Shadowed Horizons Acropolis Edition

27 DECEMBER 2004 Design Outlook

ABSTRACT:

This document will discuss the progress of the Shadowed Horizons project since it's switch to the Java programming language. It will also outline new goals as well as restate goals that have not yet been achieved. An attempt will be made to show a projected time line for the project Background information will be provided about the behind the scenes development process.

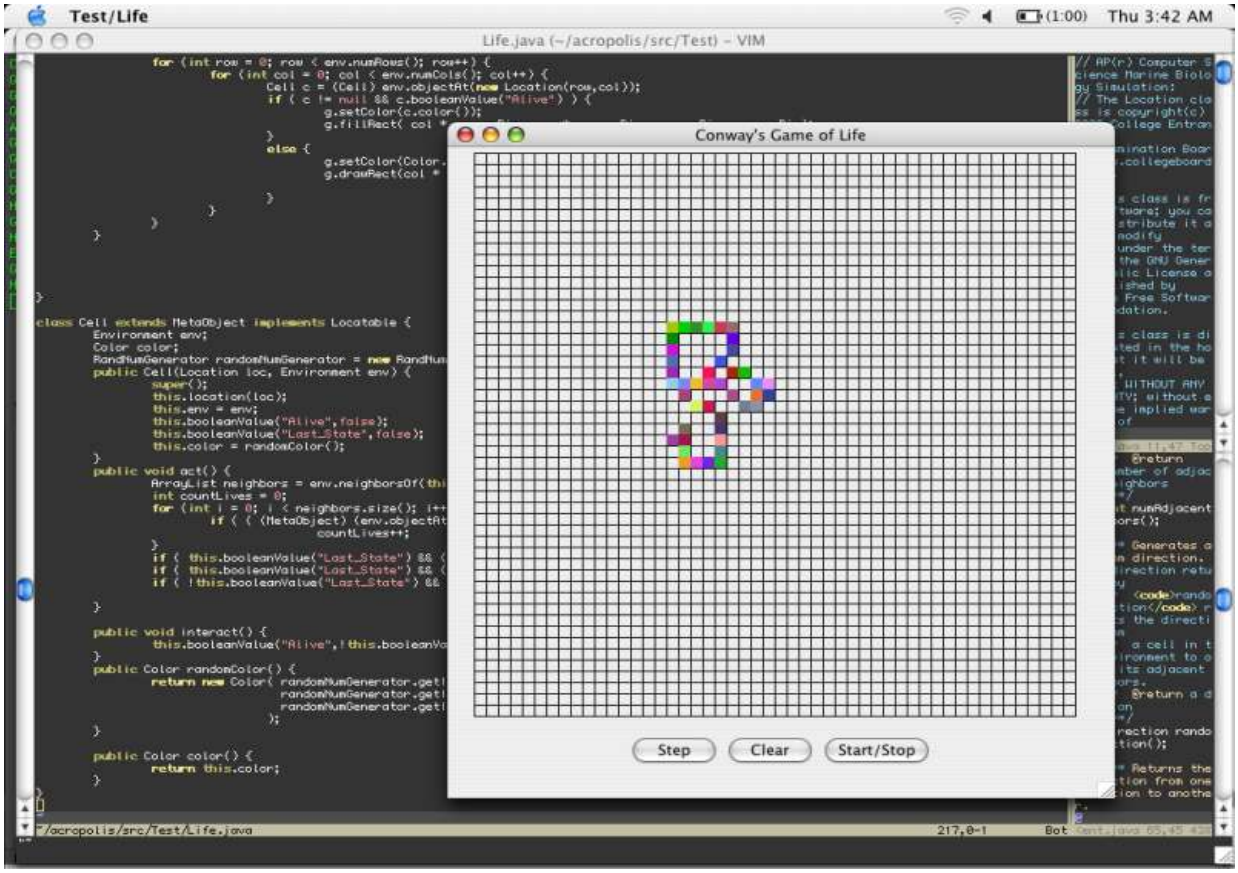


TABLE OF CONTENTS:

Foreword by Gregory Brown of smtose.org:

It seems like only yesterday that I started working on Shadowed Horizons. However, It's been a Java project for months, and it has been almost a year since it had it's perl beginnings. Needless to say, it has been an interesting ride. This month has marked the first public release of the Java code, though the CVS has been buzzing for months.

