save Statistics Activity . . . . .	25
36	Record Eye Use Case . . . . .	25
37	Record Eye Activity . . . . .	26
38	Domain Model . . . . .	30

# **1 Introduction**

## **1.1 Project Background**

The Open Gaze And Mouse Analyzer is a company involved with the development of eye tracking technology. This technology allows users to interact with a device compatible with the OGAMA software. This also provides another means of communication by providing hands-free communication with the device the user is using. OGAMA uses an eye-tracking device designed and created by The Eye Tribe. This device connects to a user's computer through a USB 3.0 connection. The device tracks the users eye movements through a series of sensors within the front of the device.

## **1.2 Project Vision**

The aim of this project is to provide an extension to the already existing functionality of the OGAMA software. The extension should be able to track the focus of a user on 3D models. It should be able to generate a heat-map of the users focus that can be applied to the 3D model. An extra functionality should be that the extension can track the focus on a video. Heat-maps should also be possible to render to the video.

## 2 Use Case/Service Contracts

### 2.1 Heat Maps

This section will cover all the aspects of the heat maps use case (Figure 1). This use case is chosen as it is a vital part of the entire system. The data that is recorded will be used to create the heat maps that can then later be used to analyse the data and be used for future uses.

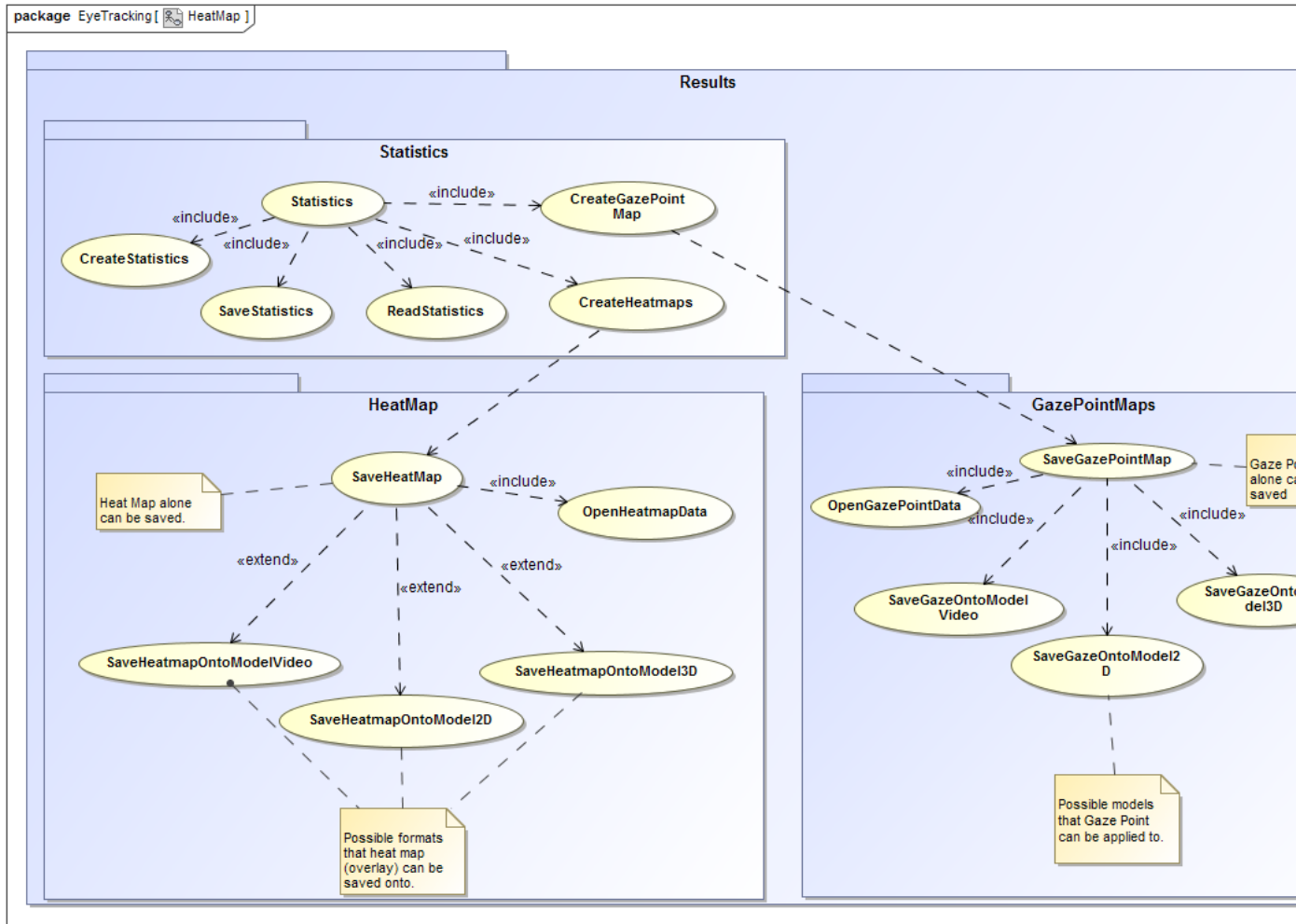


Figure 1: Heat Map Use Case

The following sub use cases are included in this use case:

#### 2.1.1 CreateHeatMap

The create heat map use case deals with the creation of the heat map from the recorded data. The heat map shows where on the media item has been viewed the most by the

user(s).

- Pre-condition: Eye tracking data must be already recorded.
- Post-condition: Heat-map of the data is created.
- Request Data Structure: Raw information recorded by eye tracking camera.
- Return data Structure: Heat-map is created

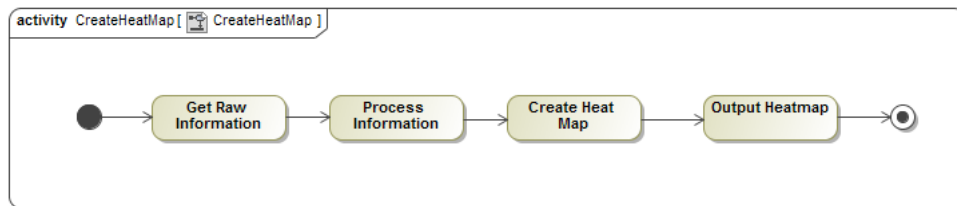


Figure 2: Create Heat Map Activity

### 2.1.2 ReadHeatMap

This use case deals with the viewing of heat-maps from their recorded data. Once the heat-map is created the users will be able to see the heat-map and view the data.

- Pre-condition: Heat-map must have been created.
- Post-condition: Heat-map of the data is viewable and can be further analysed.
- Request Data Structure: The created heat-map.
- Return Data Structure: Heat-map is viewable to the user(s).

