Project Evaluation

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 ......................................................................................................................... 3
2. Problems Encountered ........................................................................................................ 3
   2.1. Understanding Project Goals ....................................................................................... 3
   2.2. Learning Java3D .......................................................................................................... 3
   2.3. Balancing Work ........................................................................................................... 3
   2.4. Features to Implement ................................................................................................. 3
3. Source Lines of Code .......................................................................................................... 3
4. Project Duration .................................................................................................................. 4
5. Lessons Learnt ..................................................................................................................... 6
6. Future Work .......................................................................................................................... 7
   6.1. Robot Builder (New Application/Mode) ................................................................. 7
   6.2. Object Builder ............................................................................................................ 7
   6.3. Terrain Builder ........................................................................................................... 7
   6.4. Environment Builder ................................................................................................... 7
1. **Introduction**
   This document will present a summary of my experiences encountered throughout my MSE project.

2. **Problems Encountered**
   This section describes some of my biggest problems encountered during my MSE project.

   2.1. **Understanding Project Goals**
   One of the first problems I had was to fully understand the goals of this project. At a very high level the project was easy to understand, but once I started to think about some of the details I discovered I needed to ask more questions.

   After I got a decent understanding of the project goals my next major problem was visualizing how the user would interact with the application. The project was very graphical and has a lot of user interaction, which made the graphical user interface one of the most important parts of the project. I would say designing the user interface was the most challenging parts of the project.

   2.2. **Learning Java3D**
   The Java 3D part of the project posed problems in the beginning. I had a great deal of trouble finding good examples of Java 3D. The Java 3D web site had some information but it was very out of date. After many hours of searching the web I found a few good examples to get me started. I also purchased “Java 3D API Jump-Start”. It is a must read for beginner Java 3D programmers.

   2.3. **Balancing Work**
   The size of this project grew very quickly. At the end there were 3 different stand alone programs. I found it difficult to manage my time between the 3 programs. I always felt like I was neglecting some part of the project.

   2.4. **Features to Implement**
   As mentioned above the project grew very quickly and new feature were constantly being discussed. One of the design goals was to “think big”, that is to think of all the possible features that would be nice to have. It wasn’t practical to implement all the features for my MSE project, so I had to decide which features to implement. In the end my objective was to keep things simple and get everything to work. This approach only allowed about 75% of the features to be implemented, but those that were implemented were done with great care.

3. **Source Lines of Code**
   The first estimate for SLOC was made during Phase 1 and it was estimated to be 2500 SLOC. This estimate was driven from the current prototype and similar examples. I believe this estimate was low because I didn’t have a full understanding of all the features that would be implemented. That estimate was also low because at that point in the project I was planning on just having one large application as opposed to the 3 application that were developed. The next estimate was in Phase 1 and it was estimated to be 4800 SLOC. This estimate was driven from the executable prototype. The executable prototype gave me a better idea of what features would be implemented and how much code it would take to implement those features. The 4800 SLOC was a reasonable estimate to the actual 5372 SLOC.

   The following is a break down of the SLOC required for each application.

   Environment Model Builder = 2237
   Environment Object Builder = 2315
   Environment Terrain Builder = 820
   Total = 5372 source lines of code
4. **Project Duration**

The following table shows the expected vs. actual completion times for each phase of the project:

<table>
<thead>
<tr>
<th>Phase</th>
<th>Expected Finish Time</th>
<th>Actual Finish Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase 1</td>
<td>January 28, 2004</td>
<td>March 17, 2004</td>
</tr>
<tr>
<td>Phase 2</td>
<td>May 21, 2004</td>
<td>June 9, 2004</td>
</tr>
<tr>
<td>Phase 3</td>
<td>August 15, 2004</td>
<td>July 21, 2004</td>
</tr>
</tbody>
</table>

Both Phase 1 and Phase 2 were delayed. This was due to lack of preparation of the required documents and code. The delay in Phase 1 was mostly because of the prototype. I wasn’t happy with the interface at the end of January, so I waited until March when I was more comfortable with the prototype. The delay in Phase 2 was mainly due to the extended time the Architecture Design took me. It was also delayed because my CIS 844 final project took up much of my time at the end of the semester. Phase 3 was moved up a few weeks. This was driven by the fact that I needed to have my graduate material turned in by July 30.

