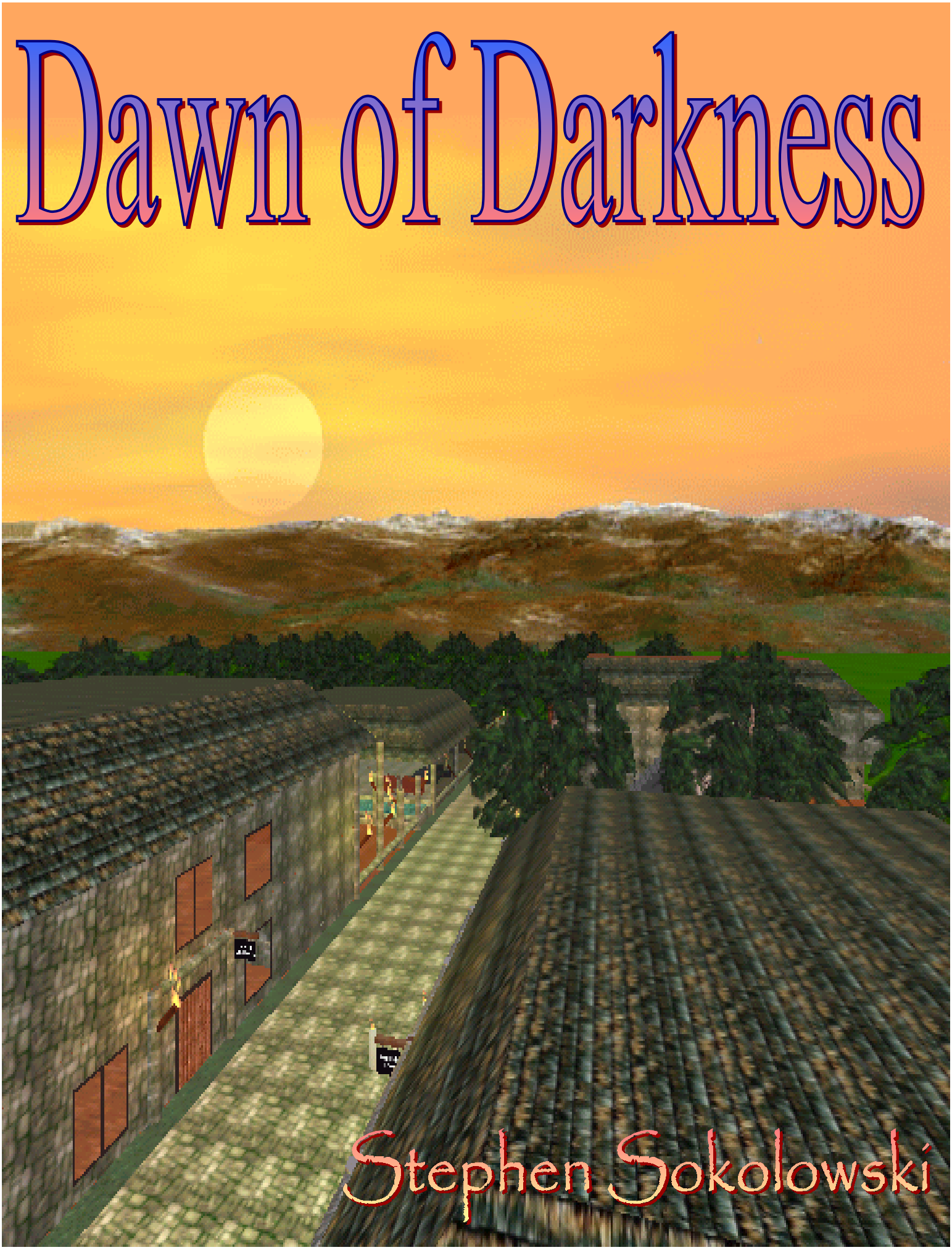
Dawn of Darkness



Stephen Sokolowski

Inside the graduation portfolio

Stephen Sokolowski · Upper Moreland High School · Spring 2001

The ten original tab titles give the order for this reconstruction of Dawn of Darkness, the role-playing game set in Gulakon. Page references lead to the surviving material.

Letter to Panel 3
Stephen’s letter to the presentation panel, dated March 27.

Proposal
The digital files contain the parent approval letter. The submitted proposal is missing.

Parent/Guardian Release 4
Typed approval of the proposal. The signed binder copy is unverified.

Résumé
The surviving résumé was revised in January 2002. The submitted version is missing.

Research Paper 5–16
Computing: Past and Present, including the original bibliography.

Log/Journal 17–20
Dated entries from January 26 through May 9.

Timeline/Checklist
The checklist is missing. Slides 9–16 record the project milestones.

Visual Basic Source Code 21–48
Two programming screenshots and five surviving class files.

Other Materials 49–72
A printed picture layout, 22 scene images, and the binder spine.

Sign of Commitment 73
The original 11-by-14-inch poster.

Presentation and supporting records

Graduation project presentation 74–95
All 22 slides. A frame from the original film appears on slide 18.

The recorded tour 96
Six film frames and access to the complete recording. Watch on YouTube

Research notes 97–99
Working notes on the history of computing and the Internet.

Sources and speaker notes 100
How the digital originals relate to the paper binder.

Stephen Sokolowski

March 27, 2001

Dear Graduation Project Presentation Panel Member:

"Life consists not of having good cards, but in playing those you hold well."

The above quotation accurately describes my outlook on life. While everyone is not gifted with the same skills, one should make the best of his or her talents. Perhaps, this philosophy is part of the reason why I have chosen to pursue a wide variety of interests throughout high school, including the publication of my own original magazine and my participation in various activities such as the chess club and the school newspaper. I have made a strong commitment to community service, through my help at a local basketball organization and church group, and strive to maintain my high academic standing. Being a religious person, I have always possessed a strong moral distinction between right and wrong.

In recent years, I have participated in many computer-related activities, one of which involves keeping an online virtual world safe for visitors of all ages. Through my experience, I have gained experience in computer programming and have developed many applications for various purposes. It is only appropriate that I should combine these two interests into completing a task which I had wanted to complete for years: developing an interactive online experience for users around the world to interact in friendly competition.

Hopefully, when you have the opportunity to view my accomplishments, you will be interested in seeing a log of pictures of the project from its start to its completion. In addition, you may wish to examine the source code of the program to ensure its authenticity. If an Internet connection is available in the presentation room, you may wish to visit the environment online to experience the results yourself. I am confident that you will gain a better sense of my accomplishments through your observation of this presentation.

I would like to extend my sincere appreciation for the service that you provide to the community by viewing the projects of Upper Moreland students. I eagerly await the opportunity to meet you in person on May 10.

Sincerely,

Stephen Sokolowski
Senior
Upper Moreland High School

I have read and approve Stephen Sokolowski's Graduation Project proposal.

Sincerely,

Henry J. Sokolowski
Parent

Computing: Past and Present

Stephen Sokolowski

It’s after midnight on a Sunday morning. Tony turns on the oven and enters the temperature at which he will be cooking a pizza. Minutes later, he puts the pizza in the oven and takes it out after the timer that he set counts down to zero. Elsewhere, Rick, almost too tired to drive home after a long night of partying, narrowly avoids an accident as he slams on his anti-lock brakes, enabling his car to stop in the nick of time. Henry, just a few blocks away, turns on an electric toothbrush to clean his teeth before retiring for the night. Down the street, a phone rings, but the occupants of the house are too tired to answer and leave their answering machine to take a message.

All of these events have one common theme: computers at work. Computing technology invades every aspect of our daily lives, even in places that may seem insignificant. Chips in modern ovens regulate the oven's temperature to tenths of a degree to help save energy. Advances in automobile technology allow a computer-controlled brake system to avoid slipping. Electric toothbrushes calculate the duration for which a person’s teeth should be brushed and turn off when finished. Answering machines save voice data on electronic media, rather than on obsolete magnetic tapes. By living in today’s society, one cannot exist for more than a few minutes without coming into contact with something that is controlled by a computer, which is why I would like to investigate the origins of today’s information-driven world.

In my experiences with computing, I have discovered a great deal about the inner workings of computers, including how to manipulate them to serve a specific user’s needs. In addition, I have been exposed to a basic history of the computer, including

the beginnings of various programming languages and different types of computer systems, both in the past and present. I have absorbed some information about the growth of the Internet, as well as solutions to many of the computing problems of the past.

However, despite my previous knowledge, I have never performed an in-depth investigation of the origins of the computer. Questions such as "What were the beginnings of the computer in ancient times?", "What was the order of development of computer systems and hardware components from the 1940s until the present?", and "What was the origin of the Internet and how did it develop?" are all valid questions that are pertinent to the information sciences field.

A variety of sources have been very useful in gathering data to find answers to these questions. A search at the Huntingdon Valley library produced several noteworthy pieces of literature on the subject. In addition, several sources of information were located through the use of the Internet. Magazines, newspapers, and other periodicals were useful in the prediction of future trends in computing. A personal interview with Dr. Marvin Aronson, a professor at Penn State University, yielded a copious amount of useful information. Finally, an article in an electronic encyclopedia allowed me to gain some experience in basic computer history. Through these sources, I was able to obtain more than enough information to write a thesis paper but have condensed relevant topics to fit on these few pages.

While many people believe that computers have been in use for little more than a half century, automated calculating machines have existed in the minds of researchers since the beginning of human development. Prior to the technological revolution, a

computer was defined as a person who was employed to perform calculations for scientists or other researchers who could then concentrate on the theory instead of the numbers and arithmetic. These *computers* labored long hours, working through tedious mathematical functions that today can be performed by a calculator. Realizing how much time could be saved by the development of a mechanical adding machine, several inventors developed such a device in 1642. It had wheels that, when turned in the proper order, would give the numerical result of a calculation. These calculations still took a great deal of time, however. It wasn't until Charles Babbage realized in 1812 that because many problems consist of the same repeated operations, a machine could be developed (later called the difference engine) that would automate these calculations by performing the various steps in succession.

The next breakthrough in computing evolved in 1890, when Herman Hollerman developed a device that could read information from cards for the census bureau. Previously, a census would take seven years to complete, but the first automated census of 1890 took just three. Hollerman later founded a company called IBM, which would become one of the leading manufacturers of computers in the 20th century.

The Mark I computer, developed in the 1930s by Howard Eiken at IBM, was the first mechanical calculator that could perform all four arithmetic operations. This and many other machines used the decimal system for their calculations, which increased processing time. Konrad Zuse later realized that speed could be increased by developing a machine that used the binary system to perform calculations.

A revolutionary breakthrough occurred in 1945 when the ENIAC was developed at the University of Pennsylvania. The system, which was the first electronic computer,

utilized over 18,000 vacuum tubes to store just one thousand bytes of data. There were too many tubes for the system to be of any practical use for much else than a calculator, which is why the EDSAC, ENIAC's successor, was developed in May 1949. Using what were referred to as "mercury delay lines" instead of vacuum tubes, the processing power was increased by a factor of six, and the size of the new machine was only one-sixth the size of the ENIAC.

In 1951, the computer began to shift from a mathematical tool to a business machine. The Ferranti Mark I, based on the original Mark I model, was the first computer on the market for business use. The British manufacturer was able to take advantage of IBM's refusal to enter the electronic computer field. IBM had previously stated that the machine would only be used for mathematics, never to become a business tool.

IBM later went back on its word to develop Card Programmed Calculators. Produced in the mid-1950s, these machines used punched cards to perform basic calculations. The same type of cards had been used since the 1800s to store data on machines such as the Census Bureau's counting machine. Thus, these cards became the staple of computing throughout the 1950s and early 1960s. When it became clear that loading and unloading the punched cards one-by-one was an enormous waste of time, IBM and its competitors began work on a "stored program" computer, the purpose of which was to eliminate the need for punchcards by storing multiple programs on a disk at the same time.

In 1960, the first computer that used transistors, instead of vacuum tubes, was developed. Transistors cut the space requirement for a computer drastically, while

allowing computers to develop at an incredible pace. By the middle of the decade, the computer field had been narrowed down to a competition between IBM and five smaller companies each holding a tiny portion of the market share. Thus, when IBM developed its New Product Line in an attempt to make all future software compatible with IBM computers, they solidified the IBM format for the majority of personal computers that are still in use today.

Even so, PC's didn't develop until 1975, when the Altair 8800 was released. The Altair 8800 was called a "minicomputer", as the term "personal computer" had not yet been coined. The Altair was more of a hobbyist machine, had to be programmed directly with the 1's and zeroes of binary code, and had only 256 bytes of memory. In the same year, digital watches and portable calculators were coming into existence at a reasonable price.

Apple computer and the Tandy Corporation (of which Radio Shack is a part) developed the second generation of personal computers in 1977, mostly designed around the ability to use them to play games. Video games had become wildly popular in the mid- to late-70s and were now the major impetus in the personal computer revolution. Apple, which entered the PC market before IBM, was given an advantage in the early days of the PC.