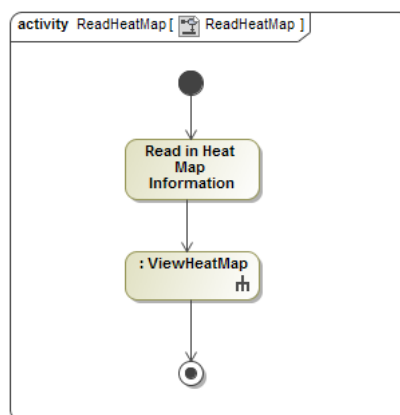


Figure 3: Read Heat Map Activity



### 2.1.3 SaveHeatMap

This use case deals with the saving of heat-maps that are created from their respective recorded data. The heat-map will be saved on the users computer to be viewed at a later date. This will save only the heat-map which can later be saved onto a model at a later date.

- Pre-condition: Raw information must have already been recorded so that new heat-map can be created.
- Post-condition: Heat-map of the data is saved.
- Request Data Structure: Raw eye tracking information
- Return Data Structure: Heat-map is saved.

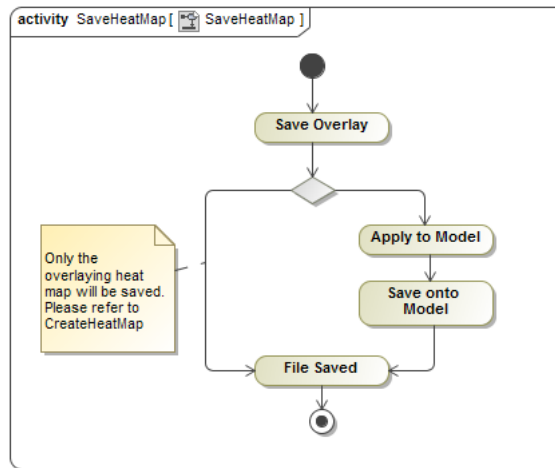


Figure 4: Save Heat Map Activity

## 2.2 Save Heat Maps

The saving of the heat-maps provides important functionality to the system as it will allow the users to save the data which can later be used to either continue research or to compare data. The heat-map will serve as an overlay for the media type and be placed over the media model. The saving of heat maps can be divided into three sub-sections, namely SaveHeatMapOntoModel3D, SaveHeatMapOntoModelVideo and SaveHeatMapOntoModel2D.

### 2.2.1 SaveHeatMapOntoModel2D

The heat-map that is created for a 2D model media type will be saved and applied to the 2D model to see the heat-map on top of the model to indicate what has been viewed the most by the user(s) test participants.

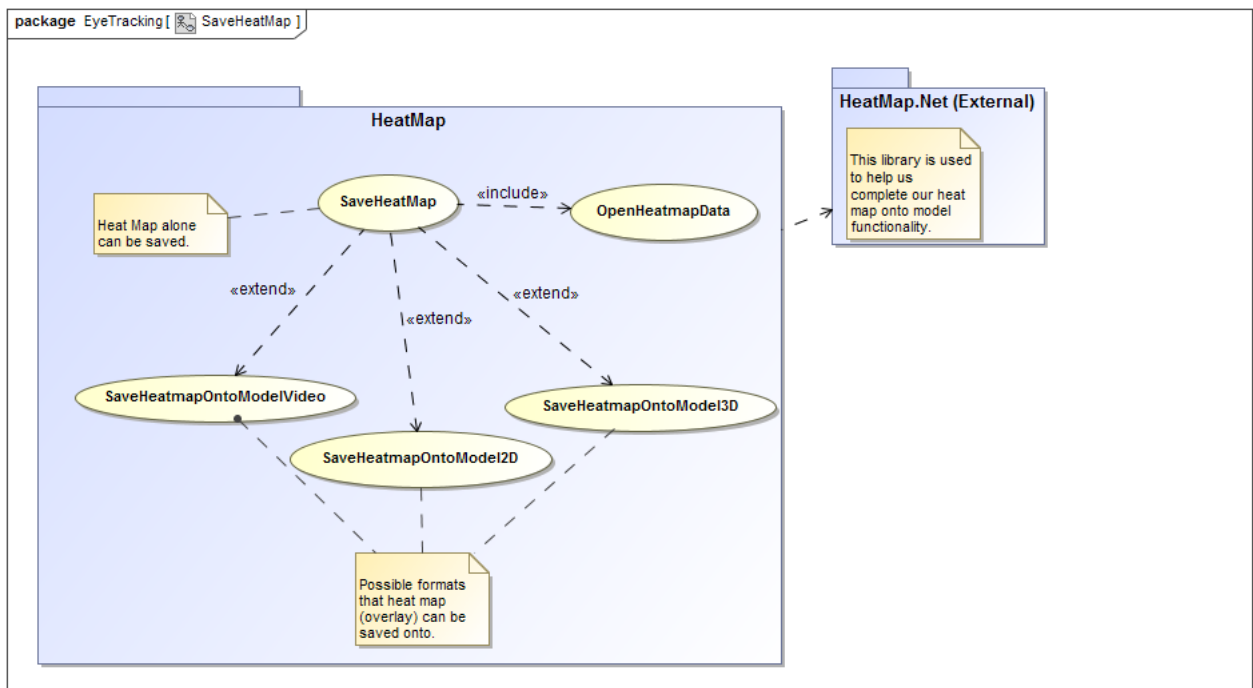


Figure 5: Save Heat Map Use Case

- Pre-condition: Eye tracking data must have already been recorded.
- Post-condition: Heat map of the data is viewable on the 2D model as an overlay.
- Request Data Structure: Raw eye tracking data.
- Return Data Structure: Heat map of the data is placed over the 2D Model.

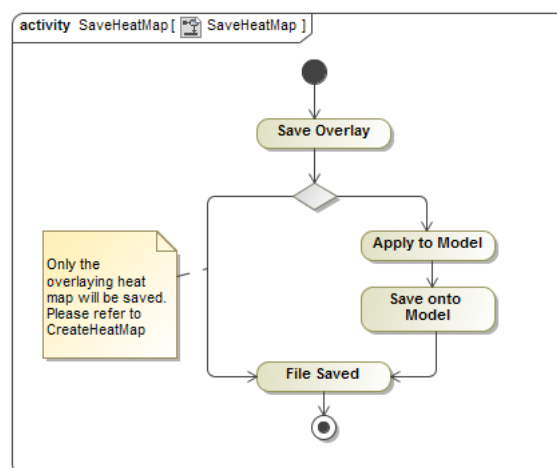


Figure 6: Save on 2D Model Activity

### 2.2.2 SaveHeatMapOntoModelVideo

The heat-map that is created for a video media type will be saved and then applied to the video to see the heat-map over the video to indicate what has been viewed the most by the users. The video model is spliced into images at either 30 or 60 frames per second of the video, the heat-maps can then be applied to the created images and recompiled into a new video.

- Pre-condition: Eye tracking data must have been recorded and video model must be available.
- Post-condition: New video is created with the heat-map on top of it.
- Request Data Structure: Video and raw information of recording.
- Return Data Structure: Heat-map of the data is placed over each frame of the video.

