

The Catacombs: A study on the usability of games to teach introductory programming

Tiffany Ralph

Department of Computer Science
Colorado State University

Tiffany Barnes

Department of Computer Science
University of North Carolina at Charlotte

Abstract

This summer I developed an in-depth study on the effectiveness of using games to teach students introductory Computer Science concepts in coordination with the entire Games2Learn Summer Group. This summer study will aid in the creation of a massively multiplayer online role playing game (MMORPG), StormHaven. StormHaven is intended to act as a significant teaching tool in the University of North Carolina (UNC) at Charlotte's Computer Science Department, replacing traditional out-of-class assignments. The study is meant to look into the various techniques that can be used in StormHaven to teach students programming concepts, including interface design and plot development, as well as whether or not StormHaven would be a significant improvement over traditional teaching resources. The information from this study will serve to better support the development of StormHaven and enable development creation of the game, hopefully spurring further expansion of the Games2Learn project. My research focused on the creation of different interfaces and determining which type students prefer and learn best from, by creating a rapid prototype level of StormHaven for use in the study, and to serve as an example of where we wish the game to go to.

Introduction

The work presented in this paper is intended to help provide support for the Game2Learn project, and to establish groundwork evidence that computer science students would enjoy learning from an MMORPG, as well as evidence that shows the game's pedagogical effectiveness.

In this paper, I will discuss briefly the concepts behind the creation of StormHaven, the motivation for creating the game, and the development of the game over the past year. I will then go on to discuss, in depth, the study conducted in the summer of 2006, including the process of development and our current results. The overall results from the study will go

towards the further development of the Games2Learn project and eventually, the expansion of the StormHaven concept into a full-scale game and learning tool.

Background

The lecture-style teaching of the 20th century, while still largely used in classrooms today, is quickly becoming an outdated model of teaching with the emergence of computers into the pedagogical setting of the classroom (Gifford & Enyedy, 1999; Steinkuehler, 2004). Computers exist in almost all classrooms in our society today, acting as surrogate teachers to thousands of children. The basic skills of using a computer are being developed at younger and younger ages, making almost all students at any grade level capable of completing tasks the prior generation was unable to do until the high school level (Prensky, 2003; Steinkuehler, 2004). The computer provides its user with means of communication, entertainment, and learning, often times offering a combination of all three. Programs like *Math Blaster* that use games to teach children math concepts have been around for years. Learning through games is not a new theory by any means, though the idea of using an MMORPG is. The majority of educational games usually consist of “drill-and-kill software,” where students are presented with questions or tasks and then rewarded for correct answers with classic arcade games (Soloway, Guzdial, & Hay, 1994). The need to step away from this type of game in educational settings is evident in the fact that the majority of games selling on the market now are more complex, multiplayer strategy games, such as *World of Warcraft*, *Lineage II*, and *Guild Wars*.

Computer and video games are obviously being played by a large proportion of students today. College students account for a large number of those playing these games and the motivation for playing them lies in the fact that they are fun. Students enjoy doing things that are fun, hence the definition of the word, and have been known to work harder, longer and with more interest in something they find engaging (Becker, 2001). It's time that we grasp the power of motivation supplied by these games for education and apply it to college-level learning, allowing students to strive for higher education through the devices that so often distract them from it.

The difficult part of developing a game for use in education lays in the transition of existing subject matter into game-friendly material. The use of lectures as a teaching tool has existed for an extensive period of time, making it easy to recreate and apply to new material. Rote learning has also been used for decades, drilling numbers and facts into students until they're able to regurgitate the information at will (Norman & Spohrer, 1996). The practice of using lectures as a

primary teaching tool derives from the traditional viewpoint that learning is the transfer of knowledge to others by simply conveying the information to them either verbally or textually (Gifford & Enyedy, 1999; Edelson, 1999). In the end of the 20th century, the perception of learning shifted from a more teacher-focused process to one where focus was based on the student, and their individual development. This theory of learning, or Learner Center Design (LCD), is based on the assumption that human intelligence is the ability to interpret and translate a physical system into an understandable mental representation (Gifford & Enyedy).