In 1975, Bill Gates realized that sales of the 8800 would increase if a version of the BASIC programming language were developed for Altair's system. He sought a contract, which he later won, with Altair to develop the language. After his success with this programming task, he founded Microsoft. In the early 1980s, IBM had lost its dominance of the hardware market to Apple, which was developing a Graphical User

Interface easier to use than IBM's DOS. IBM then awarded a contract to Microsoft to develop a version of DOS for all its PC's. Since then, Microsoft has grown exponentially. A startup company called Intel was also awarded a contract to produce processors for IBM computers, and the three major companies of the PC revolution were solidified.

By 1983, most major software packages ran on IBM computers, but there was a need for a user-friendly interface to shorten or eliminate the 20-hour estimated time requirement to learn a single software package. Microsoft released version 1.0 of its Windows operating system, which was a success because it mirrored Apple's already existent GUI. Windows continued to develop throughout the 1980s and 90s until it now controls over 95% of the operating system installations in the United States.

In January 1983, Time magazine selected the PC as its "Man of the Year", which spurred interest in computers. Reacting swiftly, competing companies flooded the market with new PC's. The average user became confused with the multitude of options that were available, resulting in a recession in the market for a few years. When the market recovered, Microsoft, Apple, and IBM competed for software dominance until the early 1990s. Microsoft eventually won the high-stakes game because of its superior advertising power.

The Internet had its origins in French philosopher Denis Diderot, who created an encyclopedia in the 18th century, wanting to make the world's store of knowledge available to ordinary citizens. Diderot's encyclopedia was the first attempt to gather as much knowledge as possible into one place. At that time, it was possible to compile all of the world's knowledge into one set of encyclopedias or even for one person to gain a

relatively good understanding of everything the world knew by attending college. As the available information increased, however, and a single person was no longer able to learn the entire world's store of knowledge in one lifetime, more sophisticated methods were sought to catalogue and make use of this immense knowledge.

The increasing store of knowledge around the turn of the 20th century was leading to a trend toward specialization among the sciences. Now, one mathematician who could have been familiar with biology and chemistry as well as all of mathematics in the 1700s couldn't even keep up with all the various branches of his own specialty. Several leading authorities in the information field believed that they could gather all of the information into one easy-to-access place to reverse this specialization movement. Though their efforts failed, H.G. Wells, well into his seventies by 1937, proposed to then-President Roosevelt that a project be funded to create a World Encyclopedia. Wells realized that all the information might not necessarily need to be stored in one specific place. He also acknowledged that the information might not be stored on paper, but on microfilm or a completely different medium. Roosevelt, however, concerned with deteriorating conditions in Europe, turned down the proposal and Wells died shortly thereafter.

Vannever Bush, head of the Office of Scientific Research and Development during World War II, revived the project after Wells' death. Technology had advanced to the point where microfilm was feasible as a storage medium; however, making use of the information was now the problem. He conceived of a "memex", a desk where a person sat that had television screens, onto which the information was projected for easy access by the user.

In the mid 1960s, Ted Nelson coined the term Hypertext, and Hypertext Transfer Protocol (HTTP) would later become the protocol for website browsing. Later, JCR Licklider made a proposal to develop what he called ARPANET, a system that would link computers across the country to conserve resources. The problem at the time was largely the incredible expense of computers. Most machines were unused during the night and on weekends, while they were badly needed during normal working hours. By using the ARPANET, the extra processing power of the computers on the West Coast could be used in the morning on the East Coast, when nobody was in the West Coast offices. Conversely, the West Coast offices could use East Coast computers to perform their calculations in the evenings.

Larry Roberts took over the project from Licklider and developed "packet switching", which was based on the technology that was used in the 1800s to enable telegrams to travel efficiently across the country. At each major city, there was a hub, where telegrams would be written down by employees and then forwarded to the next hub. The process was repeated until each telegram reached its destination. In computers, data would be broken up into packets, transferred to central hubs, and then forwarded to the appropriate location. In addition, if the connecting lines became busy, the packets could be withheld without any difficulty until the lines were free. A different operating system would be installed at these nodes, which retransmitted the data to each connected computer. As a result, the individually incompatible systems required only a simple decoding scheme to receive the data of the central hub.

The International Conference on Computer Communications in 1972 showcased the ARPANET network of computers. Onlookers were amazed and soon after wanted

to join the ARPANET or to develop their own networks. They were astounded by an electronic mail system that two programmers had written in their spare time the previous year, which allowed users to send messages to one another with ease. Because of electronic mail capabilities, the ARPANET expanded greatly over the next few years, and e-mail has been the driving force behind the Internet ever since.

Eventually, networks other than the ARPANET were formed, and a "netiquette" developed because it was impossible to determine a user's state of emotion through text-based e-mails. In addition, because e-mails couldn't be sent between computer networks, researchers set to work developing an *inter*-networking solution that would network networks.

This new Internet didn't grow very quickly until Al Gore proposed the National Information Infrastructure in 1991 as a campaign issue. Spurred by the PC revolution, people quickly became connected to the Internet. However, by the mid-1990s, people began to post entire documents on the Internet in addition to e-mail. Cataloguing and searching through the documents was impossible, because there was no universal search system in use. Thus, while the Internet offered information on almost every topic imaginable, most of this knowledge was unable to be located because there were no central search engines.

To remedy the problem, "gopher" was developed at the University of Minnesota. The gopher system was the first search engine. Eventually, other search engines evolved, both similar and different from “gopher”, but the problem of finding useful information remains largely unsolved. Search engines can still only access a small percentage of the Internet.

Computers and the Internet have had a long, detailed history that involved the dedication of countless people throughout the years. Having researched much of the history of computers and the Internet, I hope to learn more about these topics through the completion of my project. By developing a game using C++, a language that has been in existence since the early 1970s, I will have the opportunity to learn about the way computers were in the past. By using existing 3-dimensional technology, I will be able to explore the possibilities for computers in the future. Hopefully, my project will contribute in some way to the long history of computer development that has lasted for decades and is still ongoing.

Bibliography

Aronson, Marvin E. MD. Personal Interview. 14 March 2001. Eastern Center for Arts and Technology, Willow Grove, Pennsylvania. 8:50AM-9:20AM.

Campbell, Kelly, and William Aspray. Computer: A History of the Information Machine. New York: BasicBooks, 1996.

"Computer." Microsoft Encarta Encyclopedia Standard 2001. n.c.: Microsoft, 2000. CD-ROM.

"Computer History, Computer Dictionary, and Computer Glossary." Online. Accessed 15 March 2001: www.compinfo-center.com/tpdict-t.htm

Dvorak, John C. "The Modernization of E-mail." PC Magazine. N.c.: CNET Networks, 20 March 2001. Online. www.zdnet.com/pcmag/stories/opinions/0,7802,2692066,00.html

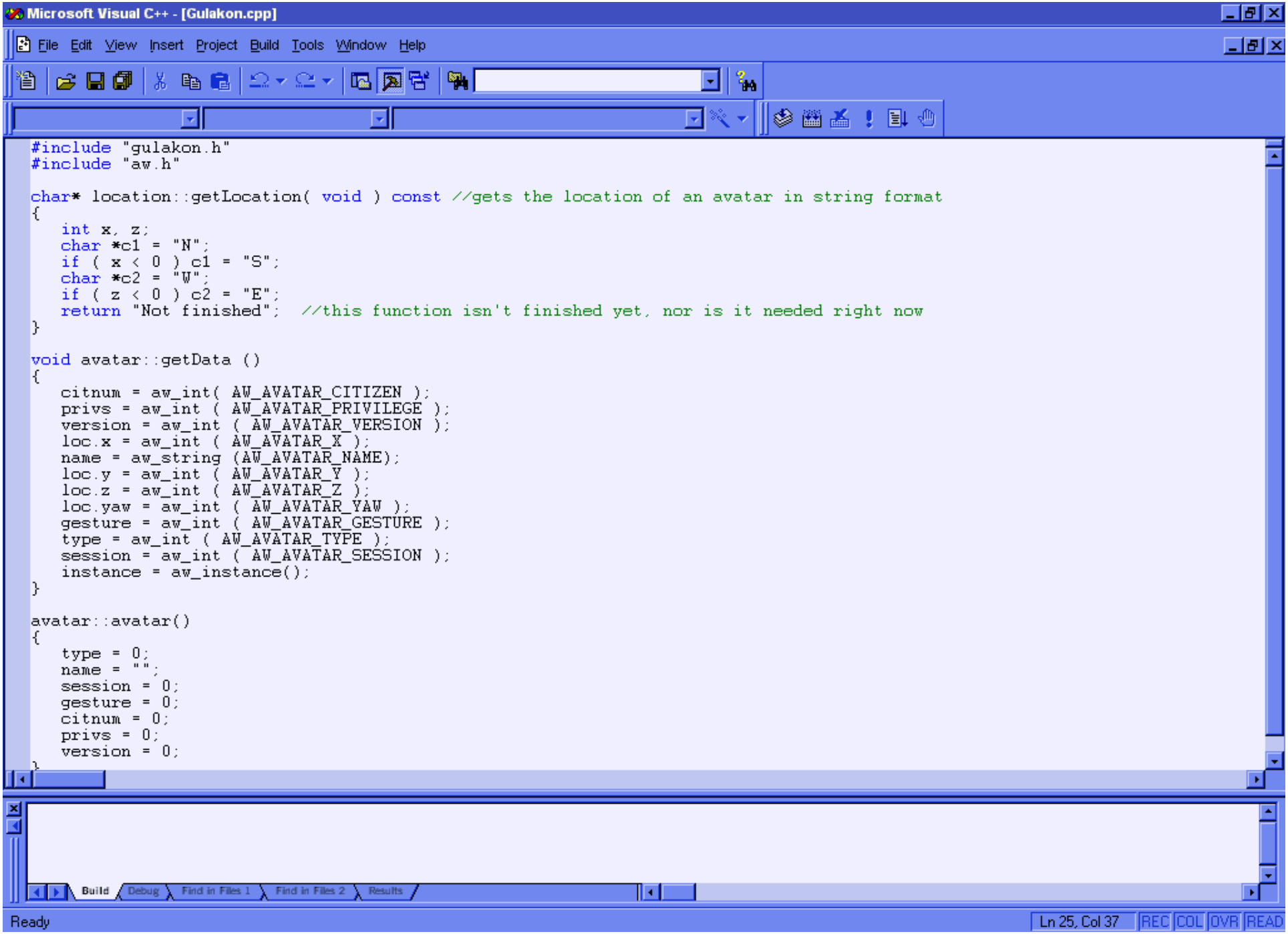
Stephen Sokolowski's Log/Journal

Date:	Hours:	Activity:
1/26	1.5	During the first day of the project class, I listened to an overview of the project and began considering ideas.
1/30	1.5	Today, I listened to Mrs. Miller talk about the proposal and downloaded the proposal to my school computer file. In addition, I created this log journal and made this first entry. As the class ended, I was still considering several ideas as the topic for my project.
2/2	1.5	Mrs. Miller showed a video on past graduation projects that she had produced for a state education agency. She also reviewed the various aspects of the project.
2/9	1.5	I made a first draft of my proposal, thinking that I would not choose the topic about which I was writing. However, I later decided to select the computer game as my project and used the draft I wrote on February 9.
2/11	1.0	I entered the Activeworlds universe and held a preliminary meeting for the project with various volunteer graphic designers.
2/13	0.2	I handed the proposal to my advisor to have it approved.
2/15	1.5	I handed in the proposal and did some preliminary research into my research paper topic.
2/19	1.0	I entered the Universe and took pictures of various impressive creations to use on my sign of commitment.
2/21	1.0	I downloaded the Paint Shop Pro software, which I used to create my sign of commitment. After downloading, installing, and configuring the software (approximately ½ hour), I spent the other half hour formatting my sign.
2/22	2.5	I finished the sign of commitment and went to a nearby OfficeMax to have the poster printed on 11-by-14 inch paper. There had been an early dismissal this day, and, because I thought that the sign was to be finished by the next day, I went to the OfficeMax in the snow. By the time I returned home, we had spent over an hour on the road.
2/27	1.6	In the morning, I handed in my sign of commitment to Mr. Levin and handed in the sign of commitment to Mrs. Miller later in the day. I then went to the library to begin research for the research portion of the project and looked for books, articles, and Internet resources.
3/7	1.5	I listened to Mrs. Miller's dissertation on good interviewing skills.
3/8	1.0	I talked with Jimmy Rothfork (from California) and the

		building of the graphic design of the project began.
3/9	1.5	I worked on the first four paragraphs of the research paper and conferenced in “response groups” where peers determined what parts of the research paper could be improved.
3/12	3.5	In the morning, I did some Internet research into my project paper. In the evening, I began the programming portion of the project. I started today by designing classes for use in the program and worked toward making the robots log in to the environment. When I quit for the day, the program had bugs and was not operational.
3/13	1.7	In the morning, I talked with Dr. Aronson, my mentor, about the project and asked for some assistance in solving a problem. In the afternoon, I did some more research into my paper.
3/15	1.5	I continued researching the paper through books and the Internet.
3/17	6.0	I began the day by working on the project paper and completing the research. I took breaks every hour, during which time I entered the Activeworlds universe and contacted the helpers again to check on their progress. Later, I worked on the writing of the paper.
3/19	1.5	I wrote the bulk of my research paper.
3/21	2.5	I completed the research paper, compiled a bibliography, and prepared to submit the paper to my advisor.
3/22	0.5	I proofread the paper again and gave it to Mr. Levin.
3/25	3.0	After struggling with C++ programming, I was forced to change the programming language of the project to Visual Basic. I began by creating classes that would be of use later in the program's development.
3/26	1.0	I continued the work on the program and took pictures of the environment for my portfolio.
3/27	2.5	I watched Mrs. Miller's presentation on the letter to the panel and wrote the letter. Later, I continued programming the classes for the project.
3/28	1.0	I continued programming the classes for the program.
3/29	1.5	I proofread and finished the letter to the panel.
3/30	2.5	Someone deleted all of the files necessary for the visual display of the game. I spent two and a half hours determining the cause of the problem.
3/31	2.0	I attempted to restore the object files to a different location on the Internet, but the person who was responsible for the deletion contacted the hosting service and had my account canceled.
4/1	3.0	I re-uploaded the files twice again, but the same person contacted the hosting services with unknown information

