Architecture 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 ..................................................................................... 6
         2.2.1.1. EMBApplication .................................................................................. 6
   2.3. Controller Package ........................................................................................... 6
      2.3.1. Class Descriptions ..................................................................................... 6
         2.3.1.1. EMBController .................................................................................... 6
   2.4. View Package .................................................................................................... 6
      2.4.1. Class Descriptions ..................................................................................... 6
         2.4.1.1. EMBView ............................................................................................ 6
         2.4.1.2. EMBThreeDimensionalView ................................................................ 7
         2.4.1.3. EMBXMLView .................................................................................... 7
         2.4.1.4. EMBDrawingView .............................................................................. 7
         2.4.1.5. EMBTerrainPreview .......................................................................... 7
         2.4.1.6. EMBTerrainFinder .............................................................................. 7
         2.4.1.7. EMBTerrainView ................................................................................ 7
         2.4.1.8. EMOBJECTPreview ............................................................................ 7
         2.4.1.9. EMOBJECTFinder ............................................................................. 7
         2.4.1.10. EMOBJECTView .............................................................................. 7
         2.4.1.11. EMBBuildingSurface ....................................................................... 7
   2.5. Model Package .................................................................................................. 8
      2.5.1. Class Descriptions ..................................................................................... 8
         2.5.1.1. EMBModel .......................................................................................... 8
         2.5.1.2. EMBEnvironment ............................................................................. 8
         2.5.1.3. EMOBJECT ....................................................................................... 8
         2.5.1.4. EMBBasicShape ............................................................................... 8
         2.5.1.5. EMBBox ............................................................................................. 8
         2.5.1.6. EMBCone ............................................................................................ 8
         2.5.1.7. EMBSphere ....................................................................................... 8
         2.5.1.8. EMBCylinder ..................................................................................... 8
         2.5.1.9. EMBTerrain ....................................................................................... 9
         2.5.1.10. EMBEnvironmentLibrary ................................................................. 9
         2.5.1.11. EMOBJECTLibrary ......................................................................... 9
         2.5.1.12. EMBTerrainLibrary ....................................................................... 9
   2.6. Sequence Diagrams .......................................................................................... 9
      2.6.1. Opening a Saved Environment ................................................................. 9
      2.6.2. Adding a Terrain to the Building Surface ................................................. 9
      2.6.3. Adding an Object to the Building Surface .............................................. 9
      2.6.4. Saving a Environment to Disk .................................................................. 10
3. Environment Object Builder .................................................................................. 10
   3.1. Package View .................................................................................................. 11
   3.2. Application Package ....................................................................................... 11
3.2.1. Class Descriptions .................................................................................. 11
  3.2.1.1. EOBApplication .............................................................................. 11
3.3. Controller Package .................................................................................... 11
  3.3.1. Class Descriptions ............................................................................... 12
     3.3.1.1. EOBController ........................................................................... 12
3.4. View Package ........................................................................................... 12
  3.4.1. Class Descriptions ................................................................................ 12
     3.4.1.1. EOBView ..................................................................................... 12
     3.4.1.2. EOBThreeDimensionalView .......................................................... 12
     3.4.1.3. EOBDrawingView ........................................................................ 12
     3.4.1.4. EOBXMLView ............................................................................ 12
     3.4.1.5. EOBSideDrawingView .................................................................. 12
     3.4.1.6. EOBFrontDrawingView ................................................................ 13
     3.4.1.7. EOBTopDrawingView .................................................................... 13
     3.4.1.8. EOBObjectPreview ...................................................................... 13
     3.4.1.9. EOBObjectFinder ......................................................................... 13
     3.4.1.10. EOBObjectView ......................................................................... 13
3.5. Model Package .......................................................................................... 13
  3.5.1. Class Descriptions ................................................................................. 13
     3.5.1.1. EOBModel ................................................................................... 13
     3.5.1.2. EOBObject ................................................................................... 13
     3.5.1.3. EOBBasicShape ........................................................................... 14
     3.5.1.4. EOBBox ....................................................................................... 14
     3.5.1.5. EOBCon ......................................................................................... 14
     3.5.1.6. EOBSphere .................................................................................. 14
     3.5.1.7. EOB Cylinder .............................................................................. 14
     3.5.1.8. EOBObjectLibrary ....................................................................... 14
3.6. Sequence Diagrams ................................................................................... 14
  3.6.1. Adding Two Box shapes to the Three Drawing Surfaces ....................... 14
  3.6.2. Modifying and Moving Two Box Shapes .............................................. 15
4. Environment Terrain Builder .......................................................................... 15
  4.1. Package View ............................................................................................ 16
  4.2. Application Package ................................................................................ 16
     4.2.1. Class Descriptions ......................................................................... 16
        4.2.1.1. ETBApplication ................................................................. 16
  4.3. Controller Package .................................................................................. 16
     4.3.1. Class Descriptions ......................................................................... 17
        4.3.1.1. ETBController ................................................................. 17
  4.4. View Package .......................................................................................... 17
     4.4.1. Class Descriptions ......................................................................... 17
        4.4.1.1. ETBView ................................................................. 17
        4.4.1.2. ETBThreeDimensionalView ............................................ 17
        4.4.1.3. ETBXMLView ............................................................... 17
        4.4.1.4. ETBDrawingView ........................................................... 17
        4.4.1.5. ETBBuildingSurface ....................................................... 18
        4.4.1.6. ETBTerrainPreview ......................................................... 18
4.4.1.7. ETB Terrain Finder ................................................................. 18
4.4.1.8. ETB Terrain View ................................................................. 18