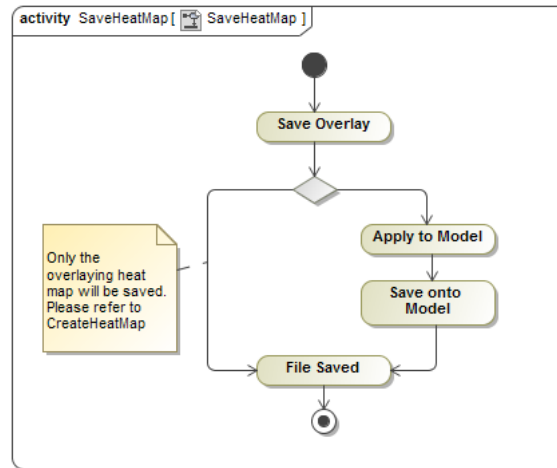


Figure 7: Save on Video Model Activity

### 2.2.3 SaveHeatMapOntoModel3D

The 3D model will have multiple images of it created from different angles Each created image can then have a heat-map placed over it.

- Pre-condition: Eye tracking data must have already been recorded and 3D model must be available for the creation of images.
- Post-condition: Heat-map of the data is viewable on the 3D model as an overlay.
- Request Data Structure: Recorded eye tracking information.
- Return Data Structure: Heat-map of the data is placed over each image of the 3D model.

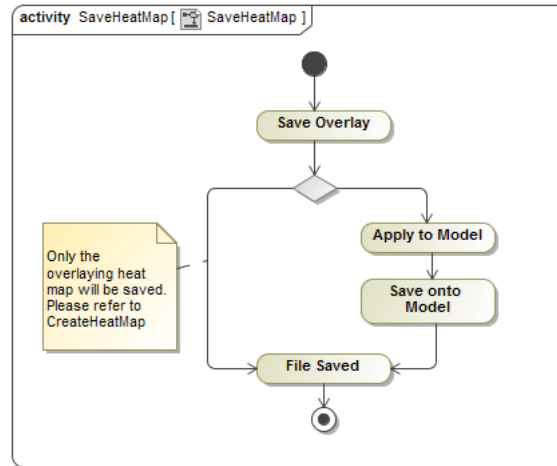


Figure 8: Save on 3D Model Activity

## 2.3 Gaze Point Maps

This section will cover all the aspects of the Gaze Point Map use case (Figure 9). Gaze point maps track where the user is looking but specifies the order in which their respective gaze moves from one location to another. This use case forms an added extra to the user and can be used to further analyse information and data that has been recorded. The data that is recorded will be used to create the gaze point that can then later be used to analyse the data and be used for future uses.

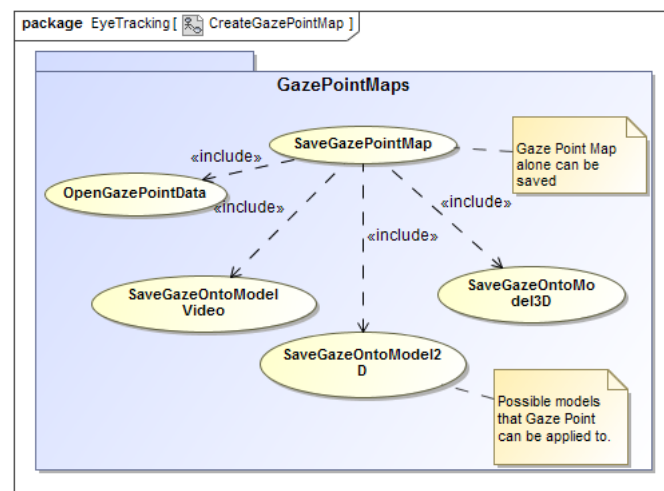


Figure 9: Gaze Point Use Case

The following sub use cases are included in this use case:

### 2.3.1 CreateGazePointMap

The create gaze point activity deals with the creation of the gaze point maps from the recorded data. The gaze point map shows the order and length in which the subject views the model.

- Pre-condition: Eye tracking data must be already recorded.
- Post-condition: Gaze point map of the data is created.
- Request Data Structure: Raw information recorded by eye tracking camera.
- Return data Structure: Gaze point map is created

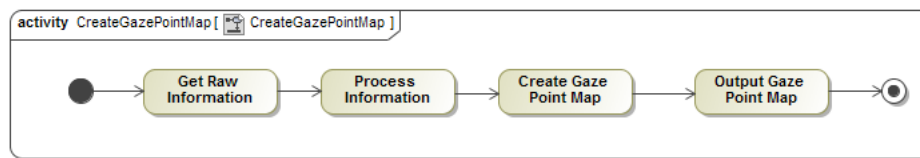


Figure 10: Create Gaze Point Map Activity

### 2.3.2 ReadGazePointMap

This activity deals with the viewing of gaze point maps from their recorded data. Once the gaze point maps have been created the users will be able to see the gaze point maps and view the data.

- Pre-condition: Gaze point map must have been created.
- Post-condition: Gaze point map of the data is viewable and can be further analysed.
- Request Data Structure: The created gaze point map.
- Return Data Structure: Gaze point map is viewable to the user(s).

### 2.3.3 SaveGazePointMap

This activity deals with the saving of gaze point map that are created from their respective recorded data. The gaze point map will be saved on the users computer to be viewed at a later date. This will save only the gaze point map which can later be saved onto a model at a later date.

- Pre-condition: Raw information must have already been recorded so that new gaze point map can be created.
- Post-condition: Gaze point map of the data is saved.

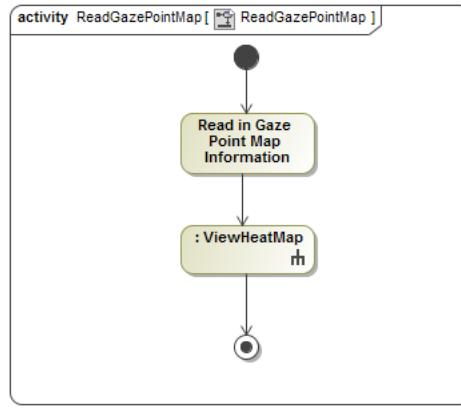


Figure 11: Read Gaze Point Map Activity

- Request Data Structure: Raw eye tracking information
- Return Data Structure: Gaze point map is saved.

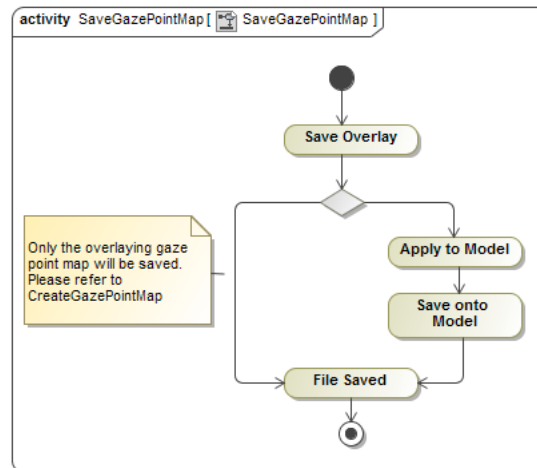


Figure 12: Save Gaze Point Map Activity

## 2.4 Save Gaze Point Maps

The saving of the gaze point map provides important functionality to the system as it will allow the users to save the data which can later be used to either continue research or to compare data. The gaze point map will serve as an overlay for the media type and be placed over the media model. The saving of gaze point map can be divided into three subsections, namely SaveGazeMapOntoModel3D, SaveGazeMapOntoModelVideo and SaveGazeMapOntoModel2D.