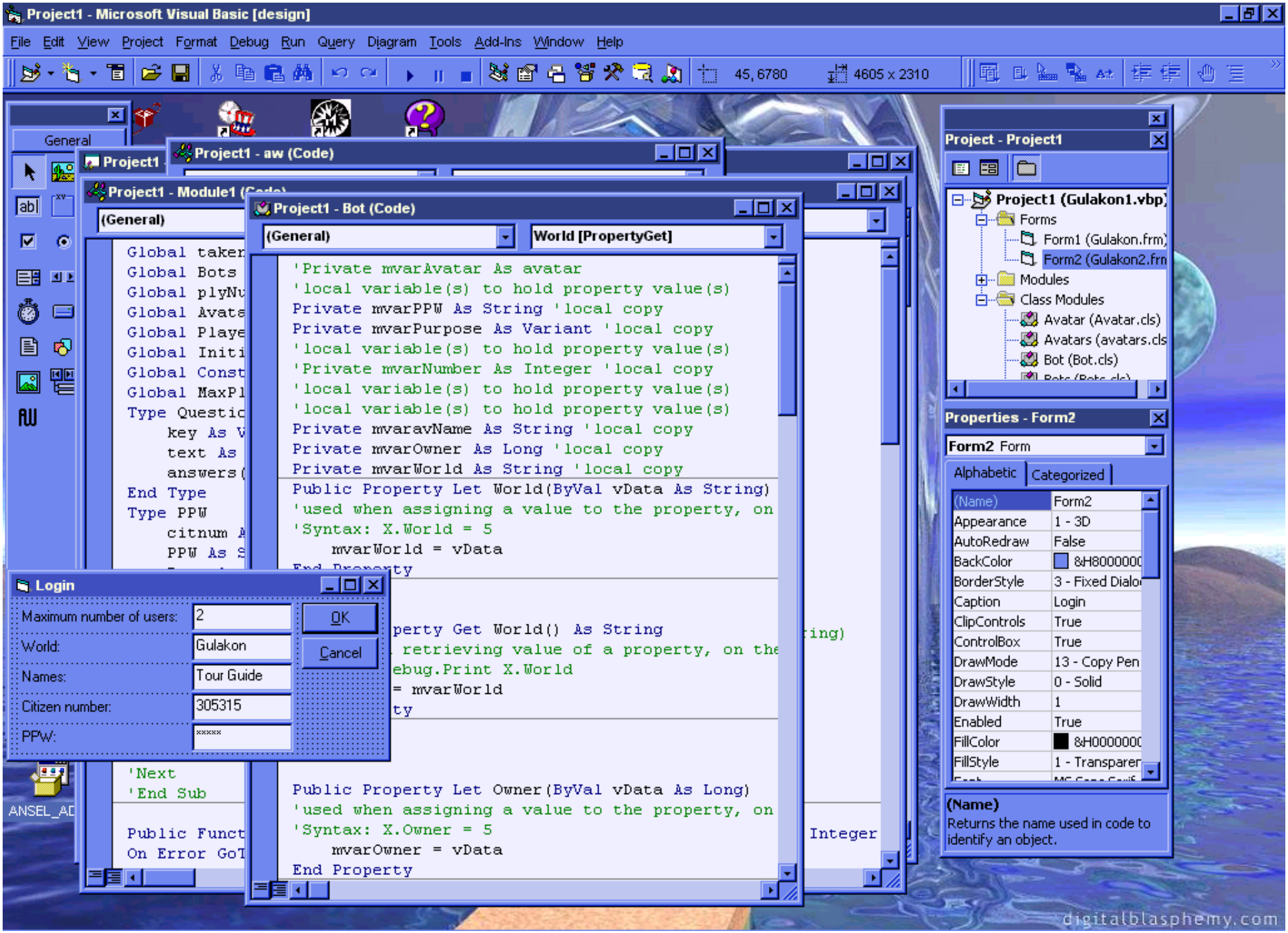
		that prompted them to cancel my account.
4/2	2.0	I attempted re-uploading the files to a total of six sites by the end of the night. All of the sites were immediately deleted.
4/3	1.0	I contacted Activeworlds support with a complaint against the person I suspected to be the perpetrator of these incidents. They agreed to help.
4/4	1.5	I worked with AW staff by sending them the necessary logs and other information so that I could upload the objects and continue my project.
4/7	3.0	I transferred the necessary files to my own web server (shoemakervillage.org), and I resumed working on the project.
4/8	2.0	I continued working on the basic subroutines and classes necessary for the backbone of the game.
4/10	1.5	I made a breakthrough and finally was able to debug the classes necessary to make the game function. Afterward, I started on the character creation routines and the music playing routines.
4/11	1.5	I began work on my résumé.
4/12	2.0	I was able to get music to play in the game upon entering. I fixed some bugs that I hadn't noticed before. I built a "chat buffer" which stored text so that it wouldn't all be sent to the player at once and made a "command buffer" as well.
4/16	2.0	I finished the command buffer and began investigating the use of bots as "shopkeepers" and "enemies" during battles. I investigated whether it would be better to use fully controlled robots as animated and talking shopkeepers, which would be best, or simple objects which would be easier but non-animated and count less toward the bot limit that AW imposes, which is six. I decided that the best course of action would be to use bots.
4/17	1.5	I worked with Activeworlds to have the limit on the number of users that could play at once raised from 5 to 15.
4/19	1.0	I began work on creating virtual shopkeepers for the stores in the game.
4/20	2.3	I completed miscellaneous activities and began the preparation of my portfolio and watched a video of graduation project presentations in 1997. Later, I walked to the OfficeMax with a friend to purchase a binder and coverings.
4/21	3.0	My computer crashed because of a virus that may have been accidentally spread from the school network. I spent three hours scanning the computer and restoring the computer back to the day before I sent home the document.
4/22	1.5	I spent most of the time testing asynchronous commands

		with the network game. If I had used synchronous commands, the game would "freeze" after certain actions but would not with asynchronous commands. I had never used the asynchronous mode before and was forced to spend a great deal of time testing. I had not been able to get the asynchronous commands working by the end of this session.
4/23	1.0	I continued coordinating the visual part of the game and helped build for a short period of time. I E-Mailed someone with expertise on asynchronous commands and he replied with some helpful advice. Unfortunately, I was unable to work on the project until he replied.
4/24	1.5	I completed my portfolio and submitted it to Mr. Levin.
4/26	1.5	I began work on my presentation and submitted my portfolio to Mrs. Miller. In addition, I talked to a person at the administration about the feasibility of setting up the game on the school network. Because of many reasons, including a lack of time during the presentation and problems with the school computer system, I decided that the best course of action would be to prepare other visuals.
4/28	3.0	I worked on my final presentation and had it proofread. I then revised the presentation.
4/30	1.0	I continued working on the programming and attempted to complete as much as possible before the deadline.
5/1	1.0	I compiled various pieces of documentation to include in my portfolio and began designing the cover and spine of the binder in WordPerfect.
5/2	1.0	After the WordPerfect materials wouldn't print, I transferred the covers into Word and printed them.
5/3	0.2	I visited Mrs. Miller's room to insert the materials into my portfolio.
5/4	0.3	I updated this Log/Journal.
5/7	2.0	I printed, compiled, and prepared the source code and pictures of the game. I prepared my presentation and began practicing it.
5/8	3.0	I delivered the materials to my portfolio, created a video, and practiced my presentation.
5/9	2.0	I practiced my presentation.
Total	**95.8**	



The C++ project in Microsoft Visual C++

C++.bmp



Gulakon in the Visual Basic development environment

BASIC.bmp

Avatar.cls

Archive copy dated April 12, 2001 · Lines 1–47 of 276

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
  Persistable = 0  'NotPersistable
  DataBindingBehavior = 0  'vbNone
  DataSourceBehavior  = 0  'vbNone
  MTSTransactionMode  = 0  'NotAnMTSObject
END
Attribute VB_Name = "Avatar"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = False
Attribute VB_Description = "Used for avatars for the AW environment."
Attribute VB_Ext_KEY = "SavedWithClassBuilder6" ,"Yes"
Attribute VB_Ext_KEY = "Top_Level" ,"Yes"
'local variable(s) to hold property value(s)
Private mvarX As Long 'local copy
Private mvarY As Long 'local copy
Private mvarZ As Long 'local copy
Private mvarYaw As Integer 'local copy
Private mvaravName As String 'local copy
Private mvarSession As Long 'local copy
Private mvarCitnum As Long 'local copy
Private mvarBuild As Integer 'local copy
Private mvaravType As Byte 'local copy
Private mvarPrivilege As Long 'local copy
Private mvarGesture As Byte 'local copy
Private mvarWorld As String 'local copy
'local variable(s) to hold property value(s)
'local variable(s) to hold property value(s)
'Private mvarNumber As Integer 'local copy
'Public Property Let Number(ByVal vData As Integer)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Number = 5
'    mvarNumber = vData
'End Property
'local variable(s) to hold property value(s)
Private mvarIndex As Integer 'local copy

Public Sub TeleportBack()
Form1.aw(Index).AwTeleportX = X
Form1.aw(Index).AwTeleportY = Y
Form1.aw(Index).AwTeleportZ = Z
Form1.aw(Index).AwTeleportWarp = False
Form1.aw(Index).AwTeleport session
End Sub
```

Avatar.cls

Archive copy dated April 12, 2001 · Lines 48–93 of 276

```
48
49  Public Property Let Index(ByVal vData As Integer)
50  'used when assigning a value to the property, on the left side of an assignment.
51  'Syntax: X.Index = 5
52      mvarIndex = vData
53  End Property
54
55
56  Public Property Get Index() As Integer
57  'used when retrieving value of a property, on the right side of an assignment.
58  'Syntax: Debug.Print X.Index
59      Index = mvarIndex
60  End Property
61
62
63
64
65
66  'Public Property Get Number() As Integer
67  'used when retrieving value of a property, on the right side of an assignment.
68  'Syntax: Debug.Print X.Number
69  '    Number = mvarNumber
70  'End Property
71
72
73  Public Sub GetData(Index2 As Integer)
74  Number = Index2
75  X = Form1.aw(Index2).AwAvatarX
76  Yaw = Form1.aw(Index2).AwAvatarYaw
77  Y = Form1.aw(Index2).AwAvatarY
78  Z = Form1.aw(Index2).AwAvatarZ
79  Build = Form1.aw(Index2).AwAvatarVersion
80  Citnum = Form1.aw(Index2).AwAvatarCitizen
81  Gesture = Form1.aw(Index2).AwAvatarGesture
82  Privilege = Form1.aw(Index2).AwAvatarPrivilege
83  avName = Form1.aw(Index2).AwAvatarName
84  session = Form1.aw(Index2).AwAvatarSession
85  avType = Form1.aw(Index2).AwAvatarType
86  Index = Index2
87  End Sub
88
89  Public Sub Eject(Optional Time As Long = 5)
90  End Sub
91
92  Public Property Let World(ByVal vData As String)
93  'used when assigning a value to the property, on the left side of an assignment.
```

Avatar.cls

Archive copy dated April 12, 2001 · Lines 94–137 of 276

```
'Syntax: X.World = 5
    mvarWorld = vData
End Property


Public Property Get World() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.World
    World = mvarWorld
End Property