Edelson (1999) took the theory behind LCD a step further to develop a theory called Learning-for-Use (LoU), which focuses on giving students conceptual understanding of material that will allow them to apply the knowledge gained when it is relevant. LoU is based on principles similar to those of LCD, teaching students through “the construction and modification of knowledge structures” (Edelson, 1999), but also develops the knowledge gained in a manner as to facilitate its application. The effective development of new knowledge for learners begins with the introduction of the material to be learned in a way that relates to the learners prior knowledge, building on it. Scaffolding reorganizes the information to be gained into a form that supports the use of the knowledge in later, more advanced situations through elaboration and integration (Jackson, Soloway & Krajcik, 1998; Edelson, 1999; Quintana, Eng & Soloway, 1999; Rosson & Carroll, 1996). The use of computers to teach students allows for adaptive (automatic based on student understanding) and adaptable (manual advancement applied by user) scaffolding, providing an intelligent tutoring system (Jackson, Krajcik & Soloway, 1998). The application of these theories has led to the development of iconic programming tools that can be adjusted and aids in the growth of skill-level from novice to expert, in teaching CS1 (Cillier, Calitz & Greyling, 2005) and consumer-friendly programming software, such as Dreamweaver.

A popular example of scaffolding is games. Most new games start out at a level users find simple, letting players get adjusted to the game’s controls and allow for introduction of the plot. The best example of a game type that adjusts difficulty to its players is MMORPGs. In these games players start out building their character and developing relationships with other players. In most MMORPGs players can then start to complete quests that increase in difficulty as the players gain more levels for their characters. When players reach levels that make the computer generated characters too easy to play against, several MMORPGs allow for player vs. player, where users can fight or play against other users that may be at even higher levels than themselves. The worlds created by these games are almost as well established as our own.

Players can interact with other players, they can get married, have children, setup shops, trade and barter, almost anything you can imagine. The popularity of these games is apparent when you consider that due to out-of-game trading of in-game items, the MMORPG *Everquest* has the real world's seventy-seventh largest economy, having a GNP per capita between Russia and Bulgaria (Steinkuehler, 2004). Using these games for teaching is a nascent topic that is currently being researched at several universities due to the nature of these games and the power of using them as motivation for learning (Jackson, Krajcik & Soloway, 1998; Quintana, Eng & Soloway, 1999; Gumhold & Weber, 2005). MMORPGs require cognitive thinking, problem-solving, reading, mathematics, collaboration with other people, everything that goes with participating in a society effectively (Steinkuehler, 2004), but are still fun and interesting to play. MMORPGs also allow users to create custom representations of themselves through clothing, class, and even race, creating a diverse game population that lets players be themselves (Soloway, Guzdial, & Hay, 1994). The nature of large-scale MMORPGs as being able to support diverse populations while still maintaining player individuality is a strong pushing force behind the implementation of StormHaven as an MMORPG. The motivation provided by using StormHaven in connection with lectures will hopefully raise retention and enrollment at UNCCharlotte's Computer Science program.

Methods

The study conducted this summer consisted of developing two rapid prototype versions of StormHaven. One of the versions was developed using RPG Maker XP and provides students with a more exploratory gaming experience than the other, which was created using the BioWare Aurora Neverwinter Nights Toolset and has the user follow linear stages of game play. The development of the games was divided among the six students working on the project this summer, three assigned to each game. I belonged to the group creating the Neverwinter Nights version, titled The Catacombs, which aimed to create two interactive quests that students will be randomly selected to play through. The two quests share a similar plot, rescuing a family from the village catacombs, but use different game interfaces to teach students concepts. One interface is based in conversations with the player's unruly spell book to learn programming semantics, while the other uses interaction with items to complete scripts that do various tasks and develop more syntax-based knowledge. Twenty students will be asked to participate in the study, all of which should have taken or plan to be taking an introductory Computer Science course. The students will be given a pre-test and post-test to see how much the game actually helps them learn, along with an interview after playing to see if they would have preferred to use

something like the game to learn introductory Computer Science concepts. The student's onscreen actions will be recorded with a video camera focused on the screen, positioned so as to not get the student in the shot. A logging system is also to be used with the Catacombs game, where all student interactions will be recorded to the game's log with time stamps so that we can determine exactly when students completed specific tasks.

Due to various events out of our control, the majority of the game's development fell to my shoulders. The plot was developed by all of the members of my group, as well as the code and concepts that should be taught to the students. The actual development of the two versions of the game was mostly done by myself, give or take a few scripts and dialogues.