Documentation, accessibility of the code, and encouragement of community involvement have never been higher. We've created a wiki system, automated javadoc generation and cvs snapshots, and updated the project's website. It seems like we finally are on the path towards a professional product.

With the help of James Edward Gray II, and recently Mike Jackson, I have tried to make Shadowed Horizons more stylistically correct, more generalized, and as useful as possible. Though there are a large number of issues yet to be addressed, we have accomplished the base objectives outlined in our previous design outlook.

The more we accomplish, the bigger the dream becomes. This is a project that will always focus primarily on creativity. I really hope that some day, others can have just as much fun using my software as I had writing it. In the more immediate future, I hope there are people out there that will help my achieve my dream, while achieving their own dreams along the way.

However, not all of my goals are so idealistic. Functionality has become more and more of a priority. In a way, I'm learning to grow up and be mature about this project, which is something that is past due. If you've ever tried to write a game before, and got frustrated about how long it takes to get off the ground, you will know why I'm writing this software.

If you are motivated by the premise of this work, and think you can help me, or do it better than me in some place or another, let me know. I maintain that this project is about creativity first. You're welcome to join in, just contact me and say the word.

I hope you like what I've done with the project, but more importantly, I'd like to know what you want to see happen. So please, email me if you have any feedback you'd like to share. (GBrow1@newhaven.edu)

With no further ado, lets get down to the nitty-gritty of Shadowed Horizons.
As usual, if you want to skip the nostalgia, then don't read the history page.

Thank you for your support,
Greg Brown, smtose.org

History of smtose.org

Formerly The SMTOSE Organization and consisting of three members, Allen Turechek, Emily Laskin, and Gregory Brown, this development group had struggled for over a year to produce an adventure game, known as The Arachne Project. With conflicts in interest and a serious lack of time on the hands of some of the developers, the group all but collapsed in spring. The only developer who was working actively at the time was Greg and the project was redubbed Shadowed Horizons.

During this time, Greg became well aquainted with an experienced coder named James Edward Gray II. With James's guidance, much work was done to clean up the old Arachne code in hopes of creating a much more capable piece of gaming software.

Within months, Shadowed Horizons was getting more and more generalized. This led to the concept that it would be more suitable as a game software framework, rather than a game itself.

Though it was heading this way for quite some time, the project officially switched focus on October 30, 2004. This was roughly one year after the last Arachne release by The SMTOSE Organization (arachne-halloween). It switched from perl to Java, and also clearly defined its goal as a gaming SDK, rather than an application. The organization also switched its name to the less presumptuous smtose.org

On December 16th, 2004 smtose.org released the first Java version of Shadowed Horizons (acropolis). This release sent sH over the 100 download mark, while turning out to be the best single day performance in the history of the project in terms of page views and downloads.

This leads us to today, where the project still thrives, under active development. Though there will be a small break in development between 28 December 2004 and 15 January 2005 due to the fact that Greg Brown and James Edward Gray II will be working on an application for a Codefest Grant from RubyCentral, the project will be under active development consistently during the Spring of 2005. We are still greatly in need of assistance, and hopefully this document will help bring developers to our project. If nothing else, please tell us what we can do to improve this document and/or the project as a whole. You too can end up in this project's history if you choose to.

Project Overview

This outlook will be somewhat more terse than the previous design outlook. There will not be as much general discussion about the goals of the project, but rather a more pointed approach will be taken. This having been said, a more complete outlook that will combine both specific and general information will be completed in the near future. For now, we will address the goals of the original outlook and report on our progress. The out line that was presented in the original outlook will be addressed point for point, with the comments posted in italics. After this has been addressed, we will state the our new goals.

1. The MetaObject System
    a. A powerful and flexible attribute system
    Implemented via a series of HashMaps and accessor methods
    b. A capable inventory system with helper methods
    Implemented via a HashMap with a recursive search method
    c. A way for objects to act() on their own as well as interact() with others
    act() has been included as an abstract method in MetaObject
    An event system is still needed to make use of this.