Public Sub Teleport(X As Long, Y As Long, Z As Long, Yaw As Long, Optional World As
 String = LoginWorld$)

End Sub

Public Property Let Gesture(ByVal vData As Byte)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Gesture = 5
    mvarGesture = vData
End Property


Public Property Get Gesture() As Byte
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Gesture
    Gesture = mvarGesture
End Property



Public Property Let Privilege(ByVal vData As Long)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Privilege = 5
    mvarPrivilege = vData
End Property


Public Property Get Privilege() As Long
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Privilege
    Privilege = mvarPrivilege
End Property
```

Avatar.cls

Archive copy dated April 12, 2001 · Lines 138–184 of 276

```
138
139
140
141 Public Property Let avType(ByVal vData As Byte)
142 'used when assigning a value to the property, on the left side of an assignment.
143 'Syntax: X.avType = 5
144     mvaravType = vData
145 End Property
146
147
148 Public Property Get avType() As Byte
149 'used when retrieving value of a property, on the right side of an assignment.
150 'Syntax: Debug.Print X.avType
151     avType = mvaravType
152 End Property
153
154
155
156 Public Property Let Build(ByVal vData As Integer)
157 'used when assigning a value to the property, on the left side of an assignment.
158 'Syntax: X.Build = 5
159     mvarBuild = vData
160 End Property
161
162
163 Public Property Get Build() As Integer
164 'used when retrieving value of a property, on the right side of an assignment.
165 'Syntax: Debug.Print X.Build
166     Build = mvarBuild
167 End Property
168
169
170
171 Public Property Let Citnum(ByVal vData As Long)
172 'used when assigning a value to the property, on the left side of an assignment.
173 'Syntax: X.Citnum = 5
174     mvarCitnum = vData
175 End Property
176
177
178 Public Property Get Citnum() As Long
179 'used when retrieving value of a property, on the right side of an assignment.
180 'Syntax: Debug.Print X.Citnum
181     Citnum = mvarCitnum
182 End Property
183
184
```

Avatar.cls

Archive copy dated April 12, 2001 · Lines 185–230 of 276

```
185
186 Public Property Let session(ByVal vData As Long)
187 'used when assigning a value to the property, on the left side of an assignment.
188 'Syntax: X.Session = 5
189     mvarSession = vData
190 End Property
191
192
193 Public Property Get session() As Long
194 'used when retrieving value of a property, on the right side of an assignment.
195 'Syntax: Debug.Print X.Session
196     session = mvarSession
197 End Property
198
199
200
201 Public Property Let avName(ByVal vData As String)
202 'used when assigning a value to the property, on the left side of an assignment.
203 'Syntax: X.avName = 5
204     mvaravName = vData
205 End Property
206
207
208 Public Property Get avName() As String
209 Attribute avName.VB_UserMemId = 0
210 'used when retrieving value of a property, on the right side of an assignment.
211 'Syntax: Debug.Print X.avName
212     avName = mvaravName
213 End Property
214
215
216
217 Public Property Let Yaw(ByVal vData As Integer)
218 'used when assigning a value to the property, on the left side of an assignment.
219 'Syntax: X.Yaw = 5
220     mvarYaw = vData
221 End Property
222
223
224 Public Property Get Yaw() As Integer
225 'used when retrieving value of a property, on the right side of an assignment.
226 'Syntax: Debug.Print X.Yaw
227     Yaw = mvarYaw
228 End Property
229
230
```

Avatar.cls

Archive copy dated April 12, 2001 · Lines 231–276 of 276

```
231
232 Public Property Let Z(ByVal vData As Long)
233 'used when assigning a value to the property, on the left side of an assignment.
234 'Syntax: X.Z = 5
235     mvarZ = vData
236 End Property
237
238
239 Public Property Get Z() As Long
240 'used when retrieving value of a property, on the right side of an assignment.
241 'Syntax: Debug.Print X.Z
242     Z = mvarZ
243 End Property
244
245
246
247 Public Property Let Y(ByVal vData As Long)
248 'used when assigning a value to the property, on the left side of an assignment.
249 'Syntax: X.Y = 5
250     mvarY = vData
251 End Property
252
253
254 Public Property Get Y() As Long
255 'used when retrieving value of a property, on the right side of an assignment.
256 'Syntax: Debug.Print X.Y
257     Y = mvarY
258 End Property
259
260
261
262 Public Property Let X(ByVal vData As Long)
263 'used when assigning a value to the property, on the left side of an assignment.
264 'Syntax: X.X = 5
265     mvarX = vData
266 End Property
267
268
269 Public Property Get X() As Long
270 'used when retrieving value of a property, on the right side of an assignment.
271 'Syntax: Debug.Print X.X
272     X = mvarX
273 End Property
274
275
276
```

avatars.cls

Archive copy dated April 12, 2001 · Lines 1–45 of 135

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
  Persistable = 0  'NotPersistable
  DataBindingBehavior = 0  'vbNone
  DataSourceBehavior  = 0  'vbNone
  MTSTransactionMode  = 0  'NotAnMTSObject
END
Attribute VB_Name = "Avatars"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = False
Attribute VB_Ext_KEY = "SavedWithClassBuilder6" ,"Yes"
Attribute VB_Ext_KEY = "Collection" ,"Avatar"
Attribute VB_Ext_KEY = "Member0" ,"Avatar"
Attribute VB_Ext_KEY = "Top_Level" ,"Yes"
'local variable to hold collection
Private mCol As Collection

Public Function SearchList(avName1 As String)
Dim indexvar As Integer
For indexvar = 1 To Count
    If Avatars(its(indexvar)).avName = avName1 Then
        SearchList = (its(indexvar))
        Exit Function
    End If
Next
SearchList = -1
End Function
'Public Function SearchNum(avNum As Integer)
'Dim indexvar As Integer
'For indexvar = 1 To Avatars.Count
'    If Avatars.Item(indexvar).Number = avNum Then
'        SearchNum = indexvar
'        Exit Function
'    End If
'N ext
'SearchNum = -1
'End Function
Public Function SearchSession(avSession As Long)
Dim indexvar As Integer
For indexvar = 1 To Count
    If Avatars(its(indexvar)).session = avSession Then
        SearchSession = its(indexvar)
```

avatars.cls


```
46          Exit Function
47      End If
48  Next
49  SearchSession = -1
50  End Function
51
52
53  Public Function Add(Optional sKey As String) As avatar
54      'create a new object
55      Dim objNewMember As avatar
56      Set objNewMember = New avatar
57
58
59      'set the properties passed into the method
60      'objNewMember.X = X
61      'objNewMember.Y = Y
62      'objNewMember.Z = Z
63      'objNewMember.Yaw = Yaw
64      'objNewMember.avName = avName
65      'objNewMember.session = session
66      'objNewMember.Citnum = Citnum
67      'objNewMember.Build = Build
68      'objNewMember.avType = avType
69      'objNewMember.Privilege = Privilege
70      'objNewMember.Gesture = Gesture
71      'objNewMember.World = World
72      'objNewMember.Number = Number
73      If Len(sKey) = 0 Then
74          mCol.Add objNewMember
75      Else
76          mCol.Add objNewMember, sKey
77      End If
78
79
80      'return the object created
81      Set Add = objNewMember
82      Set objNewMember = Nothing
83
84
85  End Function
86
87  Public Property Get Item(vntIndexKey As Variant) As avatar
88  Attribute Item.VB_UserMemId = 0
89      'used when referencing an element in the collection
90      'vntIndexKey contains either the Index or Key to the collection,
```

avatars.cls

Archive copy dated April 12, 2001 · Lines 91–135 of 135

```
91      'this is why it is declared as a Variant
92      'Syntax: Set foo = x.Item(xyz) or Set foo = x.Item(5)
93    Set Item = mCol(vntIndexKey)
94  End Property
95
96
97
98  Public Property Get Count() As Long
99      'used when retrieving the number of elements in the
100     'collection. Syntax: Debug.Print x.Count
101     Count = mCol.Count
102 End Property
103
104
105 Public Sub Remove(vntIndexKey As Variant)
106     'used when removing an element from the collection
107     'vntIndexKey contains either the Index or Key, which is why
108     'it is declared as a Variant
109     'Syntax: x.Remove(xyz)
110
111
112     mCol.Remove vntIndexKey
113 End Sub
114
115
116 Public Property Get NewEnum() As IUnknown
117 Attribute NewEnum.VB_UserMemId = -4
118 Attribute NewEnum.VB_MemberFlags = "40"
119     'this property allows you to enumerate
120     'this collection with the For...Each syntax
121     Set NewEnum = mCol.[_NewEnum]
122 End Property
123
124
125 Private Sub Class_Initialize()
126     'creates the collection when this class is created
127     Set mCol = New Collection
128 End Sub
129
130
131 Private Sub Class_Terminate()
132     'destroys collection when this class is terminated
133     Set mCol = Nothing
134 End Sub
135
```

Bot.cls

Archive copy dated May 8, 2001 · Lines 1–50 of 150

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
  Persistable = 0  'NotPersistable
  DataBindingBehavior = 0  'vbNone
  DataSourceBehavior  = 0  'vbNone
  MTSTransactionMode  = 0  'NotAnMTSObject
END
Attribute VB_Name = "Bot"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = False
Attribute VB_Ext_KEY = "SavedWithClassBuilder6" ,"Yes"
Attribute VB_Ext_KEY = "Member0" ,"Avatar"
Attribute VB_Ext_KEY = "Top_Level" ,"Yes"
'Private mvarAvatar As avatar
'local variable(s) to hold property value(s)
Private mvarPPW As String 'local copy
Private mvarPurpose As Variant 'local copy
'local variable(s) to hold property value(s)
'Private mvarNumber As Integer 'local copy
'local variable(s) to hold property value(s)
'local variable(s) to hold property value(s)
Private mvaravName As String 'local copy
Private mvarOwner As Long 'local copy
Private mvarWorld As String 'local copy
Public Property Let World(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.World = 5
    mvarWorld = vData
End Property


Public Property Get World() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.World
    World = mvarWorld
End Property



Public Property Let Owner(ByVal vData As Long)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Owner = 5
    mvarOwner = vData
End Property


Public Property Get Owner() As Long
```

Bot.cls

Archive copy dated May 8, 2001 · Lines 51–100 of 150

```
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Owner
    Owner = mvarOwner
End Property



Public Property Let avname(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.avName = 5
    mvaravName = vData
End Property


Public Property Get avname() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.avName
    avname = mvaravName
End Property









Public Sub Logout()
End Sub

'Public Property Let Number(ByVal vData As Integer)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Number = 5
'    mvarNumber = vData
'End Property


'Public Property Get Number() As Integer
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Number
'    If IsObject(mvarNumber) Then
'        Set Number = mvarNumber
'    Else
''    End If
'End Property



Public Function Login(Number As Integer) As Integer
Form1.aw(Number).AwInit AW_BUILD
Form1.aw(Number).AwCreate 0, 0
```

Bot.cls

Archive copy dated May 8, 2001 · Lines 101–150 of 150

```
'Form1.aw(Number).AwCallbackSet AW_CALLBACK_LOGIN
Form1.aw(Number).AwLoginName = avname
Form1.aw(Number).AwLoginPrivilegePassword = PPW
Form1.aw(Number).AwLoginOwner = Owner
rc = Form1.aw(Number).AwLogin
If rc <> 0 Then Login = rc: Exit Function
rc = Form1.aw(Number).AwEnter(World)
Login = rc
End Function

Public Property Let purpose(ByVal vData As Variant)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Purpose = 5
    mvarPurpose = vData
End Property


Public Property Set purpose(ByVal vData As Variant)
'used when assigning an Object to the property, on the left side of a Set statement.
'Syntax: Set x.Purpose = Form1
    Set mvarPurpose = vData
End Property


Public Property Get purpose() As Variant
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Purpose
    If IsObject(mvarPurpose) Then
        Set purpose = mvarPurpose
    Else
        purpose = mvarPurpose
    End If
End Property



Public Property Let PPW(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.PPW = 5
    mvarPPW = vData
End Property


