Component Design
For Environment Model Building Tool (EMBT)
Version 1.0

Submitted in partial fulfillment of the requirements of the degree of MSE

Esteban Guillen
CIS 895 – MSE Project
Kansas State University
# Table of Contents

1. Introduction ................................................................................................................. 5
2. Environment Model Builder .......................................................................................... 5
   2.1. Package View ........................................................................................................ 5
   2.2. Application Package .............................................................................................. 5
       2.2.1. Class Descriptions and Diagrams ................................................................. 5
           2.2.1.1. EMBApplication ................................................................................ 5
       2.3. Controller Package ............................................................................................ 6
           2.3.1. Class Descriptions and Diagrams ............................................................. 6
               2.3.1.1. EMBController ............................................................................... 6
               2.3.1.2. EMBBuildingSurfaceMouseHandler ............................................. 6
               2.3.1.3. EMBObjectPropertiesWindow ......................................................... 6
       2.4. View Package ...................................................................................................... 7
           2.4.1. Class Descriptions and Diagrams ............................................................... 7
               2.4.1.1. EMBView ......................................................................................... 7
               2.4.1.2. EMBThreeDimensionalView ........................................................... 7
               2.4.1.3. EMBXMLView ............................................................................... 8
               2.4.1.4. EMBDrawingView .......................................................................... 8
               2.4.1.5. EMBTerrainPreview ......................................................................... 8
               2.4.1.6. EMBTerrainFinder ........................................................................... 9
               2.4.1.7. EMBTerrainView ............................................................................. 9
               2.4.1.8. EMBObjectPreview ......................................................................... 10
               2.4.1.9. EMBObjectFinder .......................................................................... 10
               2.4.1.10. EMBObjectView ......................................................................... 10
               2.4.1.11. EMBBuildingSurface ................................................................. 11
       2.5. Model Package .................................................................................................... 11
           2.5.1. Class Descriptions and Diagrams ............................................................... 11
               2.5.1.1. EMBModel ...................................................................................... 11
               2.5.1.2. EMBEnvironment ................................................................. 12
               2.5.1.3. EMBObject ................................................................................... 12
               2.5.1.4. EMBBasicShape ........................................................................... 13
               2.5.1.5. EMBBox ......................................................................................... 14
               2.5.1.6. EMBCone ....................................................................................... 14
               2.5.1.7. EMBSphere .................................................................................... 15
               2.5.1.8. EMBCylinder ............................................................................... 15
               2.5.1.9. EMBTerrain ................................................................................. 15
               2.5.1.10. EMBObjectLibrary ............................................................... 16
               2.5.1.11. EMBTerrainLibrary ................................................................. 16
3. Environment Object Builder ......................................................................................... 17
   3.1. Package View ........................................................................................................ 17
   3.2. Application Package ............................................................................................. 17
       3.2.1. Class Descriptions and Diagrams ............................................................... 17
           3.2.1.1. EOBApplication ........................................................................... 17
       3.3. Controller Package ............................................................................................ 18
           3.3.1. Class Descriptions and Diagrams ........................................................... 18
3.3.1.1. EOBController ................................................................. 18
3.3.1.2. EOBBoxPropertiesWindow ........................................ 18
3.3.1.3. EOBConPropertiesWindow ........................................ 19
3.3.1.4. EOB Cylinder PropertiesWindow ................................. 20
3.3.1.5. EOB Front Mouse Handler ........................................... 21
3.3.1.6. EOB Side Mouse Handler ............................................ 21
3.3.1.7. EOB Sphere PropertiesWindow .................................... 21
3.3.1.8. EOB Top Mouse Handler ............................................. 22
3.4. View Package .................................................................. 22
3.4.1. Class Descriptions and Diagrams .................................. 22
3.4.1.1. EOBView ................................................................. 22
3.4.1.2. EOB Three Dimensional View .................................... 23
3.4.1.3. EOB Drawing View .................................................. 23
3.4.1.4. EOB XML View ....................................................... 24
3.4.1.5. EOB Side Drawing View ........................................... 24
3.4.1.6. EOB Front Drawing View ......................................... 24
3.4.1.7. EOB Top Drawing View .......................................... 25
3.4.1.8. EOB Object Preview ................................................ 25
3.4.1.9. EOB Object Finder .................................................. 25
3.4.1.10. EOB Object View .................................................... 26
3.5. Model Package ................................................................ 26
3.5.1. Class Descriptions and Diagrams .................................. 26
3.5.1.1. EOBModel ............................................................... 26
3.5.1.2. EOB Object ............................................................. 27
3.5.1.3. EOB Basic Shape .................................................... 27
3.5.1.4. EOB Box ................................................................. 28
3.5.1.5. EOB Cone ............................................................... 29
3.5.1.6. EOB Sphere ............................................................ 30
3.5.1.7. EOB Cylinder ......................................................... 31
3.5.1.8. EOB Object Library ................................................ 31
4. Environment Terrain Builder .............................................. 31
4.1. Package View ................................................................... 32
4.2. Application Package ....................................................... 32
4.2.1. Class Descriptions and Diagrams ................................ 32
4.2.1.1. ETB Application ...................................................... 32
4.3. Controller Package .......................................................... 32
4.3.1. Class Descriptions and Diagrams ................................ 32
4.3.1.1. ETB Controller ......................................................... 32
4.4. View Package .................................................................. 33
4.4.1. Class Descriptions and Diagrams ................................ 33
4.4.1.1. ETB View ................................................................. 33
4.4.1.2. ETB Three Dimensional View .................................. 33
4.4.1.3. ETB XML View ...................................................... 34
4.4.1.4. ETB Drawing View ................................................ 34
4.4.1.5. ETB Building Surface ............................................. 34
4.4.1.6. ETB Terrain Preview .............................................. 35
<table>
<thead>
<tr>
<th>Section</th>
<th>Class/Package</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4.1.7</td>
<td>ETBTerrainFinder</td>
<td>35</td>
</tr>
<tr>
<td>4.4.1.8</td>
<td>ETBTerrainView</td>
<td>35</td>
</tr>
<tr>
<td>4.5</td>
<td>Model Package</td>
<td>36</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Class Descriptions and Diagrams</td>
<td>36</td>
</tr>
<tr>
<td>4.5.1.1</td>
<td>ETBModel</td>
<td>36</td>
</tr>
<tr>
<td>4.5.1.2</td>
<td>ETBTerrain</td>
<td>36</td>
</tr>
<tr>
<td>4.5.1.3</td>
<td>ETBTerrainLibrary</td>
<td>37</td>
</tr>
</tbody>
</table>
1. **Introduction**
   This document will provide brief descriptions and class diagrams of the applications and classes for the EMBT. A detail description of the methods and attributes is provided in the Javadoc documentation.