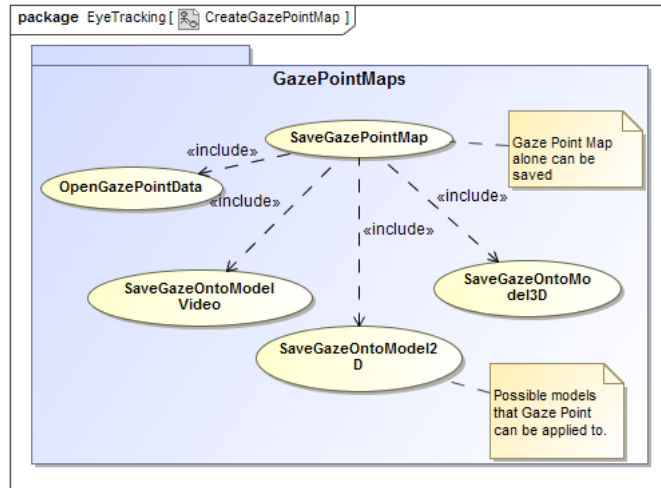


Figure 13: Save Gaze Point Map Use Case

#### 2.4.1 SaveGazeMapOntoModel2D

The gaze point map that is created for a 2D model media type will be saved and applied to the 2D model to see the gaze point map on top of the model to indicate what has been viewed the most by the user(s) test participants.

- Pre-condition: Eye tracking data must have already been recorded.
- Post-condition: Gaze point map of the data is viewable on the 2D model as an overlay.
- Request Data Structure: Raw eye tracking data.
- Return Data Structure: Gaze point map of the data is placed over the 2D Model.

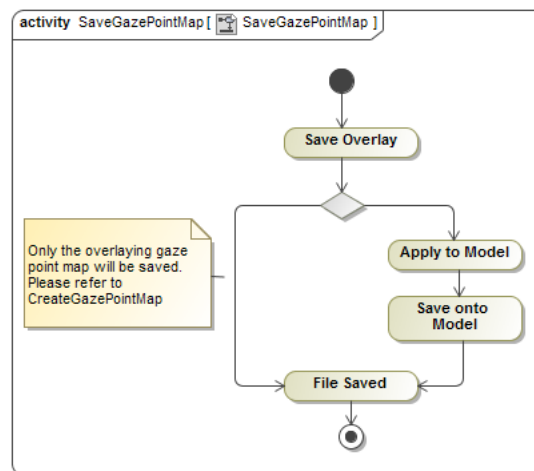


Figure 14: Save on 2D Model Activity

### 2.4.2 SaveGazeMapOntoModelVideo

The gaze point map that is created for a video media type will be saved and then applied to the video to see the gaze point map over the video to indicate what has been viewed the most by the users. The video model is spliced into images at either 30 or 60 frames per second of the video, the gaze point map can then be applied to the created images and recompiled into a new video.

- Pre-condition: Eye tracking data must have been recorded and video model must be available.
- Post-condition: New video is created with the gaze point map on top of it.
- Request Data Structure: Video and raw information of recording.
- Return Data Structure: Gaze point map of the data is placed over each frame of the video.

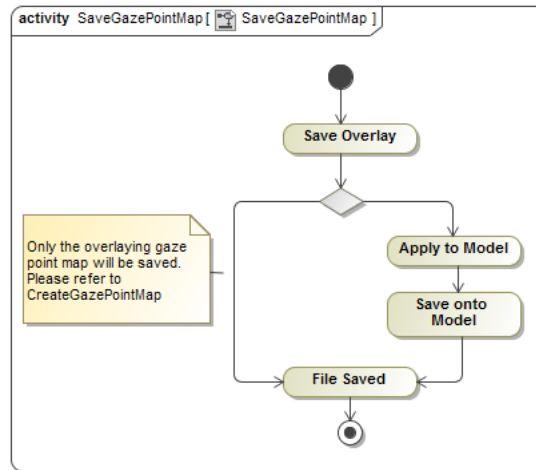


Figure 15: Save on Video Model Activity

### 2.4.3 SaveGazeMapOntoModel3D

The 3D model will have multiple images of it created from different angles Each created image can then have a gaze point map placed over it.

- Pre-condition: Eye tracking data must have already been recorded and 3D model must be available for the creation of images.
- Post-condition: Gaze point map of the data is viewable on the 3D model as an overlay.
- Request Data Structure: Recorded eye tracking information.
- Return Data Structure: Gaze point map of the data is placed over each image of the 3D model.



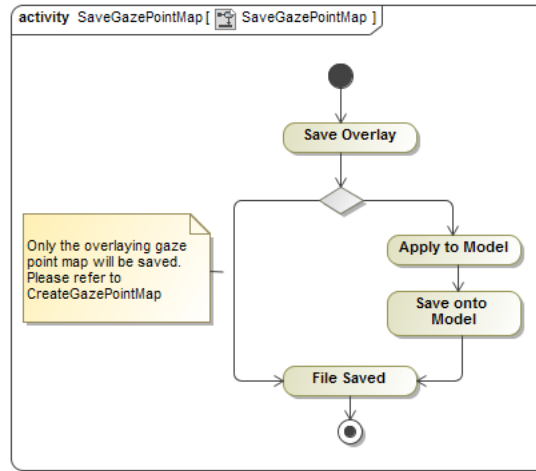


Figure 16: Save on 3D Model Activity

## 2.5 Models

This section details all the different types of media or models that heat-maps can be created for and saved onto. There are three types of models, namely Video, 2D and 3D. The heat maps are created based on the models and their types.

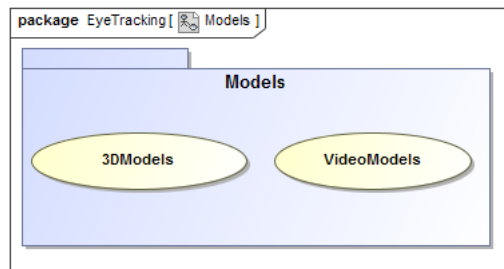


Figure 17: Models Use Case

### 2.5.1 3D Models

3D Models will be opened and rendered by the system, and will then be made viewable to the user. The model will have images created of it at multiple angles. The heat-map can then be saved onto the models individual images as well as opened as the stand alone heat maps of each image.

#### 1. ImportModel

This will allow for a 3D model media type to be imported into the system for further processing. The imported model can be used for recording data once the relevant images have been created and creating a relevant heat-map based on it which can then be applied to it at a later stage.