2. The Environment System
    a. Linking the MetaObject System and JavaMBS's Environment implementation together
    This has been implemented but requires some additional work to be considered clean. Originally, a MetaObject was included as one of Environment's fields with accessor methods. However, now all Environments inherit MetaObject's methods directly. This makes the addition of a MetaObject as an internal object of the Environment obselete. However, the use of the properties() method, which relies on the original design is widespread, and changing it will be an interesting challenge.
    b. Providing any additional systems that may be needed to enhance the JavaMBS implementation
    The Location class has been extended to use x, y coordinates as well. This still needs some work. The Environment system has unfortunately been victim of a patch as you go style approach which leaves a number of fragmented improvements lacking an obvious organization. However, the addition of named points will prove to be useful.

Project Overview (cont.)

3. The World System
    a. An organized and manageable collection of Environments
    Implemented via an arbitrary depth nested container system
    b. The ability to add, remove, set, and provide copies of Environments
    Adding, providing copies and setting active environment have been implemented. Removing environments is under development, and needs to be cleaned before it is implemented.
    c. The ability to determine the Environment of a given object.
    Under development
    d. The ability to move objects from one Environment to the next
    Implemented via the Portal system which makes use of named points.

4. The Engine
    a. The ability to give time to each object in an Environment to perform it's intended action
    Most likely will use a threaded event system
    b. A system to apply and make appropriate changes defined by sets of Rules (such as Barter / Combat / Experience Systems)
    Most likely will be implemented via a scripting system or datafiles. JRuby is being considered, though Mike Jackson has been pushing Jython. Homegrown solutions are not out of question.

5. Quality Documentation
    a. Javadoc compatible documentation system-wide.
    Implemented and continuously updated via client-side Bash scripting
    b. Continued updates to project outlooks, specifications, etc.
    Not as often as we'd like, but currently being worked on.

In short, we've accomplished the majority of our goals in some way or another. However, the ways we have implemented these features is far from perfect. The biggest hole is currently in the Engine department, namely an event system. This will become a project of increased focus in the near future.

Before we go into the more immediate goals, lets address the key components of Shadowed Horizons as a whole.

1. The MetaObject System

Having some sort of generic but powerful object oriented system has worked for a number of programming languages and other development systems. This is why it is the key component to the Shadowed Horizons toolkit.

Continued focus will be placed on improving the functionality of this system with special focus on making it compatible with the event system when it is created.

2. The Environment System

This two dimensional grid based system needs to be truly generic. It also needs to be as powerful and easy to manage as possible. We need to fix up the JavaMBS code that doesn't fit in with our code. If we find our needs greatly exceed the capability of the JavaMBS system, we'll implement our own.

3. The World System

This is the part of the Shadowed Horizons toolkit currently on the bleeding edge. It needs to be generalized, so that it can represent an arbitrary hierarchy. It is on it's way to such a path, but needs to be able to read from data files and other sources to be truely worthwhile

4. The Engine

This will be the part that glues the toolkit together. We need to provide a nice Event system to make use of the powerful objects the toolkit provides. With this in place, the project will be at a functional level.

Specific Goals:

- Make the world system clean and easy to implement

- Provide the ability to create World objects from data files

- Create a world editing utility that allows visual creation of world hierarchy and structure.

- Implement an event system that works well with the provided Environment, World, and MetaObject systems.

- Clean up Environment so that it uses a single MetaObject for its system. Since it now inherits from MetaAdapter, we can eliminate the properties() method.

- Standardize code style throughout the project.

- Improve the world system to allow more in depth search of Environments and other functionalities.

*Though this list is short, the actual amount of tasks to do are great, and change all the time. If you want to know the latest, send me an email!

HELP WANTED!

The following positions are currently available.

MetaObject Component Developer:

Will design / assist with designing the MetaObject Component.
May focus on one of the key systems (Inventory, Attributes)
or take a general approach to the problem.

Environment Component Developer:

Will work on making the connection of the JavaMBS code with our code. Will develop additional helper methods to support any needs that the JavaMBS software does not fulfill.