The overarching plot of both versions of the game is that the player starts in a small village, the only other person there being a woman. The player finds out that the woman is a mother, her children have gone down into the Catacombs and that her husband has gone down to find them. It is getting late and she has grown concerned for their safety, asking the player to go and find them. Depending on the version that the student is playing, they either need to find the correct magical scroll to get in the door (Fig. 1), or answer the questions of their rather sarcastic spell book to unlock it (Fig. 2). Once the player gets into the catacombs, they are approached by Talino, the woman's husband, and find out that he has been injured and cannot go on. He begs the player to go on and find his children, giving the student what little gold he has, along with gemstones he's found in the caverns. The player continues into the catacombs, finding treasure and fighting monsters until they come across a large pit which a little girl is on the other side of. In the Konijn version of the game, the player uses the gemstones they received from the father to unlock the power of an ancient rune stone (Fig. 3) to build a bridge and get across to her. In the Grimore version, the player consults with their spell book once more to cast a spell that



Figure 1: Looking at the scrolls



Figure 2: Grimore the Sarcastic Spell Book

Figure 3: Filling in the Rune Stone.



builds the bridge to save the girl. Both versions use nested for loops to build the bridge and save the girl, making sure the student is prepared for the final task with the little boy, which uses nested for and if loops. The little girl leads the player to her brother and once more the student either constructs a spell with Grimore or a rune stone, depending on the game version.

The goal of the two games is to introduce students to two different interfaces while keeping the concepts taught the same. The main concepts that are meant to be taught are conditionals, variable declaration, loops, and nesting. Scaffolding is used to introduce the students to the more difficult concepts, along with the more difficult interfaces in the game. The first task that the students are asked to complete is rather easy, either choosing a scroll or answering three simple questions on if/then loops and initializing variables. The second task in the game, with the bridge, uses either a fill-in-the-blank interface or more difficult questions, Grimore now hurting the player for wrong answers. The final task uses similar interfaces as the second, though now the player is given more gemstones than is required for the rune stone and Grimore

asks even more difficult questions. All of the scripts developed by the students are stored in their journal and they are expected to use these scripts to aid them with the more difficult tasks.

Development of The Catacombs was done in the BioWare Aurora Neverwinter Nights Toolset. While the toolset offers several wizards for mapping, character building, item interaction, and even scripting, it does not offer the level of assistance needed to create The Catacombs and the more difficult interfaces used in the game. The Konijn version of the game uses several custom scripts that bend the initial purpose of some of the game's built in functions and items to create the rune stones and journal entries that go along with them.

The Grimore version of the game was initially written with a series of dialogue trees where the trapped family members asked the player how to code certain things, choosing the correct lines of code from multiple choices. This technique focused more on syntax errors than semantics and often times students would ignore the code and just guess the correct answer. Grimore was introduced in the hopes that students would actually read the questions presented, along with all of the options, and select the correct answer. The idea of having the spell book ask the player the questions rather than the trapped family members also made more sense with the plot idea that the family was stranded. It also made sense that Grimore would hurt the player for getting incorrect answers, whereas stranded children probably would not.

The built in journal function plays a key role in both quests, acting as a real-time script display and storage device for the student's scripts. As the student answers questions and puts in gemstones, their journal updates to show them the code that they've created so far. This feature helps students get a better grasp on what Grimore is asking, as well as show them what part of the script they need to fill on next with the rune stones. If in later tasks the students get confused or don't understand a concept, they can look back on code they've already written for assistance.

Results & Analysis

Time constraints disallowed the running of the study with the number of students initially desired while I was still there for it, but studies will continue throughout the next semester and hopefully until enough data is received.

All of the students that have played the Grimore version of the game so far have completed the game in the 20 minutes allowed. Students that have played the Konijn version of the game played the entire 20 minutes, being asked to stop by the proctor before they can complete the quests. The first quest where the student is asked to select the correct scroll took one player as long as 10 minutes to complete, after he died 7 times and asked for help from the proctor. I believe the Konijn version of the game is so difficult for students because it uses more complex game controls than the Grimore version, and the tasks focus more on correcting syntax error than semantics. In general, subjects have spent more time exploring the caverns, interacting with computer players, and learning the game's controls, which I believe accounts for some of the time constraint problems.

Subjects tested so far on either quest have stated in interviews that they “could see using the game as a re-enforcer for a class, something to do before a test.” One subject in particular who had difficulty completing the quest in the allotted time responded to the question of whether or not he thought the game could be used to teach introductory programming by saying “Yeah! I mean, it would be awesome if like, after a lecture, the professor just said ‘Alright, get to level 43 this weekend.’ [*He laughs.*] I would have definitely wanted to be in that class. You guys should develop this into a game that’s like that.”