2. **Environment Model Builder**
   The Environment Model Builder is a graphical tool to create an environment from a terrain and objects. The tool will have a building surface to place the terrain and objects. The user will be able to move the objects to the desired location. There will also be a three dimensional view to observer what the terrain and objects look like in 3D. The following sections will describe the different packages of the Environment Model Builder in detail.

2.1. **Package View**

2.2. **Application Package**

2.2.1. **Class Descriptions and Diagrams**

2.2.1.1. **EMBApplication**
   This class is just intended to have the main method for this program and create the EMBController and set it visible.
2.3. Controller Package

2.3.1. Class Descriptions and Diagrams

2.3.1.1. EMBController
This class is the main frame of the application. It will handle all the menu item actions. It is responsible for loading files, saving files to disk, and saving EMBEnvironments to the library.

<table>
<thead>
<tr>
<th>EMBController</th>
</tr>
</thead>
<tbody>
<tr>
<td>model: EMBModel</td>
</tr>
<tr>
<td>view: EMBView</td>
</tr>
<tr>
<td>obj_lib: EMBObjectLibrary</td>
</tr>
<tr>
<td>terr_lib: EMBTerrainLibrary</td>
</tr>
<tr>
<td>desktop: JDesktopPane</td>
</tr>
<tr>
<td>menuBar: JMenuBar</td>
</tr>
<tr>
<td>fileMenu: JMenu</td>
</tr>
<tr>
<td>newItem: JMenuItem</td>
</tr>
<tr>
<td>openItem: JMenuItem</td>
</tr>
<tr>
<td>exportXML: JMenuItem</td>
</tr>
<tr>
<td>chooser: JFileChooser</td>
</tr>
<tr>
<td>xmlFileName: String</td>
</tr>
<tr>
<td>xmlFile: File</td>
</tr>
</tbody>
</table>

- EMBController(): void
- addView(): void
- setExportXMLMenuItemAction(): void
- setNewMenuItemAction(): void
- setOpenMenuItemAction(): void
- exportXML(): void
- parseXML(): void
- setZoomFactor(): void

2.3.1.2. EMBBuildingSurfaceMouseHandler
This class is responsible for handling mouse events for the building surface. In particular it will wait for mouse clicks and determine if one of the objects was clicked on. If an object is clicked on it will provide a properties window for that object.