World Component Developer:

Will focus on the functions necessary to provide a framework for a dynamic gaming world. Will be able to use Environment classes and be able to perform the necessary functions as outlined in this document.

Engine Component Developer:

Will focus on writing a VERY generic engine that will be able to take a number of sets of defined rules and apply them to the game to alter its operation. Will ensure that the engine is compatible with MetaObject action methods while still maintaining independence as a standalone component.

Other Developers:

GUIs, Networking, Stuff that goes bump in the night. Convince us it is relevant and you'll be a part of the team.

Tester / Quality Assurance Technician:

Break stuff. File bug reports. Tell us what's ugly. Tell us when your computer catches on fire because of something we did.

Technical Writer / Web Developer:

Help write documentation in JavaDoc format. Help write specifications and project outlooks. Keep the website and technical documents up to date. Run mailing lists, forums, etc.

Consultant / Trainer:

Get us up to speed on the technical necessities to complete this project. Offer suggestions for learning resources. Read source code, offer criticism. Help with design tips, and otherwise share experiences.

Evangelists

Spread the news about our software. Get people to use it. Get us more people for the above mentioned programs. Support us by soliciting the program any way you can.

Graphic Designers:

Make pretty things... get recognized. (Website / Logos / GUI stuff / ETC)

Legal:

Convince us to shut down our 'fabrica' in Peru. And more importantly, make sure we are doing our licensing right. Preferably a FSF representative and/or someone with experience in copyright law.