4.5. Model Package ........................................................................ 18
4.5.1. Class Descriptions ................................................................. 18
  4.5.1.1. ETB Model ................................................................. 18
  4.5.1.2. ETB Terrain ................................................................. 18
  4.5.1.3. ETB Terrain Library .................................................... 18

4.6. Sequence Diagrams ................................................................. 19
  4.6.1. Modifying the Elevation of a Section of the Terrain .......... 19

5. Formal Specification for the Environment Model Builder ............ 19
  5.1. USE Model ........................................................................ 19
1. **Introduction**
   The purpose of this document is to provide an architectural design for the Environment Model Building Tool. The design will show class diagrams and sequence diagrams. Each class will have a brief description about its purpose. The last section will provide a formal specification of the Environment Model Builder application.

   A detailed definition of all classes can be found at:


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. Finally there is a XML view to show the textual description of the terrain and objects. The following sections will describe the different packages of the Environment Model Builder in detail.

2.1. **Package View**

   ![Package View Diagram]

2.2. **Application Package**

   ![Application Package Diagram]
2.2.1. Class Descriptions

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

![](image)

2.3.1. Class Descriptions

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.

2.4. View Package

![](image)

2.4.1. Class Descriptions

2.4.1.1. EMBView

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

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 EMBBuildingSurface

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.

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

2.4.1.8. EMBObjectPreview
This class is a container for the EMBObjectFinder and EMBObjectView. It will also add the currently selected EMBObject to the EMBBuildingSurface

2.4.1.9. EMBObjectFinder
This class is responsible for providing a list of all available EMBObjects in the EMBObjectLibrary for the user to select.

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

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.
2.5. Model Package

2.5.1. Class Descriptions

2.5.1.1. EMBModel
This class is responsible for holding the current EMBEnvironment that is being built and making it available to other classes.

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

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

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

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

2.5.1.8. EMBCylinder
This class represents a cylinder shape. It holds all the information necessary to represent a three dimensional cylinder.
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 a mesh of polygons with the vertices mapping to the collection of coordinates.

2.5.1.10. **EMBEnvironmentLibrary**
This class will hold all the EMBEnvironments that are saved to disk.

2.5.1.11. **EMBObjectLibrary**
This class will hold all the EMBObjects that are saved to disk.

2.5.1.12. **EMBTerrainLibrary**
This class will hold all the EMBTerrains that are saved to disk.

### 2.6. Sequence Diagrams

The following sequence diagrams show some of the main functions that the Environment Model Builder will perform

#### 2.6.1. Opening a Saved Environment

The following sequence diagram show the sequence of actions involved in reading in a saved environment.