The following lists how much time was spent on each Phase.

- Phase 1 = 3870 minutes or 65 hours
- Phase 2 = 5970 minutes or 100 hours
- Phase 3 = 16410 minutes or 273 hours
- Total = 438 hours

During Phase 1 using the COCOMO model I estimated that the total project would take 840 hours. This estimate was almost twice as long as the actual time. The reason for the overestimate was that COCOMO assumes the project is a large industrial strength project. This assumption adds time for integration testing and interaction between team members. My project was small and didn’t require heavy interaction with team members.

During Phase 2 using a bottom-up approach to cost estimation I was able to predict the remaining time for Phase 3 would be 270 hours. This estimate was as close to the actual time of 273 hours that one could hope for. The bottom-up approach allowed me to measure my productivity up to that point in the project. My remaining SLOC estimate made during Phase 2 was very close to the actual and this allowed for my time estimate, which was based on my productivity and remaining SLOC, to be very accurate.

The following chart breaks down how much time was spent on each phase.
The following chart breaks down how much time was spent doing coding, design, documentation, and meetings for Phase 1.

The following chart breaks down how much time was spent doing coding, design, documentation, and meetings for Phase 2.
The following chart breaks down how much time was spent doing coding, design, documentation, and meetings for Phase 3.

5. Lessons Learnt
This project was a great learning experience for me. I got a taste of what it is like to work on a large project. One of the most valuable lessons I learned was to make your best effort to fully understand what you are trying to build. I found that at some points in the project I had to stop coding and think about design issues. This caused me some frustration as I like to focus on one part of the project as opposed to switching back and forth between designs and coding. I also got the chance to work in a
group project setting. Any chance I get to work in a group is beneficial since I will be working in a
group setting for the rest of my career. The most valuable lesson I will take away from this project is
how thinking through the design will make you think more deeply about what you are building. To be
more specific I thought my design was almost complete with just class diagrams. I thought I had a
good understanding of how the project would be implemented. But when I started creating a few
sequence diagrams I realized I needed to take some more time to think about how the objects would
interact.

6. Future Work
Not all the features described in the Vision Document were implemented in this project. The
remaining features will be implemented as future CIS 690 project or maybe a MS project. The
following documents what feature remain to be implemented.

6.1. Robot Builder (New Application/Mode)
There is a need for an application to build the actual robots the run in the simulation. This could
be a new application or a running mode of the Object Builder. This application would be very
similar to the Object Builder, but would need to be tailored to robot specific requirements. The
Robot Builder would require new primitive shapes. This could simply be accomplished with
wrapping a tag around the current primitives (box, cone, sphere, and cylinder). For example a
sensor could be a new primitive and implemented as just a standard box with a sensor tag
wrapped around its XML definition. The sensor primitive would also need additional attributes
which are specific to the properties of the sensor. Those attributes would also be included in the
XML definition.

6.2. Object Builder
- Moving objects as a group (a select feature).
- Making objects a hierarchy of objects instead of just a collection of primitives.
- Making the building surfaces scrollable to allow for bigger objects to be created.
- Adding the ability to rotate primitive shapes.
- Improving the search feature.
- Adding a zoom-in and zoom-out feature to the 2D building surfaces.
- Being able to set the bounding volume for the object being built (will help with collision
detection).

6.3. Terrain Builder
- Having the ability to add a texture surface to a region of the terrain (e.g., grassy, rocky, ect.)
- Being able to set the size of the terrain being built instead of having just a default
1000x1000 size.
- Improving search feature.

6.4. Environment Builder
- Making the building surface scrollable to allow for bigger models to be built.
- Allow multiple terrains to be placed on a model instead of the current limit to 1 terrain.
- Have a feature to dynamically place all objects on the top of the surface at their location.
- Being able to set the size of the model being built instead of having just a default
1000x1000 size.
- Adding a mouse over feature on the building surface to identify objects.
- Improving search feature.