<table>
<thead>
<tr>
<th>EMBBuildingSurfaceMouseHandler</th>
</tr>
</thead>
<tbody>
<tr>
<td>selected: EMBObject</td>
</tr>
<tr>
<td>mouseClicked(in evt: MouseEvent): void</td>
</tr>
</tbody>
</table>

2.3.1.3. EMBObjectPropertiesWindow
This class provides the properties window for an object. It will provide controls to move the object to a new location.
2.4. View Package

2.4.1. Class Descriptions and Diagrams

2.4.1.1. EMBView
This class is a container for the EMBThreeDimensionalView, EMBDrawingView, and EMBXMLView.

```
EMBView
  canvas: EMBDrawingView
  xml: EMBXMLView
  threeD: EMBThreeDimensionalView
  pane: JTabbedPane
  panel: JPanel
```

2.4.1.2. EMBThreeDimensionalView
This class will show the three dimensional view of the current EMBEnvironment. From this view the user will be able to view the EMBEnvironment from any angle.
2.4.1.3. EMBXMLView

This class is responsible for displaying the XML definition of the current EMBEnvironment. The XML will represent the contents that will be saved to disk.

2.4.1.4. EMBDrawingView

This class is a container for the EMBBuildingSurface, EMBTerrainPreview, EMOObjectPreview.

2.4.1.5. EMBTerrainPreview

This class is a container for the EMBTerrainFinder and EMBTerrainView. It will also add the currently selected EMBTerrain to the EMBModel.
2.4.1.6. EMBTerrainFinder
This class is responsible for providing a list of all available EMBTerrains in the EMBTerrainLibrary for the user to select.

- current: EMBTerrain
- EMBTerrainFinder(in terrains: Object[]): void
- searchDB(): void
- select(): void
- getCurrent(): EMBTerrain

2.4.1.7. EMBTerrainView
This class is responsible for providing a thumb-nail view of the currently selected EMBTerrain.
2.4.1.8. **EMObjectPreview**
This class is a container for the EMObjectFinder and EMObjectView. It will also add the currently selected EMObject to the EMBBuildingSurface and EMBModel.

```java
EMObjectPreview
- finder: EMObjectFinder
- view: EMObjectView
- scroll: JScrollPane
- add: JButton
- selected: String
- ml: MouseListener

EMObjectPreview().void
updateFinder().void

ButtonListener
```

2.4.1.9. **EMObjectFinder**
This class is responsible for providing a list of all available EMObjects in the EMObjectLibrary for the user to select.

```java
EMObjectFinder
- current: EMObject

EMObjectFinder(In obj: Object[]): void
searchDB(): void
getCurrent(): EMObject
select(): void
```

2.4.1.10. **EMObjectView**
This class is responsible for providing a thumb-nail view of the currently selected EMObject.
2.4.1.11. **EMBBuildingSurface**

This class is responsible for displaying the top 2-D view of the current EMBEnvironment. From this view the user will be able arrange the objects and terrains that have been added to it. This view will also allow for removal of terrains and objects.

```plaintext
- EMBBuildingSurface(): void
- paint(g: Graphics): void
- update(observer: Observable, object: Object): void
- zoomOut(): void
- zoomIn(): void
```
2.5.1.2. EMBEnvironment

This class represents the current environment that is being built. It will be composed of numerous EMBTerrains and EMBObjects. It will also be responsible for building its XML definition.

2.5.1.3. EMBObjec

This class is a collection of EMBBasicShapes that are to be used in the EMBEnvironment. It is also responsible for building its XML definition.
2.5.1.4. EMBBasicShape

This class is the super class for the primitive shapes; EMBBox, EMBCone, EMBSphere, and EMBCylinder. It is responsible for building the XML definition for the primitive shapes.
2.5.1.5. **EMBBBox**

This class represents a box shape. It holds all the information necessary to represent a three dimensional box shape.

```plaintext
EMBBBox
- length: double
- width: double
- height: double
- count: int

  - getName(): String
  - getR(): float
  - getG(): float
  - getB(): float
  - getPosition(): Point3d
  - setHot(in h: double): void
  - setDirection(in d: double): void
  - setStationary(in s: double): void
  - getObjectPosition(): Point3d
  - setObjectPosition(in p: Point3d): void
  - move(in p: Point3d): void
  - writeXML(): String
```