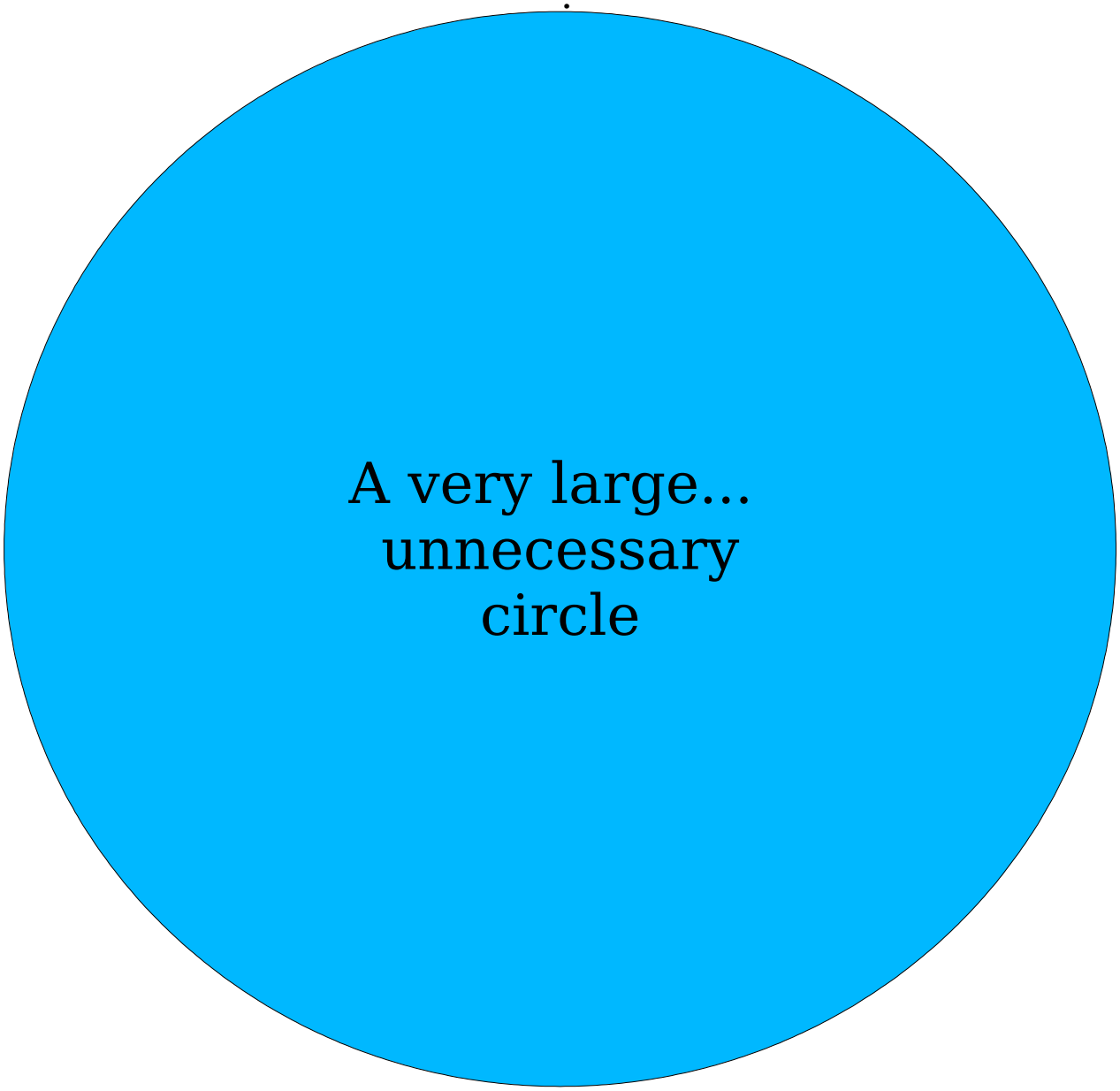
Collaboration Rules and Procedure

When you think of this project, think of two words...
“Freedom” and “Creativity”.

If you feel as if you can be of assistance while still upholding these two concepts, we applaud you and welcome your contribution.

The following pages contain the latest HACKING file, which explains our goals, our rules, and how to go about working on this project. This can always be found in distributions of our source, and on CVS.

.



Hacking Shadowed Horizons:

If you are interested in working on this project, this is the document for you.
We hope that you will enjoy the model in which we operate.
The following information is intended to guide you in the development process.

The most important thing you should know is that the number one goal of this project is to express the creativity of it's designers. What this means to you is that you will be given an incredible amount of freedom to persue your independant vision of your contribution, so long as it does not limit the freedom of existing contributors.

The second goal of this project is to be accomodating and inviting to those new to the free software scene. We are a newbie friendly project, and we don't believe any contribution is too small. We welcome the inexperienced to come learn along with us.

The third and final goal of this project is to create a piece of software that is highly customizable at the developer level as well as the user level. Though we intend to create a game with a robust and unique story (Arachne), we strive to ensure that anyone will be able to create their own story and theme to work with. The focus of this project is to provide a strong framework for games to be created on.

With these goals in mind, here are our simple rules.

0. Respect your fellow developers. The work done on this project is entirely a labor of love. We are more than open to criticism, but keep in mind that this is a program built for the fun of it.

1. Try to keep with our style of coding. We don't need obfuscated or otherwise crazy looking Java. We aren't particularly picky about style, but the more familiar your code appears, the more likely we are to accept and understand it.

2. Communicate! We will never be able to put together something great without sharing our thoughts. If nothing else, talk to me (Greg), and I will get the information out to the people who need to know.

3. Keep an open mind. There may be times where we throw out entire sections of code, resurrect from the dead a large codebase, or do other crazy things on a whim. Try to keep from having expectations of what things 'should' be like, and focus instead on making the vision you have become a reality. We won't get in your way, but don't be upset if our veiws or outlooks change from time to time.

If these rules don't scare you away, read on :)

There are many ways to help out. We need artists, musicians, creative writers, RPG Designers, Programming Trainers, Web Developers, and many other creative minds. Most of all though, we need coders. If you are not a coder, you can simply email us with your idea as to how you'd like to contribute, and we will take it into consideration.

Contact: GBrow1@newhaven.edu

If you ARE a coder...

The general procedure for making a contribution goes like this.

If it's your first time with us:
---
[0] Check out the PDF/OpenOffice file in Docs/
    This explains where we are heading with the project, and what you may want to help with.