Students’ performances on the pre and post test are currently unavailable to me; however, I hypothesize that the number of correct answers on the post test will be higher than on the pre, as students seem to be able to understand the more difficult quests. And respond more quickly as they advance through the level.

Conclusion

Based on student responses so far of the game, I can conclude that students would respond positively to the use of an MMORPG in collaboration with their Computer Science courses, as well as the fact that students seem to prefer doing semantic based-tasks and using less-control centered interfaces. When users are presented with too many steps to complete a task, they become confused and begin to repeat or miss steps, even if the motivation for the task is strong. When they become too confused or frustrated they stop trying and start guessing, completely destroying the purpose of the game and the quests. Overall, though, students responses suggest that the implementation of StormHaven in conjunction with class lectures at UNCC will

be successful at achieving our goals of higher retention and enrollment in the Computer Science program.

Future Work

Obviously further development of the game and the eventual realization of StormHaven is the desired next step in my work; however, distance limits my ability to continue work with the Games2Learn group. I will continue my studies on usability and the development of effective interfaces, though my focus will change from games to web sites. Hopefully I will still be able to help somewhat with the game's creation, though I'm sure that most of the development will take place without me.

References

- Becker, K. (2001) Teaching with games: The Minesweeper and Asteroids experience, *The Journal of Computing in Small Colleges*, Vol. 17, No. 2, 2001, 22-32
- Barnes, T., & Wick, M. Games2Learn: Teaching CS1 in a game-immersed, gender-neutral environment, Unpublished, U. of North Carolina at Charlotte
- Edelson, D. C. (2001). Learning-For-Use: A framework for the design of technology-supported inquiry activities, *Journal of Research in Science Teaching*, Vol. 38, No. 3, 355-385
- Gifford, B. R., & Enyedy, N. D. (1999) Activity Centered Design: Towards a Theoretical Framework for CSCL. In *Proceedings of CSCL1999*, (Palo Alto, CA, 1999), pp. 189-196
- Gumhold, M., & Weber, M. Motivating CS students with game programming. In *Proceedings of the 6th International Conference on New Educational Environments (ICNEE)*, (Neuchatel, Switzerland, 2004)
- Jackson, S. L., Krajcik, J., & Soloway, E. (1998) The design of guided learner-adaptable scaffolding in interactive learning environments. *Conference on human Factors in Computing Systems*, ACM Press/Addison-Wesley Publishing Co., Los Angeles, CA.
- Mayer, R. E. (2003). The promise of multimedia learning: using the same instructional design methods across different media. *Learning and Instruction*, Vol. 13, 125-139.
- Norman, D. A., & J. C. Spohrer (1996) Learner-centered Education. *Communications of the ACM*, Vol. 39, No. 4, April, 24-27
- Prensky, M. (2003) Digital Game-Based Learning, *ACM Computers in Entertainment*, Vol. 1, No. 1, October, 1-4
- Rosson, M. B., & Carroll, J. M. (1996) Scaffolded examples for learning object-oriented design, *Communications of the ACM*, Vol. 39, No. 4, April, 46-47

- Soloway, E., Guzdial, M., & Hay, K. E. (1994) Learner-centered design: the challenge for HCI in the 21st century. *Interactions*, Vol.1 No.2, 36-48
- Steinkuehler, C. (2004) Learning in massively multi-player online games, In Proceedings of the sixth international conference on learning sciences, (Mahweh, NJ, 2004) pp. 521-528
- Quintana, C., Eng, E., Carra, K., Wu, H., Soloway, E. (1999) Symphony: a case study in extending learner-centered design through process space analysis, Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit, (Pittsburgh, PA, 1999), pp.473-480
- Wallace, R., Soloway, E., Krajcik, J., Bos, N., Hoffman, J., Hynter, H. E., Kiskis, D., Klann, E., Peters, G., Richardson, D., & Ronen, O. (1998) ARTRMIS: Learner-Centered Design of an information seeking environment for K-12 education. *Human Factors in Computing Systems: CHI 98*, ACM Press, 195-202

Appendix A

This document explains the creation process and coding levels contained within the Neverwinter Nights (NWN) modules created for the 2006 Summer Study with the BioWare Aurora NWN Toolset.