2.5.1.6. **EMBCone**

This class represents a cone shape. It holds all the information necessary to represent a three dimensional cone.

```plaintext
EMBCone
- r: float
- g: float
- b: float
- hot: double
- stationary: double
- direction: double
- name: String

EMBCone(in name: String):
  - getName(): String
  - getR(): float
  - getG(): float
  - getB(): float
  - getPosition(): Point3d
  - setHot(in h: double): void
  - setDirection(in d: double): void
  - setStationary(in s: double): void
  - getObjectPosition(): Point3d
  - setObjectPosition(in p: Point3d): void
  - move(in p: Point3d): void
  - writeXML(): String
```
2.5.1.7. **EMBSphere**
This class represents a sphere shape. It holds all the information necessary to represent a three dimensional sphere.

```
EMBSphere(radius: double, height: double, count: int)
```

2.5.1.8. **EMBCylinder**
This class represents a cylinder shape. It holds all the information necessary to represent a three dimensional cylinder.

```
EMBCylinder(radius: double, height: double, count: int)
```

2.5.1.9. **EMBTerrain**
This class represents a terrain for the environment. It will consist of an elevation map and a collection of coordinates. The elevation map will specify the height of all the desired locations. The terrain will be represented by strips of triangles.
2.5.1.10. **EMBObjectLibrary**

This class will hold all the EMOObjects that are saved to the object library.

* **EMBObjectLibrary**
  - objects: Hashtable
  - libDir: File

* EMBObjectLibrary(): void
* query(name: String): EMOObject
* getKeys(): Enumeration
* addToLibrary(o: EMOObject, in name: String): void
* getData(): Enumeration
* addToModel(in obj: String): void
* parseXML(in xmlFile: File): void

2.5.1.11. **EMBTerrainLibrary**

This class will hold all the EMBTerrains that are saved to the object library.

* **EMBTerrainLibrary**
  - terrain: TriangleStripArray
  - shape: Shape3D
  - name: String
  - FLOOR_LEN: int
  - row[0..*]: int
  - green: Color3f
  - floorBG: BranchGroup
  - coords: ArrayList
  - ng: NormalGenerator
  - gi: GeometryInfo
  - points[0..*]: Point3f

* EMBTerrain(in name: String): void
* createCoords(x: int, in z: int, in coords: ArrayList): void
* createGeometry(coords: ArrayList, in col: Color3f): void
* createAppearance(): void
* updatePoint(l: int, in p: Point3f): void
* getBG(): BranchGroup
* writeXML(): String
* getHeight(index: int): float
* getData(): Point3f
* getName(): String
* setName(name: String): void
### Environment Object Builder

The Environment Object Builder is a graphical tool for building complex shapes/object from primitive shapes. The tool will have three drawing surfaces representing a two dimensional view from the top, side, and front. The user will be able to move and resize the primitive shape from any of the three drawing surfaces. There will also be a three dimensional view provided to observe the created object in 3D. Finally there will be a XML view to show the textual description of object. The following sections will describe the packages of the Environment Object Builder.

#### 3.1. Package View

<table>
<thead>
<tr>
<th>Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>EOB Application</td>
</tr>
<tr>
<td>EOB Controller</td>
</tr>
<tr>
<td>EOB Model</td>
</tr>
<tr>
<td>EOB View</td>
</tr>
</tbody>
</table>

#### 3.2. Application Package

**3.2.1. Class Descriptions and Diagrams**

**3.2.1.1. EOBApplication**

This class is just intended to have the main method for this program and create the EOBController and set it visible.

```
EOBApplication()

getData(): HasTable

addToModel(in name: String): void

addData(in t: EMBTerrain, in name: String): void

 curey(in name: String): EMBTerrain

getKeys(): Enumeration

parseXML(in xmlFile: File): void
```
3.3. Controller Package

3.3.1. Class Descriptions and Diagrams

3.3.1.1. EOBController
This class is the main frame of the application. It will handle all the menu item actions. It is responsible for loading files, saving files to disk, and saving EOBObjects to the library.