Public Property Get PPW() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.PPW
    PPW = mvarPPW
End Property


```

Player.cls

Archive copy dated May 8, 2001 · Lines 1–54 of 605

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
  Persistable = 0  'NotPersistable
  DataBindingBehavior = 0  'vbNone
  DataSourceBehavior  = 0  'vbNone
  MTSTransactionMode  = 0  'NotAnMTSObject
END
Attribute VB_Name = "Player"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = False
Attribute VB_Ext_KEY = "SavedWithClassBuilder6" ,"Yes"
Attribute VB_Ext_KEY = "Top_Level" ,"Yes"
'local variable(s) to hold property value(s)
Public midi As String
Public ChatDelayValue As Single
Private MidiNumber As Long
Private midiX As Long
Private midiZ As Long
Private mvarchatbuffer() As String
Private mvarHP As Integer 'local copy
Private mvarMP As Integer 'local copy
Private mvarRace As String 'local copy
Private questionbuffer() As Question
Private mvarStatus As String 'local copy
Private mvarBattle As Boolean 'local copy
Private mvaravIndex As Integer 'local copy
'local variable(s) to hold property value(s)
Private mvarattAtk As Integer 'local copy
Private mvarattDef As Integer 'local copy
Private mvarattMagDef As Integer 'local copy
Private mvarattSpd As Integer 'local copy
Private mvarattMAP As Integer 'local copy
Private mvarExperience As Long 'local copy
Private mvarattMagAtk As Integer 'local copy
'local variable(s) to hold property value(s)
Private mvarLevel As Integer 'local copy
'local variable(s) to hold property value(s)
Private mvarcurHP As Integer 'local copy
Private mvarcurMP As Integer 'local copy
'local variable(s) to hold property value(s)
Private mvarIndex As Integer 'local copy
'local variable(s) to hold property value(s)
Private mvarLocation As String 'local copy
Private mvarSubLocation As String 'local copy
Public Sub LocationCheck()
'Locations inside the first town are displayed below
'If IsInSquare(-492, -3712, -7901, -9522) Then
'    Location = "Riverbend Provision Shop"
'    BotMove -9038, -3502, 58
'End If
End Sub
```

Player.cls

Archive copy dated May 8, 2001 · Lines 55–104 of 605

```
Private Function IsInSquare(N, S, W, E)
If z > N Or z < S Or X < E Or X > W Then
    IsInSquare = False
    Exit Function
End If
IsInSquare = True
End Function

Public Property Let SubLocation(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.SubLocation = 5
    mvarSubLocation = vData
End Property


Public Property Get SubLocation() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.SubLocation
    SubLocation = mvarSubLocation
End Property



Public Property Let Location(ByVal vData As String)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Location = 5
    mvarLocation = vData
End Property


Public Property Get Location() As String
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Location
    Location = mvarLocation
End Property



Public Property Let Index(ByVal vData As Integer)
'used when assigning a value to the property, on the left side of an assignment.
'Syntax: X.Index = 5
    mvarIndex = vData
End Property
Public Sub ChatDelay(value1 As Single)
If value1 = 0 Then value1 = 0.001
If value1 = -1 Then value1 = ChatDelayValue
If value1 < 0 And value1 <> -1 Then value1 = 1
If value1 > 30 Then value1 = 1
AddToBuffer "delay" + Str$(value1)
End Sub
```

Player.cls

Archive copy dated May 8, 2001 · Lines 105–154 of 605

```
105
106 ' Public Sub ChatDelay(value1 As Integer)
107 'If value1 < 0 Then Exit Sub
108 'Form1.Timer1(Index).Interval = value1 * 1000
109 'End Sub
110
111 Public Sub SetMidi(midival As String)
112 midi = midival 'set midi and create object
113 If MidiNumber = 0 Then
114     midiX = Form1.aw(Index).AwObjectX = Avatars(Index).X
115     Form1.aw(Index).AwObjectY = Avatars(Index).y + 500
116     midiZ = Form1.aw(Index).AwObjectZ = Avatars(Index).z
117     Form1.aw(Index).AwObjectAction = "create solid off, visible off, noise " + midival
118     Form1.aw(Index).AwObjectModel = "midispk.rwx"
119     Form1.aw(Index).AwObjectAdd
120     MidiNumber = Form1.aw(Index).AwObjectNumber
121 Else
122     UpdateMidi midival
123 End If
124 End Sub
125
126 Public Sub UpdateMidi(Optional midival As String)
127 Form1.aw(Index).AwObjectOldX = midiX
128 Form1.aw(Index).AwObjectOldZ = midiZ
129 Form1.aw(Index).AwObjectOldNumber = MidiNumber
130 Form1.aw(Index).AwObjectX = X
131 Form1.aw(Index).AwObjectY = y + 500
132 Form1.aw(Index).AwObjectZ = z
133 If midival <> "" Then Form1.aw(Index).AwObjectAction = "create solid off, visible off,
     noise " + midival
134 Form1.aw(Index).AwObjectChange
135 MidiNumber = Form1.aw(Index).AwObjectNumber
136 End Sub
137
138 Public Sub NoSound()
139 SetMidi "silence.mid"
140 'midi = ""
141 End Sub
142
143 Private Sub DeleteSpeaker()
144 Form1.aw(Index).AwObjectNumber = MidiNumber
145 Form1.aw(Index).AwObjectX = midiX
146 Form1.aw(Index).AwObjectZ = midiZ
147 Form1.aw(Index).AwObjectDelete
148 End Sub
149
150 Public Property Get Index() As Integer
151 'used when retrieving value of a property, on the right side of an assignment.
152 'Syntax: Debug.Print X.Index
153     Index = mvarIndex
154 End Property
```

Player.cls

Archive copy dated May 8, 2001 · Lines 155–205 of 605

```
155
156
157
158 Public Property Let curMP(ByVal vData As Integer)
159 'used when assigning a value to the property, on the left side of an assignment.
160 'Syntax: X.curMP = 5
161     mvarcurMP = vData
162 End Property
163
164
165 Public Property Get curMP() As Integer
166 'used when retrieving value of a property, on the right side of an assignment.
167 'Syntax: Debug.Print X.curMP
168     curMP = mvarcurMP
169 End Property
170
171 Sub CheckLocation()
172
173 End Sub
174
175 Function Square(N, W, E, S)
176
177 End Function
178
179 Public Property Let curHP(ByVal vData As Integer)
180 'used when assigning a value to the property, on the left side of an assignment.
181 'Syntax: X.curHP = 5
182     mvarcurHP = vData
183 End Property
184
185
186 Public Property Get curHP() As Integer
187 'used when retrieving value of a property, on the right side of an assignment.
188 'Syntax: Debug.Print X.curHP
189     curHP = mvarcurHP
190 End Property
191
192
193
194 Public Property Let Level(ByVal vData As Integer)
195 'used when assigning a value to the property, on the left side of an assignment.
196 'Syntax: X.Level = 5
197     mvarLevel = vData
198 End Property
199
200
201 Public Property Get Level() As Integer
202 'used when retrieving value of a property, on the right side of an assignment.
203 'Syntax: Debug.Print X.Level
204     Level = mvarLevel
205 End Property
```

Player.cls

Archive copy dated May 8, 2001 · Lines 206–256 of 605

```
206
207
208
209 Public Sub ShowStatus()
210 ChatDelay 0
211 AddToBuffer "Race: " + Race + "  Level: " + its(Level)
212 AddToBuffer "HP: " + its(curHP) + "/" + its(HP) + "  MP: " + its(curMP) + "/" + its(MP)
213 AddToBuffer "Attack: " + its(attAtk) + "  Magic Attack: " + its(attMagAtk)
214 AddToBuffer "Defense: " + its(attDef) + "  Magic Attack: " + its(attMagDef)
215 AddToBuffer "Speed: " + its(attSpd) + "  Magic Points: " + its(attMAP)
216 AddToBuffer "Experience: " + its(Experience)
217 ChatDelay -1
218 End Sub
219
220 Public Property Let attMagAtk(ByVal vData As Integer)
221 'used when assigning a value to the property, on the left side of an assignment.
222 'Syntax: X.attMagAtk = 5
223     mvarattMagAtk = vData
224 End Property
225
226
227 Public Property Get attMagAtk() As Integer
228 'used when retrieving value of a property, on the right side of an assignment.
229 'Syntax: Debug.Print X.attMagAtk
230     attMagAtk = mvarattMagAtk
231 End Property
232
233
234
235 Public Property Let Experience(ByVal vData As Long)
236 'used when assigning a value to the property, on the left side of an assignment.
237 'Syntax: X.Experience = 5
238     mvarExperience = vData
239 End Property
240
241
242 Public Property Get Experience() As Long
243 'used when retrieving value of a property, on the right side of an assignment.
244 'Syntax: Debug.Print X.Experience
245     Experience = mvarExperience
246 End Property
247
248
249
250 Public Property Let attMAP(ByVal vData As Integer)
251 'used when assigning a value to the property, on the left side of an assignment.
252 'Syntax: X.attMAP = 5
253     mvarattMAP = vData
254 End Property
255
256
```

Player.cls

Archive copy dated May 8, 2001 · Lines 257–306 of 605

```
257 Public Property Get attMAP() As Integer
258 'used when retrieving value of a property, on the right side of an assignment.
259 'Syntax: Debug.Print X.attMAP
260     attMAP = mvarattMAP
261 End Property
262
263
264
265 Public Property Let attSpd(ByVal vData As Integer)
266 'used when assigning a value to the property, on the left side of an assignment.
267 'Syntax: X.attSpd = 5
268     mvarattSpd = vData
269 End Property
270
271
272 Public Property Get attSpd() As Integer
273 'used when retrieving value of a property, on the right side of an assignment.
274 'Syntax: Debug.Print X.attSpd
275     attSpd = mvarattSpd
276 End Property
277
278
279
280 Public Property Let attMagDef(ByVal vData As Integer)
281 'used when assigning a value to the property, on the left side of an assignment.
282 'Syntax: X.attMagDef = 5
283     mvarattMagDef = vData
284 End Property
285
286
287 Public Property Get attMagDef() As Integer
288 'used when retrieving value of a property, on the right side of an assignment.
289 'Syntax: Debug.Print X.attMagDef
290     attMagDef = mvarattMagDef
291 End Property
292
293
294
295 Public Property Let attDef(ByVal vData As Integer)
296 'used when assigning a value to the property, on the left side of an assignment.
297 'Syntax: X.attDef = 5
298     mvarattDef = vData
299 End Property
300
301
302 Public Property Get attDef() As Integer
303 'used when retrieving value of a property, on the right side of an assignment.
304 'Syntax: Debug.Print X.attDef
305     attDef = mvarattDef
306 End Property
```

Player.cls

Archive copy dated May 8, 2001 · Lines 307–357 of 605

```
307
308
309
310 Public Property Let attAtk(ByVal vData As Integer)
311 'used when assigning a value to the property, on the left side of an assignment.
312 'Syntax: X.attAtk = 5
313     mvarattAtk = vData
314 End Property
315
316
317 Public Property Get attAtk() As Integer
318 'used when retrieving value of a property, on the right side of an assignment.
319 'Syntax: Debug.Print X.attAtk
320     attAtk = mvarattAtk
321 End Property
322
323 Public Sub CureAll()
324 curHP = HP
325 curMP = MP
326 End Sub
327
328 'To fire this event, use RaiseEvent with the following syntax:
329 'RaiseEvent QuestionRespond[(arg1, arg2, ... , argn)]
330 Public Function QuestionRespond(message As String, key As Variant)
331 On Error GoTo ErrorHandler
332 Dim answers() As String
333 Dim workvar As Single
334 Select Case key
335 Case "race select"
336     Select Case message
337     Case "human"
338         AddToBuffer "You have selected to be a human."
339         AddToBuffer "Your statistics are being randomly generated - please wait..."
340         HP = 250 + Int(Rnd * 50)
341         MP = 25 + Int(Rnd * 10)
342         Race = "Human"
343         'set human characteristics
344         attMagAtk = 20 + Int(Rnd * 5)
345         attDef = 30 + Int(Rnd * 5)
346         attMagDef = 20 + Int(Rnd * 5)
347         attAtk = 35 + Int(Rnd * 5)
348         attSpd = 20 + Int(Rnd * 5)
349         attMAP = 0
350         Experience = 0
351         Level = 1
352         CureAll
353         ShowStatus
354         Status = "storytelling"
355     Case "elf"
356         AddToBuffer "You have selected to be an elf."
357         AddToBuffer "Your statistics are being randomly generated - please wait..."
```