#### 2.6.2. Adding a Terrain to the Building Surface

The following sequence diagram shows searching for a terrain and then adding it to the building surface.

#### 2.6.3. Adding a Object to the Building Surface

The following sequence diagram shows searching for an object and then adding it to the building surface.
2.6.4. Saving a Environment to Disk

The following sequence diagrams shows how an environment will be saved to disk.

3. 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

3.2. Application Package

3.2.1. Class Descriptions

3.2.1.1. EOBAplication

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

3.3. Controller Package
3.3.1. Class Descriptions

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.

3.4. View Package

3.4.1. Class Descriptions

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. The user will also be able to zoom-in and zoom-out 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. The user will also be able to zoom-in and zoom-out 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. The user will also be able to zoom-in and zoom-out 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 EOBOObject.

3.5. Model Package

3.5.1. Class Descriptions

3.5.1.1. EOBModel
This class is responsible for holding the current EOBOObject that is being built and making it available to other classes.

3.5.1.2. EOBOObject
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; EOBBox, 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.

3.5.1.5. EOBCone
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.

3.5.1.8. EOBOBJECTLibrary
This class will hold all the EOBObjects that are saved to disk.

3.6. Sequence Diagrams
The following sequence diagrams show some of the main functions that the Environment Object Builder will perform.

3.6.1. Adding Two Box shapes to the Three Drawing Surfaces
The following sequence diagram shows adding two box shapes to the drawing surfaces.
3.6.2. Modifying and Moving Two Box Shapes

The following sequence diagram show selecting a box and moving it. It also shows selecting a box and modifying its dimensions.

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

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

ETBController

save(f : File)
open(f : File)
parseXML()
exportXML()
setZoomFactor()
4.3.1. Class Descriptions

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

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

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 a mesh of polygons with the vertices mapping to the collection of coordinates.

4.5.1.3. ETBTerrainLibrary
This class will hold all the EOBTerrains that are saved to disk.
4.6. Sequence Diagrams
The following sequence diagram shows the main function of the Environment Terrain Builder.

4.6.1. Modifying the Elevation of a Section of the Terrain
The following sequence diagram shows how a user would set the elevation for a section of the terrain.

5. Formal Specification for the Environment Model Builder
The formal specification is limited to the Environment Model Builder.

5.1. USE Model

model emb

--
-- CLASSES
--

class Point2d
end

class Point3d
end

class File
class Shape3D
end

--
-- APPLICATION PACKAGE
--

class EMBApplication
operations
  init()
end

--
-- CONTROLLER PACKAGE
--

class EMBController
operations
  save(f : File)
  open(f : File)
  exportXML()
  parseXML()
  setZoomFactor()
end

--
-- VIEW PACKAGE
--

class EMBView
end

class EMBThreeDimensionalView
operations
  add(s : Shape3D)
end

class EMBXMLView
operations
  showXML()
end

class EMBDrawingView
end

class EMBBuildingSurface
operations
  paint()
  zoomOut()
  zoomIn()
  select(p : Point2d)
end
class EMBObjectPreview
operations
  addObject()
end

class EMBObjectView
operations
  display(s : Shape3D)
end

class EMBObjectFinder
operations
  searchDB()
  getCurrent() : EMBObject
  select()
end

class EMBTerrainPreview
operations
  addTerrain()
end

class EMBTerrainView
operations
  display(s : Shape3D)
end

class EMBTerrainFinder
operations
  searchDB()
  getCurrent() : EMBTerrain
  select()
end