3.3.1.2. EOBBoxPropertiesWindow
This class provides a JDialog window with controls for modifying a EOBBox.
3.3.1.3. EOBConePropertiesWindow

This class provides a JDialog window with controls for modifying a EOBCone.
3.3.1.4. EOBCylinderPropertiesWindow

This class provides a JDialog window with controls for modifying a EOBCylinder.
3.3.1.5. EOBFrontMouseHandler
This class is responsible for handling mouse events for the front building surface. In particular it will wait for mouse clicks and determine if one of the objects was clicked on. If an object is clicked on it will provide a properties window for that object.

```
@ EOBFrontMouseHandler
  selected: EOBBasicShape
  view: EOBFrontDrawingView

  EOBFrontMouseHandler(in view: EOBFrontDrawingView);
  mouseClicked(in e: MouseEvent): void
```

3.3.1.6. EOBSideMouseHandler
This class is responsible for handling mouse events for the side building surface. In particular it will wait for mouse clicks and determine if one of the objects was clicked on. If an object is clicked on it will provide a properties window for that object.

```
@ EOBSideMouseHandler
  selected: EOBBasicShape
  view: EOBSideDrawingView

  EOBSideMouseHandler(in view: EOBSideDrawingView);
  mouseClicked(in e: MouseEvent): void
```

3.3.1.7. EOBSpherePropertiesWindow
This class provides a JDialog window with controls for modifying a EOBSphere.
3.3.1.8. EOBTopMouseHandler
This class is responsible for handling mouse events for the top building surface. In particular it will wait for mouse clicks and determine if one of the objects was clicked on. If an object is clicked on it will provide a properties window for that object.

3.4. View Package

3.4.1. Class Descriptions and Diagrams

3.4.1.1. EOBView
This class is a container for the EOBThreeDimensionalView, EOBDrawingView, and EOBXMLView.
3.4.1.2. EOBThreeDimensionalView
This class will show the three dimensional view of the current EOBOBJECT. From this view the user will be able to view the EOBOBJECT from any angle.

3.4.1.3. EOBDrawingView
This class is the container for the EOBTOPDRAWINGVIEW, EOBsidEDrawingView, EOBfrontDrawingView, and EOBOBJECTPREVIEW.
3.4.1.4. EOBXMLView
This class is responsible for displaying the XML definition of the current EOBOBJECT. The XML will represent the contents that will be saved to disk.

3.4.1.5. EOBSideDrawingView
This class is responsible for providing a drawing surface for EOBBasicShapes. This view will represent the side view. The user will be able to move and change the properties of the EOBBasicShapes from this view.

3.4.1.6. EOBFrontDrawingView
This class is responsible for providing a drawing surface for EOBBasicShapes. This view will represent the front view. The user will be able to move and change the properties of the EOBBasicShapes from this view.
3.4.1.7. EOBTopDrawingView
This class is responsible for providing a drawing surface for EOBBasicShapes. This view will represent the top view. The user will be able to move and change the properties of the EOBBasicShapes from this view.

3.4.1.8. EOBObjectPreview
This class is the container for the EOBObjectFinder and EOBObjectView. It will also add the currently selected EOBBasicShapes to the EOBSideDrawingView, EOBFrontDrawingView, and EOBTopDrawingView.

3.4.1.9. EOBObjectFinder
This class is responsible for providing a list of all available EOBOObjects in the EOBObjectLibrary for the user to select.
3.4.1.10. **EOBObjectView**

This class is responsible for providing a thumb-nail view of the currently selected EOBObject.

![Diagram of EOBObjectFinder](image)

- current: EOBObject
- `EOBObjectFinder(in obj: Object[]): void`
- `searchDB(): void`
- `getCurrent(): EOBObject`
- `select(): void`

---

3.5. **Model Package**

3.5.1. **Class Descriptions and Diagrams**

3.5.1.1. **EOBModel**

This class is responsible for holding the current EOBObject that is being built and making it available to other classes.
3.5.2. EOBObject
This class is a collection of EOBBasicShapes that are to be used in the environment. It is also responsible for building its XML definition.