Player.cls

Archive copy dated May 8, 2001 · Lines 358–403 of 605

```
358          HP = 200 + Int(Rnd * 50)
359          MP = 35 + Int(Rnd * 10)
360          Race = "Elf"
361          attMagAtk = 35 + Int(Rnd * 5)
362          attDef = 20 + Int(Rnd * 5)
363          attMagDef = 30 + Int(Rnd * 5)
364          attAtk = 20 + Int(Rnd * 5)
365          attSpd = 30 + Int(Rnd * 5)
366          attMAP = 0
367          Experience = 0
368          Level = 1
369          CureAll
370          ShowStatus
371          Status = "storytelling"
372      End Select
373      SetMidi "title.mp3"
374      QuestionRespond = True
375  Case "chatdelay1"
376      workvar = Val(message$)
377      'MsgBox Description$ + Err.Description
378      If workvar <= 0 Or workvar > 30 Then QuestionRespond = False: Exit Function
379      AddToBuffer "The chat delay has been set to" + Str$(workvar) + " seconds.  The
     game will now begin - please wait...."
380      ChatDelayValue = workvar
381      ChatDelay 15
382      ChatDelay workvar
383      Select Case Race
384      Case "Human"
385          AddToBuffer "The Valyrian Kingdon has existed for thousands of centuries."
386          AddToBuffer "Through war, famine, plague, storms, and against every obstacle
     imaginable, the kingdom has survived, and even prospered."
387          AddToBuffer "It has been thirty years since the War of the East, when
     thousands were killed in a battle between the Silvestri and the Mueller Houses."
388          AddToBuffer "Since then, peace has returned, and prosperity has reigned over
     the kingdom."
389          AddToBuffer Avatars(Index).avname + ", a new knight of the Royal Guard of
     House Silvestri, has arrived in Riverbend, the capital of the kingdom."
390          AddToBuffer "Little did he know the tensions that were looming just under the
     surface..."
391          ChatDelay 0
392          AddToBuffer "/status town"
393          AddToBuffer "/music town.mp3"
394          ChatDelay ChatDelayValue
395      Case "Elf"
396      End Select
397      QuestionRespond = True
398  End Select
399  Exit Function
400  ErrorHandler:
401      Description$ = Err.Description
402      Resume Next
403  End Function
```

Player.cls

Archive copy dated May 8, 2001 · Lines 404–453 of 605

```
'local variable(s) to hold property value(s)


Public Sub UpdateLocation()
Form1.aw(Index).AwMyX = Avatars(Index).X
Form1.aw(Index).AwMyY = Avatars(Index).y + 5000
Form1.aw(Index).AwMyZ = Avatars(Index).z
Form1.aw(Index).AwStateChange
End Sub

Public Property Get Questions() As Integer
'used when retrieving value of a property, on the right side of an assignment.
'Syntax: Debug.Print X.Questions
    Questions = UBound(questionbuffer, 1)
End Property



Public Sub CheckQuestionResponse(message$)
Dim u As Boolean
If UBound(questionbuffer(Questions).answers, 1) = 0 Then
     u = QuestionRespond(message$, questionbuffer(Questions).key)
     key1 = questionbuffer(Questions).key
     If u = True Then RemoveQuestion
     AnotherQuestion key1
     Exit Sub
End If
For countvar = 1 To UBound(questionbuffer(Questions).answers, 1)
    If LCase$(message$) = questionbuffer(Questions).answers(countvar) Then
        u = QuestionRespond(message$, questionbuffer(Questions).key)
        key1 = questionbuffer(Questions).key
        If u = True Then RemoveQuestion
        AnotherQuestion key1
        Exit Sub
    End If
Next
End Sub

Private Sub AnotherQuestion(key As Variant)
Dim answers() As String
Select Case key
Case "race select"
    ReDim answers(0)
    AskQuestion answers, "chatdelay1", "Please enter a comfortable whisper delay (in
 seconds) for you to read the story."
Case "chatdelay1"
End Select
End Sub
Public Sub RemoveQuestion()
ReDim Preserve questionbuffer(UBound(questionbuffer, 1) - 1)
If UBound(questionbuffer, 1) >= 1 Then
```

Player.cls

Archive copy dated May 8, 2001 · Lines 454–503 of 605

```
454      key = 1
455  End If
456  End Sub
457
458  Public Sub AskQuestion(answers() As String, key As Variant, text As String)
459  Dim ques As Question
460  ques.answers = answers
461  ques.key = key
462  ques.text = text
463  ReDim Preserve questionbuffer(UBound(questionbuffer, 1) + 1)
464  questionbuffer(UBound(questionbuffer, 1)) = ques
465  AddToBuffer ques.text
466  End Sub
467
468  'local variable(s) to hold property value(s)
469  'Private mvarBufferLength As Integer 'local copy
470  Public Sub ClearBufferLine()
471  ReDim Preserve mvarchatbuffer(UBound(mvarchatbuffer) - 1)
472  End Sub
473
474
475  Public Property Get BufferLength() As Integer
476  'used when retrieving value of a property, on the right side of an assignment.
477  'Syntax: Debug.Print X.BufferLength
478      BufferLength = UBound(mvarchatbuffer, 1)
479  End Property
480
481
482
483  Public Sub ClearBuffer()
484  ReDim chatbuffer(0)
485  End Sub
486
487  Public Sub AddToBuffer(message As String)
488  ReDim Preserve mvarchatbuffer(UBound(mvarchatbuffer, 1) + 1)
489  mvarchatbuffer(UBound(mvarchatbuffer, 1)) = message
490  End Sub
491
492  Public Property Let avIndex(ByVal vData As Integer)
493  'used when assigning a value to the property, on the left side of an assignment.
494  'Syntax: X.avIndex = 5
495      mvaravIndex = vData
496  End Property
497
498
499  Public Property Get avIndex() As Integer
500  'used when retrieving value of a property, on the right side of an assignment.
501  'Syntax: Debug.Print X.avIndex
502      avIndex = mvaravIndex
503  End Property
```

Player.cls

Archive copy dated May 8, 2001 · Lines 504–556 of 605

```
504
505
506
507 Public Property Let chatbuffer(ByVal vnum As Integer, ByVal vData As String)
508 'used when assigning a value to the property, on the left side of an assignment.
509 'Syntax: X.Battle = 5
510     If vnum > UBound(mvarchatbuffer, 1) Then ReDim Preserve mvarchatbuffer(vnum)
511     mvarchatbuffer(vnum) = vData
512 End Property
513
514
515 Public Property Get chatbuffer(ByVal vnum As Integer) As String
516 'used when retrieving value of a property, on the right side of an assignment.
517 'Syntax: Debug.Print X.Battle
518     chatbuffer = mvarchatbuffer(vnum)
519 End Property
520
521
522 Public Property Let Battle(ByVal vData As Boolean)
523 'used when assigning a value to the property, on the left side of an assignment.
524 'Syntax: X.Battle = 5
525     mvarBattle = vData
526 End Property
527
528
529 Public Property Get Battle() As Boolean
530 'used when retrieving value of a property, on the right side of an assignment.
531 'Syntax: Debug.Print X.Battle
532     Battle = mvarBattle
533 End Property
534
535
536
537 Public Property Let Status(ByVal vData As String)
538 'used when assigning a value to the property, on the left side of an assignment.
539 'Syntax: X.Status = 5
540     mvarStatus = vData
541 End Property
542
543
544 Public Property Get Status() As String
545 'used when retrieving value of a property, on the right side of an assignment.
546 'Syntax: Debug.Print X.Status
547     Status = mvarStatus
548 End Property
549
550
551
552 Public Property Let Race(ByVal vData As String)
553 'used when assigning a value to the property, on the left side of an assignment.
554 'Syntax: X.Race = 5
555     mvarRace = vData
556 End Property
```

Player.cls

Archive copy dated May 8, 2001 · Lines 557–605 of 605

```
557
558
559  Public Property Get Race() As String
560  'used when retrieving value of a property, on the right side of an assignment.
561  'Syntax: Debug.Print X.Race
562      Race = mvarRace
563  End Property
564
565
566
567  Public Property Let MP(ByVal vData As Integer)
568  'used when assigning a value to the property, on the left side of an assignment.
569  'Syntax: X.MP = 5
570      mvarMP = vData
571  End Property
572
573
574  Public Property Get MP() As Integer
575  'used when retrieving value of a property, on the right side of an assignment.
576  'Syntax: Debug.Print X.MP
577      MP = mvarMP
578  End Property
579
580
581
582  Public Property Let HP(ByVal vData As Integer)
583  'used when assigning a value to the property, on the left side of an assignment.
584  'Syntax: X.HP = 5
585      mvarHP = vData
586  End Property
587
588
589  Public Property Get HP() As Integer
590  'used when retrieving value of a property, on the right side of an assignment.
591  'Syntax: Debug.Print X.HP
592      HP = mvarHP
593  End Property
594
595
596
597  Private Sub Class_Initialize()
598  ReDim mvarchatbuffer(0)
599  ReDim questionbuffer(0)
600  ChatDelayValue = 1
601  End Sub
602
603  Private Sub Class_Terminate()
604  DeleteSpeaker
605  End Sub
```

Players.cls

Archive copy dated April 12, 2001 · Lines 1–54 of 107

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
  Persistable = 0  'NotPersistable
  DataBindingBehavior = 0  'vbNone
  DataSourceBehavior  = 0  'vbNone
  MTSTransactionMode  = 0  'NotAnMTSObject
END
Attribute VB_Name = "Players"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = False
Attribute VB_Ext_KEY = "SavedWithClassBuilder6" ,"Yes"
Attribute VB_Ext_KEY = "Collection" ,"Player"
Attribute VB_Ext_KEY = "Member0" ,"Player"
Attribute VB_Ext_KEY = "Top_Level" ,"Yes"
'local variable to hold collection
Private mCol As Collection
'Public Function SearchForNumber(Num1 As Integer)
'Dim u As Integer
'For u = 1 To Players.Count
'    If Players.Item(u).avIndex = Num1 Then
'        SearchForNumber = u
'        Exit Function
'    End If
'Next
'SearchForNumber = -1
'End Function



Public Function Add(Optional sKey As String) As Player
    'create a new object
    Dim objNewMember As Player
    Set objNewMember = New Player

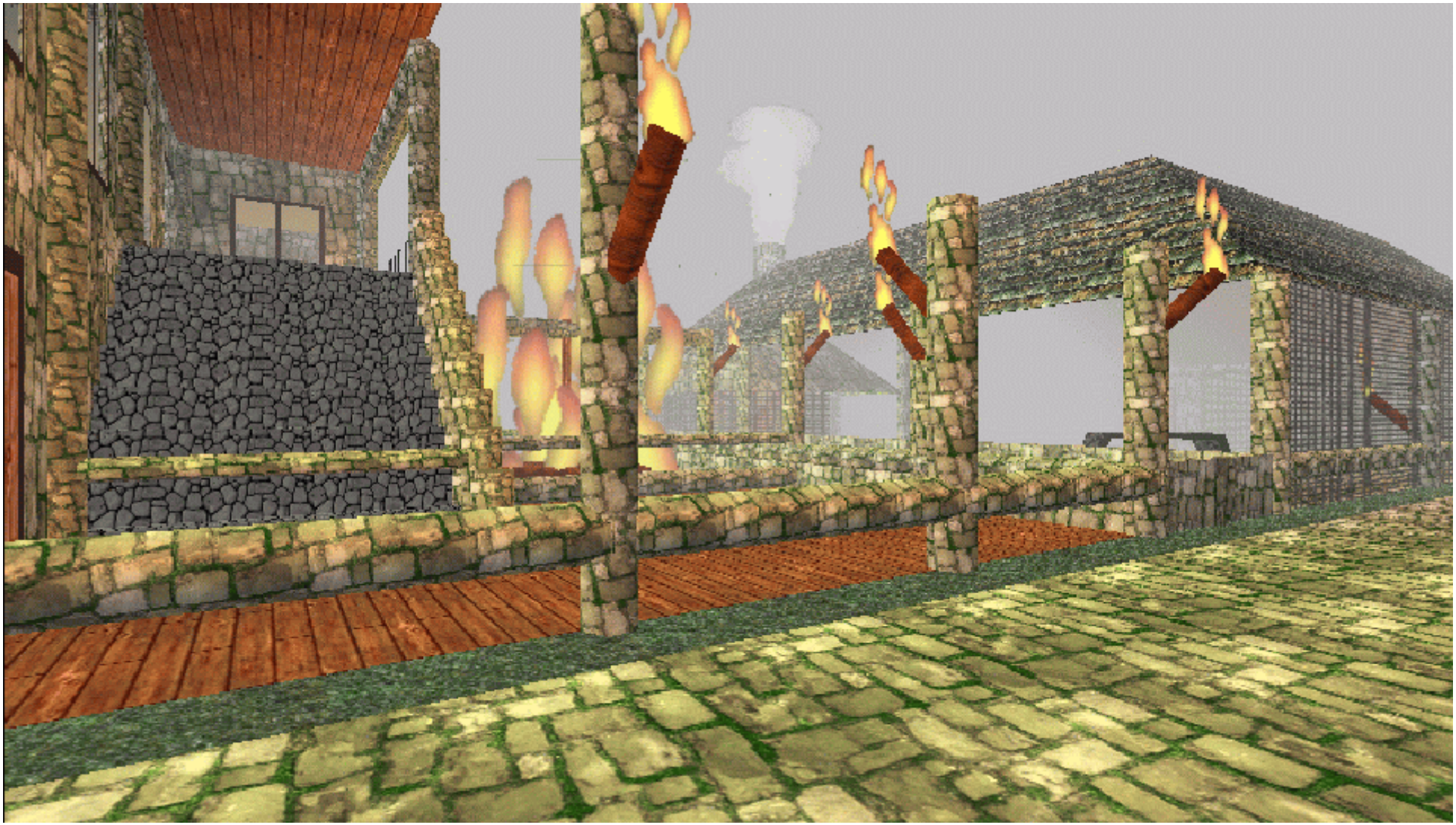

    'set the properties passed into the method
    'objNewMember.HP = HP
    'objNewMember.MP = MP
    'objNewMember.Race = Race
    'objNewMember.Status = Status
    'objNewMember.Battle = Battle
    'objNewMember.avIndex = avIndex
    If Len(sKey) = 0 Then
        mCol.Add objNewMember
    Else
        mCol.Add objNewMember, sKey
    End If