[1] If you aren't looking at the CVS version of the software right now, get it.
    A lot of work goes on every day, and it's important that you are looking at
    the most recent code base if you plan to support us. If you need documentation
    on using CVS with sourceforge.net, head to:

    http://sourceforge.net/cvs/?group_id=99903

    The currently active cvs module is: acropolis

    Still stuck? Email me: GBrow1@newhaven.edu.
    If you have the CVS version of our project, continue to the next step.

[2] Feel free to tinker with the code you download. Please read through any documentation you
    find, take a look into our code archives for some of our old stuff, keep an eye out for
    comments in the code that may give you a hint of whats cool and whats not so cool. Once
    you feel aquinted with our work, you can submit a patch if you have a great idea, or
    preferably, email us and let us know what you plan on doing, and we'll let you know
    if anyone is working on it yet.

[3] Send us your contribution (GBrow1@newhaven.edu), and if we can use it, we will put it into
    the next release. At this point, we may ask you to become a developer for us if you are
    interested.

If you are a developer for us:

[0] Grab the latest developer CVS version, including any new files (cvs update -dP)

[1] If I (greg) know what your doing, code away. If not, please be sure to contact me or
    whoever is working on the code you have modified as soon as possible.

[2] Be sure to test your code before you commit. DO NOT submit code that breaks the program
 You CAN submit code that does not compile, but you must comment out anything that causes
 other components to fail to compile, crash or otherwise act squirrely.

[3] Commit your code with an interesting and informative message.

[4] Be prepared to be criticised and/or praised for your work. Be ready to answer questions as they
    arise. This is nothing to be afraid of, and is nothing more than peer review.

[5] Rinse and Repeat as desired.

I hope that this document has been helpful. Any questions?
                                                GBrow1@newhaven.edu

Contact Information

Currently, all of the issues in the project that involves direct communication go through Gregory Brown. This will change if other individuals take control of sections of the project However, to encourage community involvement, a wiki system has been set up at www.smtose.org

This is the best place to express your opinion in the general sense.

For more specific information, please contact Greg though.

Email: GBrow1@newhaven.edu
AIM: pollymorefism

Projected Timeline

Here is the incredibly subjective time-line of our project:

> Spring 2005

All basic systems will be implemented. Simple games will be created as proofs of concept, the first genre being a tile based RPG dubbed SimpleRPG.

>Summer 2005

Basic systems will be at the functional level. Shadowed Horizons will become useful as a development toolkit. A number of programs will be created to aid novice or lazy developers in world creation and other tasks. A robust set of adventure game libraries should be taking shape

>Winter 2005
Work on Adventure game libraries should be near completion
Other genres will be worked on at this point in time

>2008?
A gaming IDE. Robust libraries for a number of gaming genres, with all generalized functions seperate from the more specific libraries. A number of utilities will make game development a breeze (we hope).

This is only a guess, and will probably be updated many times

Acknowledgments:

Though Greg has been the only active developer on this project for quite some time, there are a number of people who have supported smtose.org and Shadowed Horizons and deserve special consideration:

James Edward Gray II:
The “Invisible Hand” and number one consultant and tester. James wrote the search system for the MetaObject, and constantly complained about places where the code has been poorly designed, or has clashed with Java convention. He's continuing to do this, and it is something this project greatly benefits from.

Emily Laskin:
Linux tester when Greg is too lazy to fire up his Gentoo box. Hopefully, she will be helping with the project again soon. She often plays around with the test cases and says, “Thats cool”, or “This doesn't seem right” about one thing or another, which is very helpful. She also lets Greg go through the code with her and explain things, which is good for both of them, and good for the project.

Mike Jackson:
In a small amount of time, Mike provided a lot of hand slaps to Greg regarding Java style conventions along with a bunch of “imagine if's” and assistance with helping to find the still cloudy big picture. Mike contributed code, documentation, and ideas, and this was well appreciated.

Jacalyn Diesenhouse and the
University of New Haven Computer Science Department:

Professor Diesenhouse gave Greg plenty of time to work on Shadowed Horizons, provided opportunities to apply the knowledge gained in class to the project, and provided books and materials to help with continued research. This support was instrumental in order to allow continued development of Shadowed Horizons during the Fall Semester0

To anyone who has helped improve this project, Thanks!