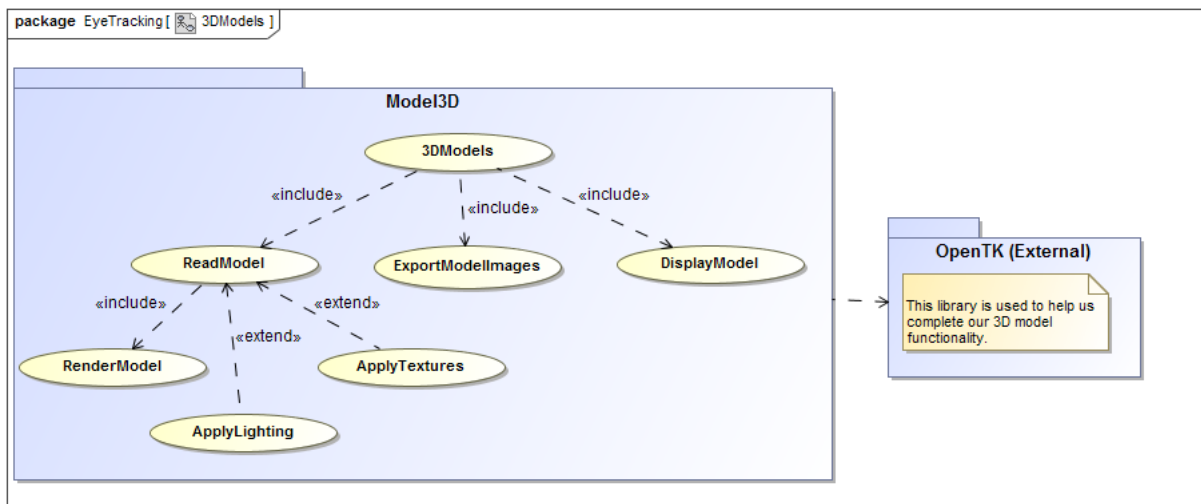


Figure 18: 3D Models Use Case

- Pre-condition: 3D model must exist.
- Post-condition: Images can be created and recording can be done on the model.
- Request Data Structure: The 3D model that user wishes to perform eye tracking on.
- Return Data Structure: Images of model will be created and returned.

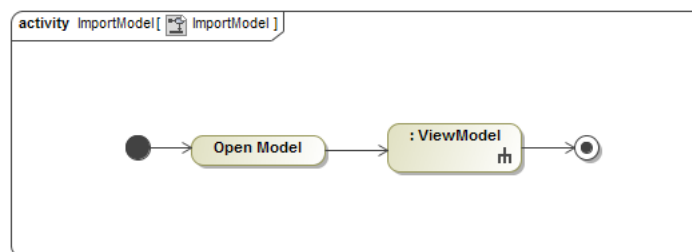


Figure 19: Import 3D Model Activity

2. ViewModel Once the model is imported the model and images created, the will be available for viewing and further processing decided by the user.
  - Pre-condition: 3D model must have been imported and images created.
  - Post-condition: The 3D model images are viewable and can be recorded on.
  - Request Data Structure: The 3D models images.
  - Return Data Structure: Viewable 3D model images.
3. ExportHeatMapModel The exported model will have the heat-map applied to it which can then be exported as a new images of the 3D model so that it can be viewed or used at a later date by the user.

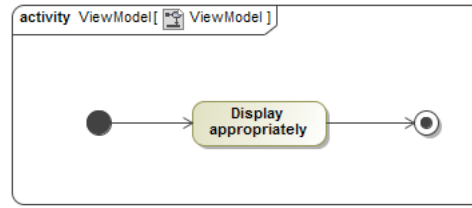


Figure 20: View 3D Model Activity

- Pre-condition: Heat-map must be created and Model Imported.
  - Post-condition: Exported model With heat-map on it to be viewed.
  - Request Data Structure: The eye tracking data must have been recorded and 3D model available
  - Return Data Structure: Model with heat-map applied.
4. Export3DModelImages The model will have images created from various angles of the model. These images can then be used for recording.
- Pre-condition: Selected model must be available.
  - Post-condition: Images will be created of model.
  - Request Data Structure: The selected model.
  - Return Data Structure: Images of 3D model.

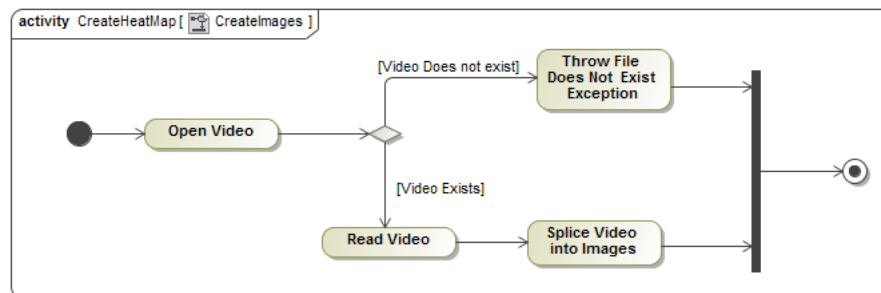


Figure 21: Create 3D Model Images Activity

### 2.5.2 2D Models

2D models will typically refer to images but can also refer to any media that is does not show a third dimension of movement but for the scope of this project will only refer to images. The heat-map can then be created and applied to the image.

#### 1. ImportModel

The 2D models can be imported into the system which can then be process further by the user. The imported model can then have recordings done and heat maps created for it.

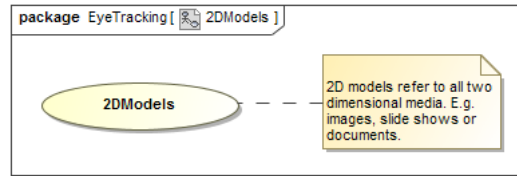


Figure 22: 2D Models Use Case

- Pre-condition: The 2D model must exist, a image preferable.
- Post-condition: Recording can be done on the model.
- Request Data Structure: The desired 2D model.
- Return Data Structure: Rendered image is imported.

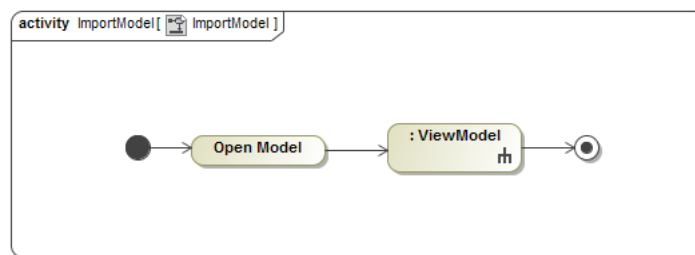


Figure 23: Import 2D Model Activity

2. ViewModel Once the model is imported the users can view the model and eye tracking can then be applied on the model. The model can the be processed further as the user sees fit.

- Pre-condition: The 2D model (image) must have been imported.
- Post-condition: Image is viewable and can be recorded on.
- Request Data Structure: The selected model image.
- Return Data Structure: Viewable image.

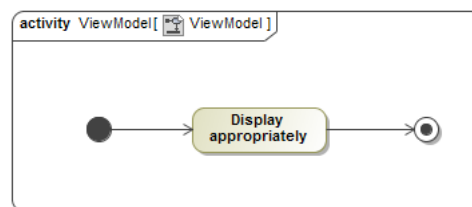


Figure 24: View 2D Model Activity

3. ExportHeatMapModel The exported model will have the heat map applied to it which can then be exported as a new 2D model so that it can be viewed or used at a later date by the user.