    'return the object created
    Set Add = objNewMember
```

Players.cls

Archive copy dated April 12, 2001 · Lines 55–107 of 107

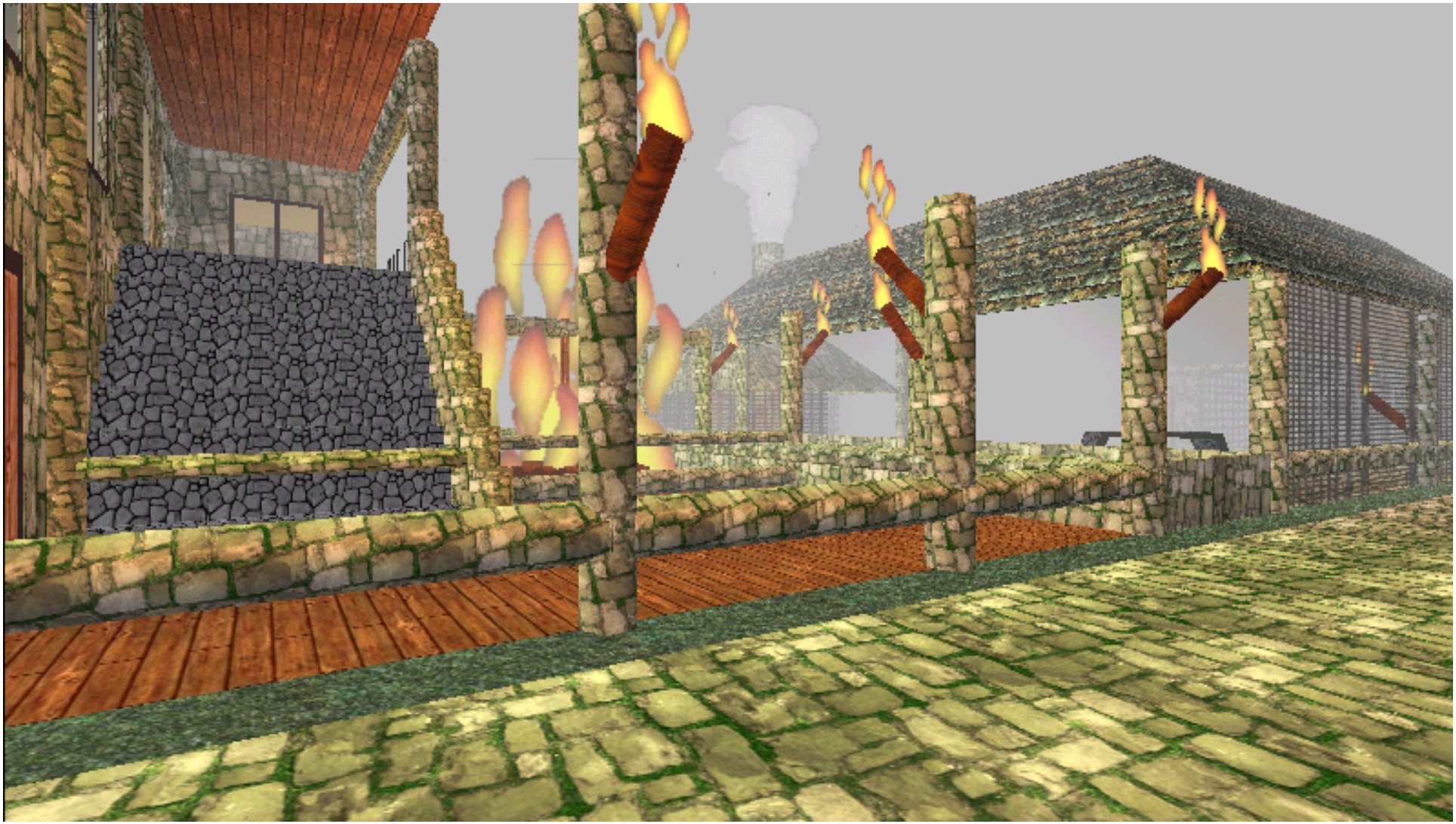
```
55      Set objNewMember = Nothing
56
57
58  End Function
59
60  Public Property Get Item(vntIndexKey As Variant) As Player
61  Attribute Item.VB_UserMemId = 0
62      'used when referencing an element in the collection
63      'vntIndexKey contains either the Index or Key to the collection,
64      'this is why it is declared as a Variant
65      'Syntax: Set foo = x.Item(xyz) or Set foo = x.Item(5)
66    Set Item = mCol(vntIndexKey)
67  End Property
68
69
70  Public Property Get Count() As Long
71      'used when retrieving the number of elements in the
72      'collection. Syntax: Debug.Print x.Count
73      Count = mCol.Count
74  End Property
75
76
77  Public Sub Remove(vntIndexKey As Variant)
78      'used when removing an element from the collection
79      'vntIndexKey contains either the Index or Key, which is why
80      'it is declared as a Variant
81      'Syntax: x.Remove(xyz)
82
83
84      mCol.Remove vntIndexKey
85  End Sub
86
87
88  Public Property Get NewEnum() As IUnknown