--
-- MODEL PACKAGE
--

class EMBEnvironmentLibrary
operations
  getData() : Set(EMBEnvironment)
  addData(e : EMBEnvironment)
end

class EMBObjectLibrary
operations
  getData() : Set(EMBOBJECT)
  addData(o : EMBObject)
end

class EMBTerrainLibrary
operations
  getData() : Set(EMB Terrain)
  addData(t : EMB Terrain)
class EMBModel
operations
getData() : EMBEnvironment
addObject(o : EMBObject)
addTerrain(t : EMBTerrain)
deleteObject(o : EMBObject)
deleteTerrain(t : EMBTerrain)
end

class EMBEnvironment
attributes
  name : String
operations
  writeXML() : String
  addObject(o : EMBObject)
  addTerrain(t : EMBTerrain)
deleteObject(o : EMBObject)
deleteTerrain(t : EMBTerrain)
end

class EMBTerrain
attributes
  name : String
operations
  getHeight(x : Real, z : Real) : Real
writeXML() : String
end

class EMBObject
attributes
  name : String
  x : Real
  y : Real
  z : Real
  length : Real
  width : Real
  height : Real
operations
  writeXML() : String
  addShape(s : EMBBasicShape)
  move(p : Point3d)
end

class EMBBasicShape
attributes
  name : String
  x : Real
  y : Real
  z : Real
operations
  writeXML() : String
end

class EMBBox < EMBBasicShape
attributes
    length : Real
    width : Real
    height : Real
end

class EMBCylinder < EMBBasicShape
attributes
    height : Real
    radius : Real
end

class EMBCone < EMBBasicShape
attributes
    height : Real
    radius : Real
end

class EMBSphere < EMBBasicShape
attributes
    radius : Real
end

--
-- ASSOCIATIONS
--

--
-- VIEW PACKAGE
--

association ThreeD between
  EMBView[1]
  EMBThreeDimensionalView[1] role threeD
end

association Canvas between
  EMBView[1]
  EMBDrawingView[1] role canvas
end

association XML between
  EMBView[1]
  EMBXMLView[1] role xml
end

association TerrPreview between
  EMBDrawingView[1]
  EMBTerrainPreview[1] role terrainPrev
end

association Builder between
  EMBDrawingView[1]
  EMBBuildingSurface[1] role builder
association ObjectDatabase between
  EMBObject[1..*] role objects
  EMBObjectLibrary[1]
end

association Shapes between
  EMBObject[1]
  EMBBasicShape[1..*] role shapes
end

-- CONSTRAINTS
--

constraints
--
-- Relations
--

-- Unique names of environments in Environment Library
--
context e : EMBEnvironmentLibrary
inv UniqueNameEnvironmentLibrary:
  e.environments->forAll(p1,p2 | p1 <> p2
      implies p1.name <> p2.name)

-- Unique names of objects in Object Library
--
context o : EMBObjectLibrary
inv UniqueNameObjectLibrary:
  o.objects->forAll(p1,p2 | p1 <> p2
      implies p1.name <> p2.name)

-- Unique names of terrains in Terrain Library
--
context t : EMBTerrainLibrary
inv UniqueNameTerrainLibrary:
  t.terrains->forAll(p1,p2 | p1 <> p2
      implies p1.name <> p2.name)

-- Unique names for all shapes of an object
--
context obj : EMBObject
inv UniqueNameObjectShapes:
  obj.shapes->forAll(p1,p2 | p1 <> p2
      implies p1.name <> p2.name)

-- Every box has positive length, width and height
--
context b : EMBBox
inv BoxPositiveLength:
-- Every object has positive length
inv BoxPositiveWidth:
b.width > 0
inv BOXPositiveHeight:
b.height > 0

--

-- Every sphere has positive radius

context s : EMBSphere
inv SpherePositiveRadius:
s.radius > 0

--

-- Every cylinder has positive height and radius

context cyl : EMBCylinder
inv CylinderPositiveHeight:
cyl.height > 0
inv CylinderPositiveRadius:
cyl.radius > 0

--

-- Every cone has positive height and radius

context c : EMBCone
inv ConePositiveHeight:
c.height > 0
inv ConePositiveRadius:
c.radius > 0

--

context EMBEnvironment::deleteObject(o : EMBObject)
pre  Current: objects->includes(o)
post Deleted: objects = objects@pre->excluding(o)

-- Added objects must be unique

context EMBEnvironment::addObject(o : EMBObject)
pre  Current: objects->excludes(o)
post Added: objects = objects@pre->including(o)