- Pre-condition: The eye data must have been recorded and model imported.

- Post-condition: Exported model with heat-map on it to be viewed.
- Request Data Structure: Recorded eye data and imported model.
- Return Data Structure: Model with heat map.

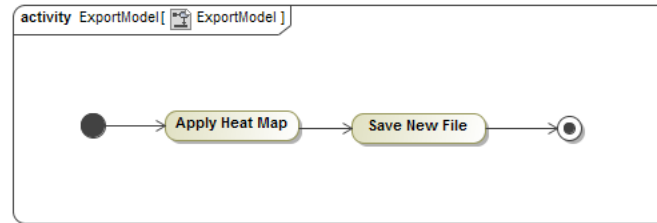


Figure 25: Export 2D Model Activity

### 2.5.3 Video Models

A video can be easily imported into the system. Eye tracking can then be done to the video model and the recorded data can be stored. The data can then be used to create a heat-map that can then be applied to the video second by second.

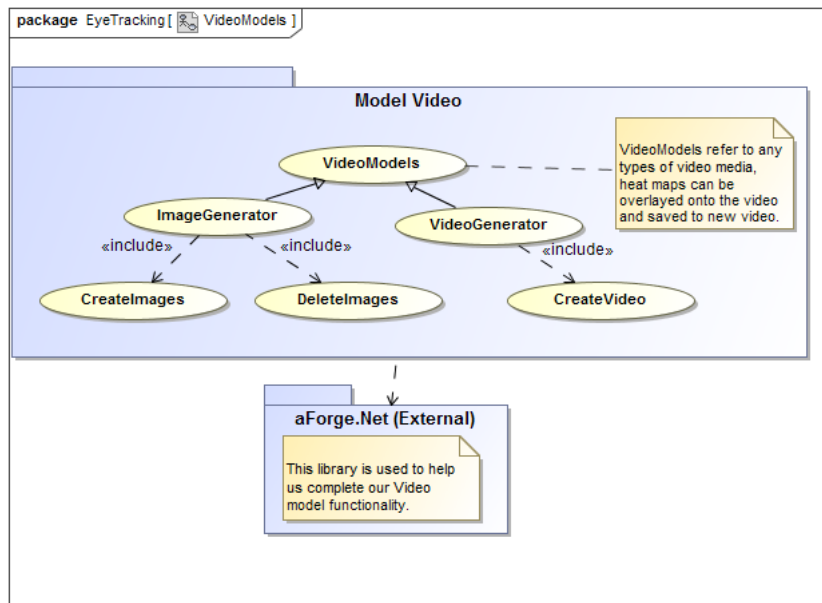


Figure 26: Video Models Use Case

#### 1. ImportModel

The video models can be imported into the system which can then be processed further by the user. The imported model can then have recordings done and heat maps created for it. Heat maps that will be recorded will show a second by second representation of the heat map data.

- Pre-condition: Desired video must exist.
- Post-condition: Recording can be done on the video.
- Request Data Structure: The video model.
- Return Data Structure: Rendered video is imported.

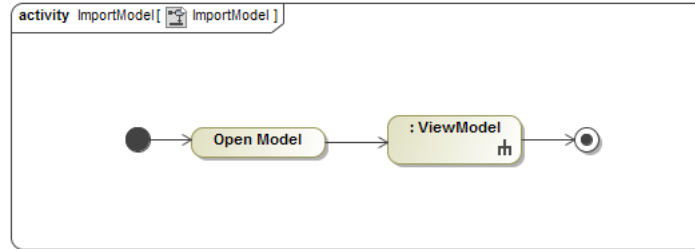


Figure 27: Import Video Model Activity

2. ViewModel Once the model is imported the users can view the model and eye tracking can then be applied on the model. The model can the be processed further as the user sees fit.

- Pre-condition: Desired video model must be available.
- Post-condition: Video can be played and recordings taken.
- Request Data Structure: Requested video.
- Return Data Structure: Playable video.

3. ExportHeatMapModel The exported model will have the heat-map applied to it which can then be exported as a new video model so that it can be viewed or used at a later date by the user. The exported video model will have the heat map recorded onto the original video and saved in the appropriate format.

- Pre-condition: Eye tacking data must have been recorded and desired video imported.
- Post-condition: New video with heat-map overlay will be saved..
- Request Data Structure: Eye tracking data as well as the desired model.
- Return Data Structure: Video with heat map is saved.

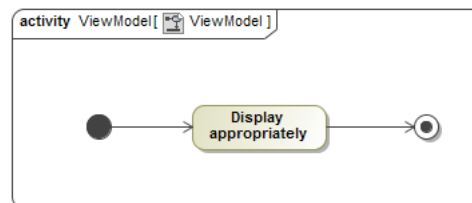


Figure 28: View Video Model Activity

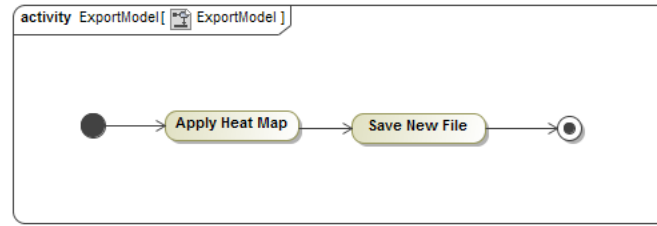


Figure 29: Export 3D Model Activity

## 2.6 Raw Information

When recording happens on the system the raw information is saved. The raw information is important as it forms the basis for creating heat maps and statistics for specific models. The raw information can be saved on the system and processed further by the user. The raw information can then be read as the raw data or put into a statistical format.

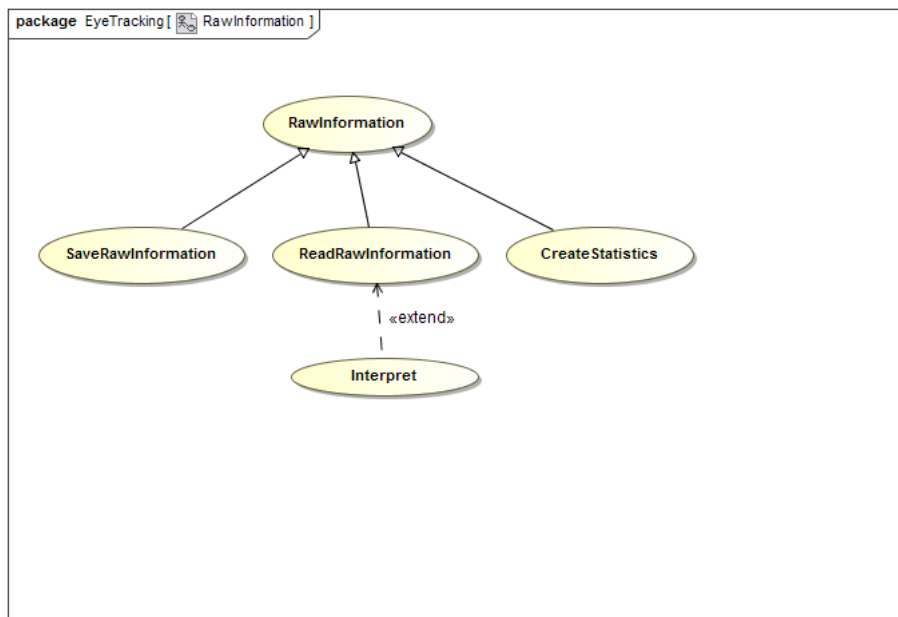


Figure 30: Raw Information Use Case

### 2.6.1 SaveRawInformation

When recording is done on the media the raw information will need to be saved so that it can be used at a later stage. This raw information can then be reopened later for the relevant model and used by user for further processing and interpretation.