1. **Module 1: Grimore: The Sarcastic Spell Book**

- a. This version of the module uses dialogue trees to walk users through rescuing the Fryar family from the Catacombs. The player's spell book reveals spells to the wizard as he answers semantic questions, allowing the wizard to cast the spells to assist and rescue members of the family.
- b. Plot development:
 - i. The player starts out by meeting the mother in the town square to learn that her children have gone down into the Catacombs and that her husband has gone down to find them. It is getting late and she has grown concerned for their safety, asking the player to go and find them.
 - ii. The player goes to the entrance of the Catacombs to find it locked. When he tries to read his spell book, it refuses to give him the spell to unlock the door without first answering questions. The book eventually reveals the spell and the door opens so the player may go down into the Catacombs.
 - iii. The player is approached by the father who tells him that he has been exhausted by the monsters in the first room and cannot go on. He gives

the player exp and some treasure and then asks him to go farther in to find his children.

- iv. The player encounters creatures in the first large room that they must destroy before they can move on, either by using magic skills or their weapon.
 - v. The player goes farther into the Catacombs to find the first child, a young girl, standing on the other side of a large pit. Once again, the book won't reveal a spell to the player without first asking questions on the spell's semantics. The player answers the questions, casting the spell, creating the bridge.
 - vi. After the bridge is created, the player crosses the bridge and it disappears once they get across it. The player then cannot go on without first talking to the girl. She tells the player to follow her and goes on to find her brother in another room, by a locked door.
 - vii. The brother tells the player that he cannot open the door, but thinks that the rune stone beside him might have something to do with it. The player speaks with Grimore once more and reveals a spell that will fill in the missing inscriptions from the rune stone. When the player finishes, the rune stones create a key and when the player opens the door the brother and sister leave.
 - viii. The player returns to the village to find the family reunited, where he/she is rewarded and thanked.
- c. Game development: (the numerals correspond with the plot numerals above)
- i. After talking to the wife, the local variable **overall** changes to 1. If the player attempts to talk to the wife a second time, she gives another conversation, based on the fact that the **overall** variable is set to 1. Her conversation file is **queststart**.
 - ii. Opening the door changes the variable **overall** to 1, making it possible for the player to talk to Talino, the father, even if the player hasn't talked to the wife. The door conversation is **opendoor2** and calls the script **opendoor** when completed to actually open the door.
 - iii. Talino's (whose tag is **Talino**) first conversation is called with **overall == 1** and at the end of it, the following scripts are called:
 1. **mangem** → rewards the user 30xp and 3 gemstones

2. **moveawaytalino** →
 - a. sets **overall** to 2
 - b. moves Talino to waypoint **npcgoto1**, that is located in the entrance way
 - c. makes Talino disappear and go to waypoint **npcgoto** in **Watershed Downs** after 4 seconds, which is where he is when the player returns to that level
- iv. The encounter is created by a trigger **X2_UNDERDARK002** and summons 2-3 very weak monsters for the player to kill.
- v. There is a pile of rubble up on top of the hallway that the trigger **grimore2** uses for the conversation **grimore2** to create the bridge. The conversation calls the following scripts at the end:
 1. **overallset003** → sets the variable **overall** to 3
 2. **bridgebuilt** → sends player to the waypoint **bridge** on the other Catacombs map (tag **catacombs2**) and uses the visual effect for transportation
- vi. This plot element contains two conversation scripts
 1. When the player is perceived by the girl, the script **sc_girlbridge** is used that starts the conversation thread in **bridgebuilt**. The conversation starts the script **byebyebridge** that area transitions the player back to the first catacombs map, on the other side of the pit.
 2. Second conversation involved the little girl:
 - a. is activated when **overall** == 3 (checked by **overallc003**)
 - b. activates script **follo** that
 - i. sets **overall** to 4
 - ii. sends the girl to the waypoint **girlgo1**
 - c. if the player tries to interrupt the girl while she goes to her brother, she uses the variable check of **overall** to tell them to let her go and then continues on her way.
- vii. The brother, Perarry (tagged **Perarry**), uses the conversation **littleboy** and starts talking when variable **overall** is set to 4 (checked by **overallc004**). The player then speaks to the stone, which triggers a

conversation with another hidden rubble called **grimore3** (both the conversation and the rune). A series of scripts is used.