3.5.1.3. EOBBasicShape
This class is the super class for the primitive shapes; EOBBBox, EOBCone, EOBSphere, and EOBCylinder. It is responsible for building the XML definition for the primitive shapes.
3.5.1.4. EOBBox

This class represents a box shape. It holds all the information necessary to represent a three dimensional box shape.
This class represents a cone shape. It holds all the information necessary to represent a three dimensional cone.
3.5.1.6. EOBSphere

This class represents a sphere shape. It holds all the information necessary to represent a three dimensional sphere.
3.5.1.7. EOBCylinder

This class represents a cylinder shape. It holds all the information necessary to represent a three dimensional cylinder.

```
EOBCylinder
```

- `top`: Double
- `side`: Double
- `front`: Double
- `height`: double
- `local_height`: double
- `radius`: double
- `local_radius`: double
- `count`: int

```
EOBCylinder(in x: double, in y: double, in z: double, in r: double, in h: double, in g: float, in b: float)
drawFront(in g: Graphics2D): void
drawSide(in g: Graphics2D): void
drawTop(in g: Graphics2D): void
getHeight(): double
getRadius(): double
rebuild(): void
modify(in h: double, in r: double): void
writeXML(): String
getTopBounds(): Rectangle2D
getSideBounds(): Rectangle2D
getFrontBounds(): Rectangle2D
```

3.5.1.8. EOBOObjectLibrary

This class will hold all the EOBOObjects that are saved to disk.

```
EOBOObjectLibrary
```

- `objects`: Hashtable
- `libDir`: File

```
EOBOObjectLibrary(): void
addToModel(in obj: String): void
query(in name: String): EOBOObject
getKeys(): Enumeration
addToLibrary(in obj: EOBOObject, in name: String):
exportXML(in string: String): void
getData(): Enumeration
parseXML(in xmlFile: File): void
```

4. Environment Terrain Builder

The Environment Terrain Builder is a graphical tool for building surfaces to be used by the Environment Model Builder. The tool will provide a building surface to allow the user to specify the elevation of a given region on the surface. The user will also be able to define physical properties of
the surface. A three dimensional view will be provided to observe how the terrain will look in 3D. Finally a XML view will be provided to give a textual description of the terrain. The following sections will describe the packages of the Environment Terrain Builder.

4.1. Package View

4.2. Application Package

4.2.1. Class Descriptions and Diagrams

4.2.1.1. ETBApplication

This class is just intended to have the main method for this program and create the ETBController and set it visible.

4.3. Controller Package

4.3.1. Class Descriptions and Diagrams

4.3.1.1. ETBController

This class is the main frame of the application. It will handle all the menu item actions. It is responsible for loading files, saving files to disk, and saving ETBTerrains to the library.
4.4. View Package

4.4.1. Class Descriptions and Diagrams

4.4.1.1. ETBView

This class is a container for the ETBThreeDimensionalView, ETBDrawingView, and ETBXMLView.

4.4.1.2. ETBThreeDimensionalView

This class will show the three dimensional view of the current ETBTerrain. From this view the user will be able to view the ETBTerrain from any angle.
4.4.1.3. ETBXMLView
This class is responsible for displaying the XML definition of the current ETBTerrain. The XML will represent the contents that will be saved to disk.

4.4.1.4. ETBDrawingView
This class is a container for the ETBBuildingSurface and ETBTerrainPreview.

4.4.1.5. ETBBuildingSurface
This class is responsible for providing a surface to create an ETBTerrain. It will provide the user an interface to specify the elevation of sections of the ETBTerrain.
4.4.1.6. ETBTerrainPreview
This class is a container for the ETBTerrainView and ETBTerrainFinder. It will provide the ability to add the currently selected ETBTerrain to the ETBBuildingSurface.

4.4.1.7. ETBTerrainFinder
This class is responsible for providing a list of all available ETBTerrains in the ETBTerrainLibrary for the user to select.

4.4.1.8. ETBTerrainView
This class is responsible for providing a thumb-nail view of the currently selected ETBTerrain.
4.5. Model Package

4.5.1. Class Descriptions and Diagrams

4.5.1.1. ETBModel
This class is responsible for holding the current ETBTerrain that is being built and making it available to other classes.

4.5.1.2. ETBTerrain
This class represents a terrain for the environment. It will consist of an elevation map and a collection of coordinates. The elevation map will specify the height of all the desired locations. The terrain will be represented by strips of triangles.
4.5.1.3. ETBTerrainLibrary
This class will hold all the EOBTerrains that are saved to disk.

```
 ETBTerrainLibrary
    terrains: HashTable
    libDir: File
```

```java
ETBTerrainLibrary(): void
addData(in t: ETBTerrain, in name: String): void
query(in name: String: ETBTerrain
getKeys(): Enumeration
getData(): Enumeration
addToModel(in name: String): void
exportXML(in name: String): void
parseXML(in xmlFile: File): void
```