- Pre-condition: Recording on a media type must have already taken place.
- Post-condition: Data is saved and can be used later.

- Request Data Structure: Incoming x- and y-axis data coming from the Eye Tribe recording server.
- Return Data Structure: The recorded eye tracking data text file is created.

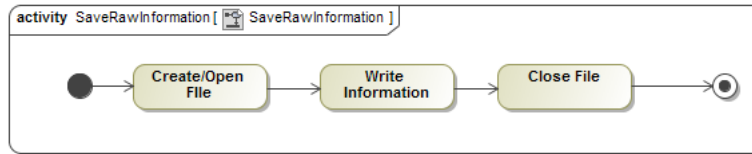


Figure 31: Save Raw Information Activity

### 2.6.2 ReadRawInformation

Data can be read from a previously saved file. This will allow the user to import the raw information for the relevant model which can then be used for comparison or further processing.

- Pre-condition: Recorded data must have been saved in a text file.
- Post-condition: Data can now be used to create heat-maps and statistics.
- Request Data Structure: The recorded eye tracking data text file.
- Return Data Structure: Data array with all data points that can then be further processed by application.

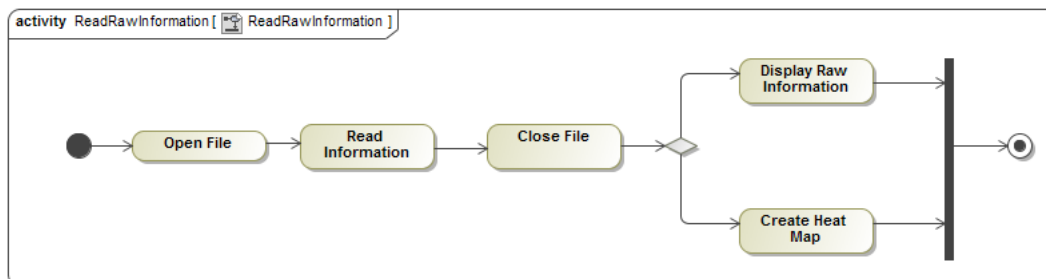


Figure 32: Read Raw Information Activity

### 2.6.3 CreateStatistics

The writing of statistics is part of the Statistics use-case and all information can be found there regarding the sub use case



## 2.7 Statistics

Statistics on all the data will be collected and will then be used to make a statistical page based off the model that can be analysed and then it can be easily compared at any time.

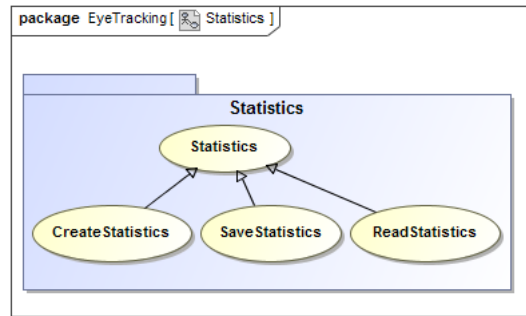


Figure 33: Statistics Use Case

### 2.7.1 CreateStatistics

Using the recorded data we can create statistical data which can then be used for analysis. This data can then be viewed or used to compared data.

- Pre-condition: Recorded data must exist.
- Post-condition: Data is turned into statistical data.
- Request Data Structure: Text file containing recorded data.
- Return Data Structure: Statistics will be created.

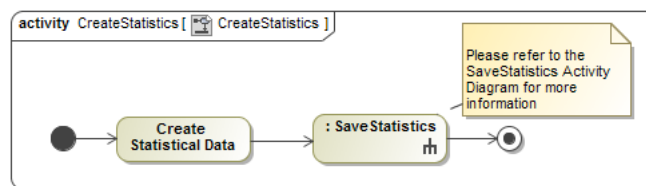


Figure 34: Create Statistics Activity

### 2.7.2 SaveStatistics

The statistical data can be saved into a file so that they can be printed out and be used further by users. This will take all created statistical data and save it into a pdf, excel or csv file.

- Pre-condition: Statistical data must exist.
- Post-condition: Data is put into a report.
- Request Data Structure: Created statistical data.
- Return Data Structure: Save report in pdf, excel or csv format.

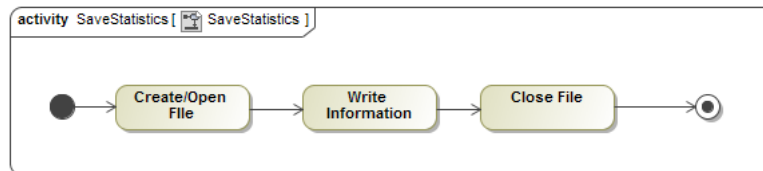


Figure 35: Save Statistics Activity

## 2.8 Record Eyes

The recording of the eyes will be done using the Eye Tribe camera or any other (compatible) eye tracking camera. The data recorded in this process will allow heat maps and statistics to be generated. The heat maps can then be turned into overlays for the appropriate media types. The record eye use case (Figure 36) makes use of sub use cases from other use cases that are listed below, such as SaveRawInformation. The recording of the information is important as this is used throughout the project and its functionality.

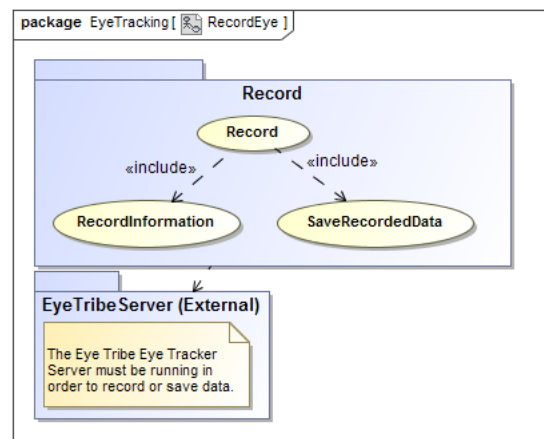


Figure 36: Record Eye Use Case

### 2.8.1 RecordInformation

The data will be recorded and saved into a text file which can be further processed and interpreted, which will then return the relevant information which can then be used to carry out functions in the system.

- Pre-condition: Calibration must have been previously completed and user must be viewing the specified media type, so that recording of data can take place.
- Post-condition: Data is recorded about the eye movements and saved by another function.
- Request Data Structure: N/A.
- Return Data Structure: Raw data in x and y co-ordinates will be returned.