1. **grimore3** → visual effect on the runestone (**rune**) and creates key **finallock** for **FenceDoor**
2. **overallset005** → creates sets **overall** to 7 (I know, it's terrible)
Opening the door causes the script **kiddies** to run, that sends **girl** to waypoint **girlgo** and **Perarry** to waypoint **boygo**, both of which are in **Watershed Downs** (this last bit makes the key irrelevant, but think the presence of the key helps the reality of the plot a bit)
- viii. The player leaves the Catacombs and finds the family either gathered, or gathering, around the mother. Talino approaches and thanks the player again, giving him/her some more experience. All of the NPCs run lines in their last conversation file under the condition that **overall == 7**. The little boy tells the player to tell the test proctor that they've finished the game.
- d. The scripts created in this version of the module were created by either the built in script wizard, [Lilac Soul's NWN Script Generator](#), Paul Hospers (thanatos_ayanami@yahoo.com), or myself.
- e. While the general formatting of item names is a bit random, scripts dealing with the variable **overall** are named according to a very simple system (minus a few exceptions).
 - i. When setting/changing the variable, the script name is **overallset00_** where the blank is the number of the variable.
 - ii. When checking the variable, the script name is **overallc00_** where the blank is the number of the variable you wish to check (if checking to see if **overall == 3**, the script would be named **overallc003**).
- f. Conversation files used in this module's predecessor were created in collaboration with Alex Godwin and Amanda Chaffin.

2. Module 2: The Konijn Gemstones

- a. This version of the module uses the power of old magical items throughout the Catacombs to rescue the Konijn family. The player must interact with rune stones and other various old magical items to restore the old magic and save the family.
- b. Plot development:

- i. The player starts out by meeting the mother in the town square to learn that her children have gone down into the Catacombs and that her husband has gone down to find them. It is getting late and she has grown concerned for their safety, asking the player to go and find them (same as in Grimore).
- ii. The mother informs the player that the door to the Catacombs cannot be opened without first casting a specific unlocking spell which can be found around the village. There are four scripts around the village and the player must look at them and determine which one is the correct one to open the door. If they attempt to use more than one at once or the wrong one, they are killed instantly.
- iii. Once the player enters the Catacombs, they are approached by Talino, the father, who tells them that he was overwhelmed by monsters and after defeating them, could not go on. He also gives the player 9 gemstones that he found in chests throughout the room, all of which have 'strange' inscriptions on them.
- iv. The player continues through the Catacombs and encounters monsters in the large room and must defeat them before they can go on, either with magic or their weapon.
- v. As the player approaches a large pit, they hear a voice from across it, which turns out to be the little girl. She explains to the player that her and her brother crossed the pit by putting gemstones into the rune stone near the pit, using a scroll they found on the ground as a guide. The player is instructed to use the scroll and fill in the missing bits of the rune stone with the gems that Talino gave them. If they have any problems doing it, they can just walk closer to the pit and the little girl will try and explain it again.
- vi. Once the player fills in the rune stone, a spell is cast and the bridge is formed so that the player can get across. The corridor behind the player is now caved in, blocking their way back. When the player crosses the bridge and the little girl sees him, she runs forward and tells him that the bridge is disappearing. They are then moved to the other side of the pit and the bridge is gone.

- vii. The little girl tells the player that her brother is farther in and has found a door that he is trying to open. She asks the player to follow her and then leads him to her brother, no matter what the player answers. She also gives the player the gemstones that she has found throughout the caves.
 - viii. When the player talks to the little brother, he finds out that door is sealed and uses a rune stone like the bridge to unseal. The boy gives the player his gems, along with a scroll that shows the translation and gaps in the stone.
 - ix. The player fills in the rune stone, like the one that cast the bridge and once complete, the rune stone casts the spell to unlock the door.
 - x. When the door opens, the children run through it and out to find their parents in the Village.
 - xi. After the player returns to the village, he can talk to the family and receive thanks and money for his effort and time. The little boy also tells the player that he/she has finished the task and should tell the proctor (this is for the summer study and can be changed for normal game play).
- c. Game Development (numerals correspond to those above):
- i. The conversation with the mother, tagged **Dorani**, uses the check for **overallchk000** for the first thread of conversation. Once the player has received the instructions, the variable **overall** is set to 1 by **overallset001**. If the player tries to talk to her again before completing the quest, it uses the thread with the check **overallchk001**.
 - ii. The door uses a trigger to check the scrolls that the player has, tagged **X2_EPFIRETRAP001** that calls the script **_catdoor_opener** when entered by player. The script runs through an if/else if/else check to see whether or not the player has a scroll tagged **killerdoor**. If the player has this scroll, they are killed instantly. They need the scroll tagged **dooropen**.
 - iii. Talino, tagged **Talino**, approaches the player, running the OnPerception script **startconv**, that starts the conversation **rescuednpc**. There is no variable check on the first thread of conversation, just in case the player did not talk to Dorani. At the end of the conversation **moveawaytalino** is

run, giving the player the gemstones needed for the bridge rune stone.