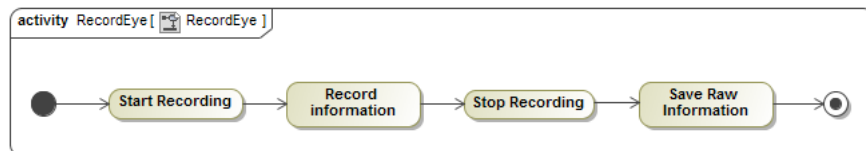


Figure 37: Record Eye Activity

## 3 Required Functionality

### 3.1 Render 3D scene

**Priority: Critical**

Given a 3D scene, in a format such as a BLEND file or an OBJ file, the scene must be able to be rendered to a display. The rendered scene will be placed in a window for viewing so that the eye tracker can "see" the scene.

**Pre-condition:** Open the 3D rendering software with the 3D scene file.

**Post-condition:** Have a window open that has the scene loaded and rendered.

### 3.2 Eye-tracking on a 3D scene

**Priority: Critical**

The eye tracking software must be used or augmented in such a way that specific areas in the 3D scene can capture viewing information. This is will be done given information from the eye tracker itself.

**Pre-condition:** Have OGAMA and eye tracking equipment setup and running.

**Post-condition:** Captured information into a file.

### 3.3 Create heat mapped texture

**Priority: Critical**

From the eye tracking information a heat map in the form of a texture map must be produced. The format of the texture will depend on the 3D software used.

**Pre-condition:** Have captured information from OGAMA and eye tracking equipment.

**Post-condition:** Captured information made into a heat map image.

### 3.4 Map heat mapped texture back onto scene

**Priority: Critical**

The heat mapped texture of the object must be able to be mapped correctly back onto the scene, that is the placement must be as was the original texture.

**Pre-condition:** Have heat mapped texture..

**Post-condition:** 3D scene now has new heat mapped texture applied.

### 3.5 Real-time heat mapping

**Priority: Nice-to-have**

As the scene is being viewed it will also be coloured according to the eye trackers information.

**Pre-condition:** Have OGAMA and eye tracking equipment setup and running.

**Post-condition:** Captured information used to colour the scene.

### 3.6 Interactive Maps

**Priority: Important**

This function involves overlaying a map with data captured by the Eye tribe eye tracking technology and stored using the OGAMA freeware. The data will be retrieved from a database which has already been integrated into the OGAMA freeware that will be used for this project. Through the use of the analytical features provided by OGAMA, its functionality will be extended into the visualization of the information it has stored. This includes the attention maps, scan paths and replay functions.

**Pre-condition:** Have the Eye tribe eye tracking technology running during the activity.

**Post-condition:** Output a 2D representation on the interactive map with the necessary data overlaying it.

### 3.7 In-Video Eye Tracking

**Priority: Nice-to-have**

This function overlays video frames with eye tracking attention maps while watching a video. This is an especially tricky function due to the fact that OGAMA analytical feature is able to run in parallel with data being viewed, but not render the attention map or scan path at run time.

**Pre-condition:** Have OGAMA freeware integrated into video viewing software. Have the Eye tribe eye tracking technology running during the activity.

**Post-condition:** Run time editing of video frames overlaid with attention maps and scan paths.

### 3.8 Post-video Eye Tracking

**Priority: Nice-to-have**

This function involves rendering a video with an attention map and scan path overlaid after it has been watched. This function is more achievable than the In-video eye tracking because the OGAMA software is able to record eye gaze and movement on image slideshows thus it can be used on a slideshow of frames from the video to render a video coupled with this analyzed data overlaying it.

**Pre-condition:** Have OGAMA freeware integrated into video viewing software. Have the Eye tribe eye tracking technology running during the activity.

**Post-condition:** Output a video with its frames overlaid with attention maps and scan paths.

### 3.9 Suitable Video Output Formats

**Priority: Nice-to-have**

This function aims allow the post-video eye tracking video to be outputted in more than one video format. The default video output provided by OGAMA is AVI, but through this function it would be possible to output videos in more popular formats such as

MP4, WMV, MKV etc. This allows the product to be used by many more devices, some of which have limited memory or only support specific video formats.

**Pre-condition:** Have OGAMA freeware integrated into video viewing software. Have the Eye tribe eye tracking technology running during the activity.

**Post-condition:** Output a video in the format that the user has selected.

## 4 Domain Model

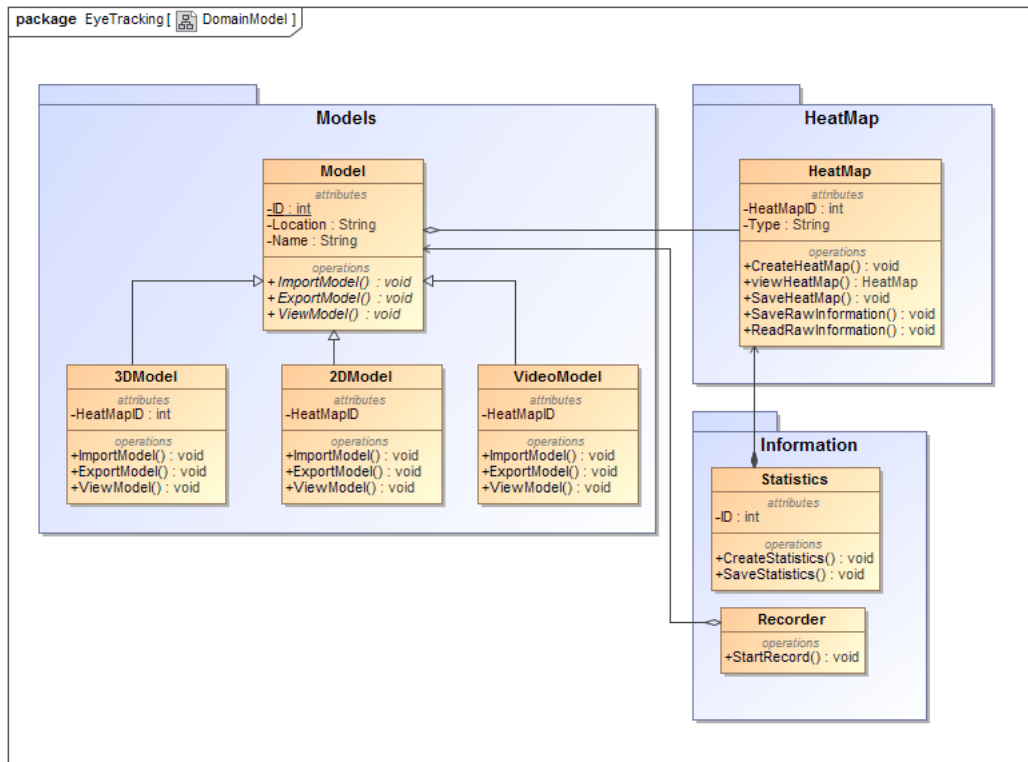


Figure 38: Domain Model