The gemstones are tagged and resrefed¹ as follows, in order of insertion:

1. **gem1/classunlockdo002**
 2. **gem2/function003**
 3. **gem3/function002**
 4. **gem4/classunlockdo003**
 5. **gem5/function005**
 6. **gem6/function004**
 7. **gem7/classunlockdo006**
 8. **gem8/001**
 9. **gem9/classunlockdo005**
- iv. The encounter area is tagged **X2_GNOLL001** and spawns two Gnoll shaman.
- v. Two generic triggers, both tagged **GenericTrigger**, surround the rune stone and when the player enters them, it runs the script **trigger**, which starts the conversation **bridgebuilder**. The thread that checks to see if **overall** is equal to three by running **overallchk003**. The end of that thread sets **overall** to 5 and **hasWrittenSpell** to 1. If the player enters the trigger again, the thread that checks **overall** as being 5 runs, offering the player more assistance. The rune translation in front of the stone is tagged **RuneTranslation** and the rune stone is tagged **x0_RuneStone1**. When the player clicks on the stone, it opens the stone's inventory. When the inventory is closed, the script **gemtest** is run that sees if the player has put in the right stone. Also, within the inventory, there exists the markers for which number in the spell the player is working on. The numbers are custom images and are imported to the game through the hak pack called **stormhaven2**. The items are controlled by **gemtest** as well and are not removable or moveable in the stone's inventory, controlled through the item properties. Each number is an item and has a unique tag and resref, listed below (the resref is one number above the actual number that the item is).
1. **001/002**
 2. **002/003**

¹ Blueprint ResRefs are used with certain functions instead of tags and cannot be edited after item creation.

3. **003/004**
 4. **004/005**
 5. **005/006**
 6. **006/007**
 7. **007/008**
 8. **008/009**
 9. **009/010**
- vi. When last gem is set into the stone, **gemtest** jumps the player to the waypoint **IBridge** on the area map tagged **TheCatacombs** and with resref **thecatacombs002**. The girl in this map is tagged **girlbad** and OnPerception runs the script **sc_girlbridge**, which starts the conversation **bridgebuilder** and sets the **overall** variable to 6. The thread that runs checks with **sc_006** and then runs the script **byebyebridge** to teleport the player to the waypoint **bridgebuilt** and sets **overall** to 7. The waypoint is on the original catacombs map where the build isn't built and the girl is already placed there, tagged **girl**.
 - vii. The player talks with the girl and runs the thread triggered by **overallchk007** and then the conversation runs the script **givegem1** to give the player the girl's gems for the next quest and then the script **follo**, which has the girl run to the waypoint **girlgo1**. It also updates the following variables to the following values: **overall** = 8, **hasWrittenSpell** = 4, **hasGem** = 2.
 - viii. The little boy is tagged **boy** and runs the conversation **littleboy**. The thread runs that is started by meeting the conditional in **overallchk008**. The boy gives gems and updates the journal with **givegem2** and sets the variables with **at_003**.
 - ix. The script that runs the rune stone tagged **x0_RuneStone2** is called **gemtestboy** and runs pretty much exactly like the first one, using the same items for the numbers and everything. The spell opens the door at the end of the level, tagged **FenceDoor**, and sets **overall** to 10.
 - x. The door opening executes the script **kiddies** which causes the children to run to waypoints in the village, tagged **girlgo** and **boygo**.

- xi. The family gives various rewards if the player talks to them again, all in their original conversations with threads that are triggered under the script **overallchk007_gr**.
- d. The scripts created in this version of the module were created by either the built in script wizard, [Lilac Soul's NWN Script Generator](#), Paul Hospers (thanatos_ayanami@yahoo.com), Amanda Chaffin, or myself.
- e. All conversation files written in this module are modified versions of those in the Grimore version and were written all by me.