89  Attribute NewEnum.VB_UserMemId = -4
90  Attribute NewEnum.VB_MemberFlags = "40"
91      'this property allows you to enumerate
92      'this collection with the For...Each syntax
93      Set NewEnum = mCol.[_NewEnum]
94  End Property
95
96
97  Private Sub Class_Initialize()
98      'creates the collection when this class is created
99      Set mCol = New Collection
100 End Sub
101
102
103 Private Sub Class_Terminate()
104     'destroys collection when this class is terminated
105     Set mCol = Nothing
106 End Sub
107
```





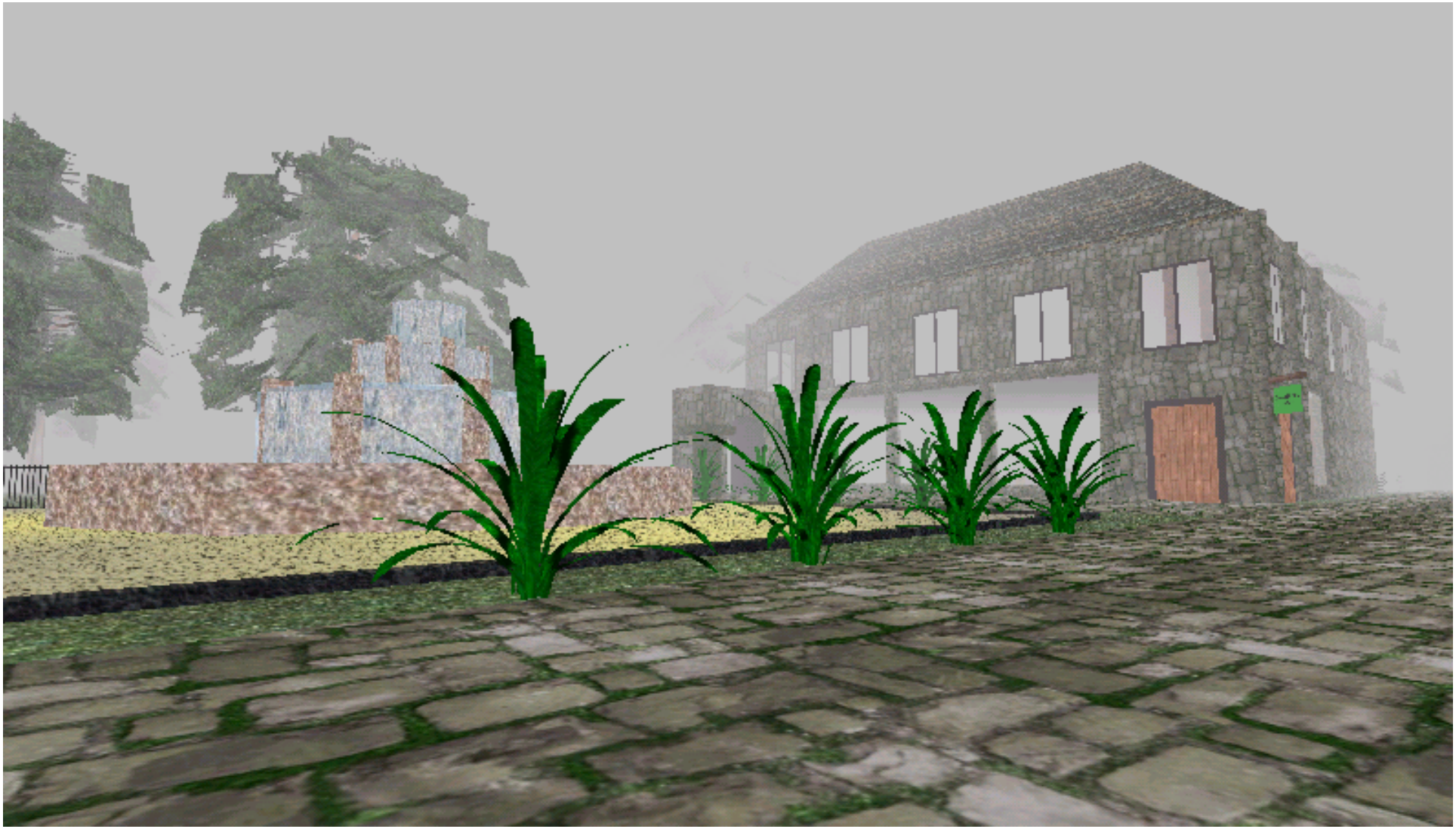
A lake in the mountains

gulakon031201.bmp



Torches along the stone arcade

gulakon040201.bmp



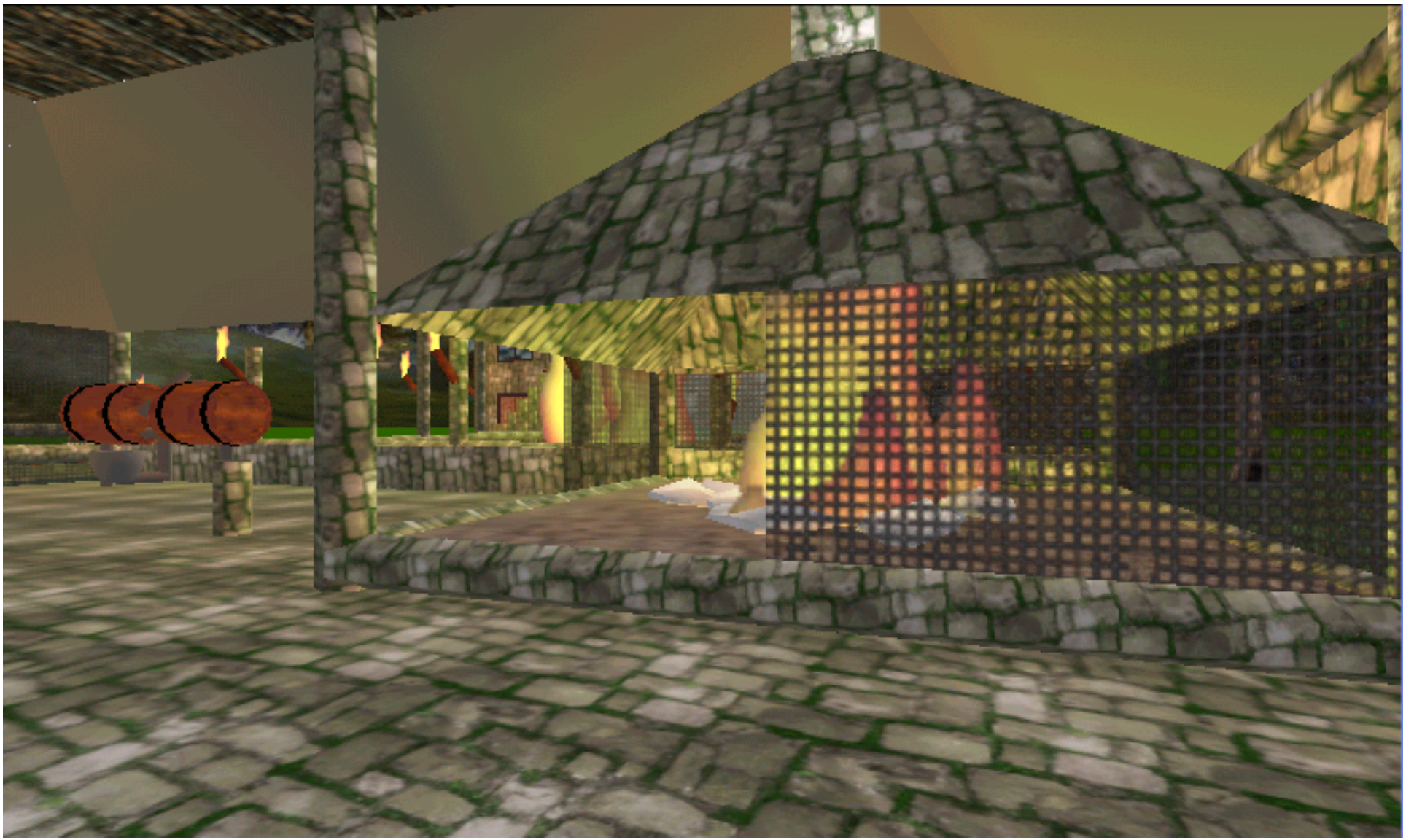
A misty street with unfinished buildings

gulakon40201.bmp



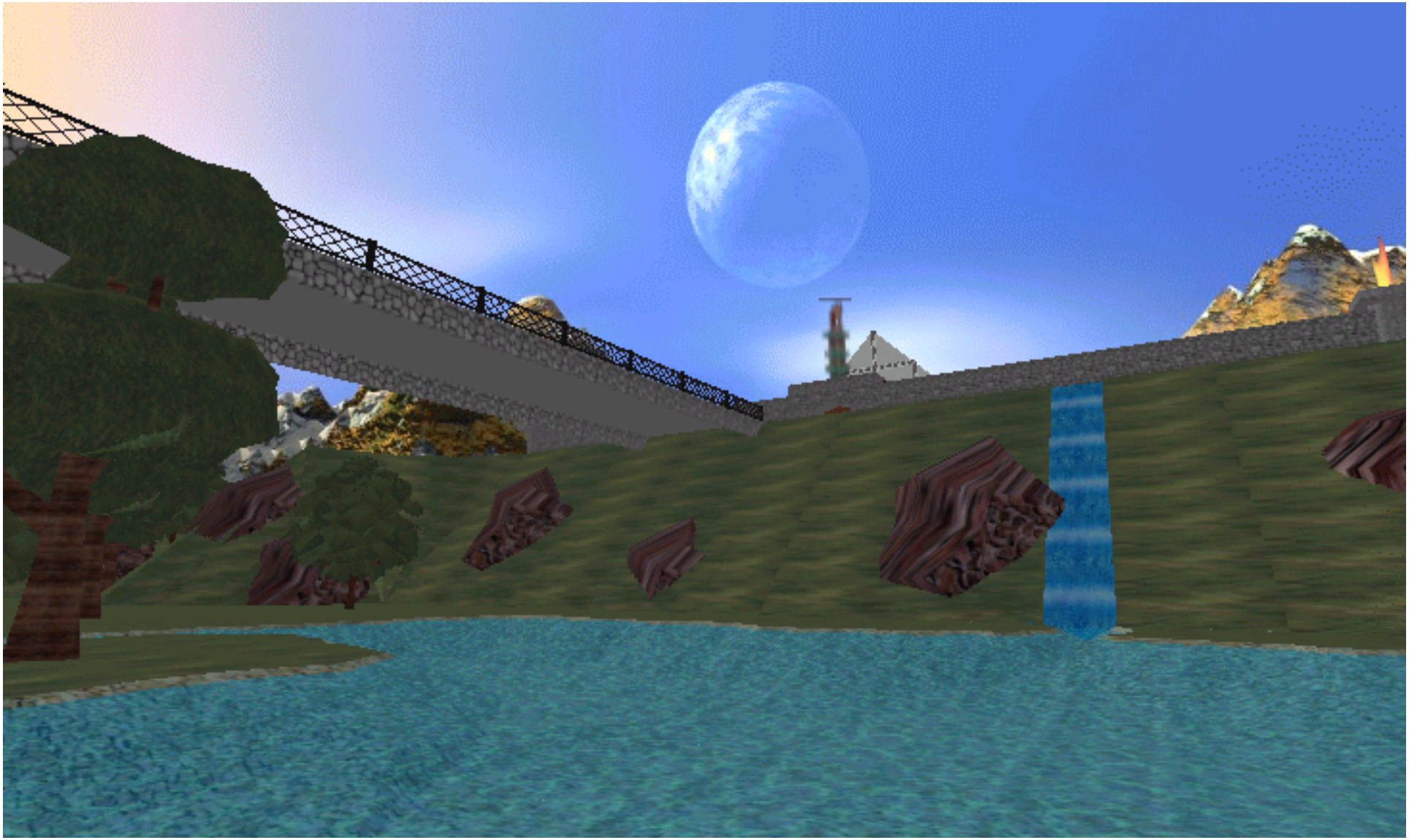
Red chairs around a stone table

gulakon040201-3.bmp



An open-sided pavilion on a cobblestone square

gulakon041601.bmp



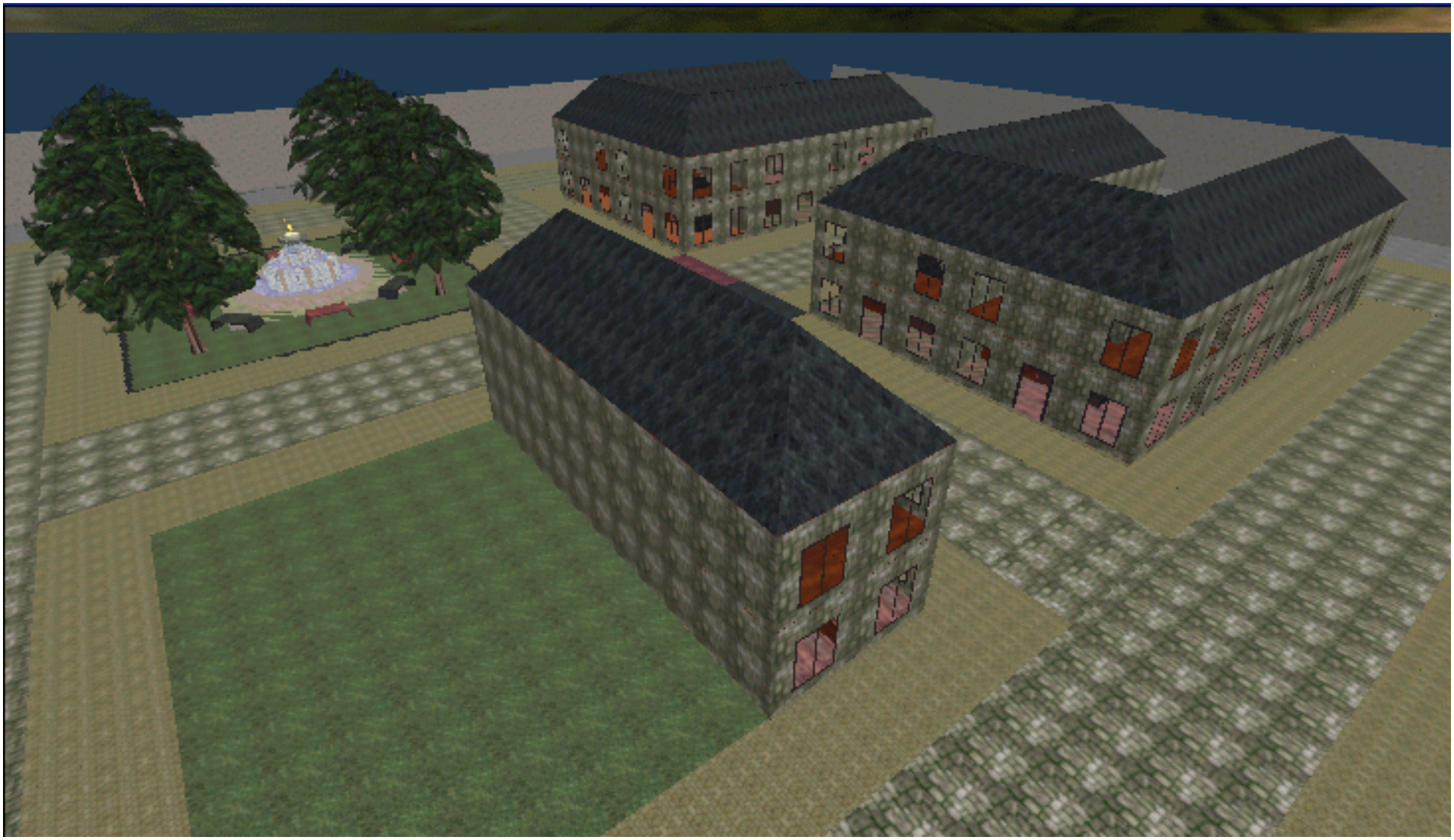
A waterfall and bridge beneath the moon

gulakon041601-2.bmp



Timber posts inside a stone hall

gulakon042101-10.bmp



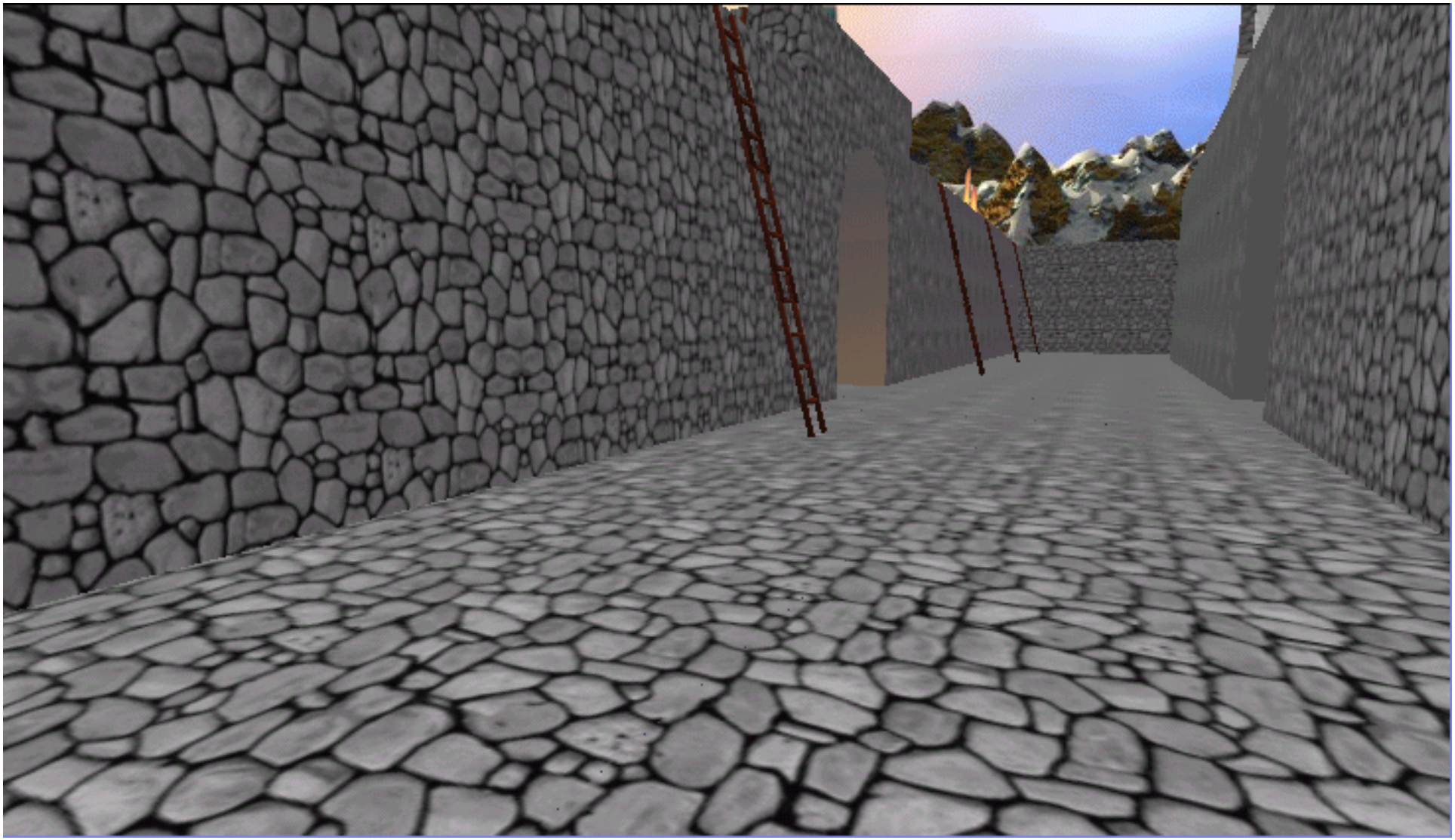
Stone buildings seen from above

gulakon042401.bmp



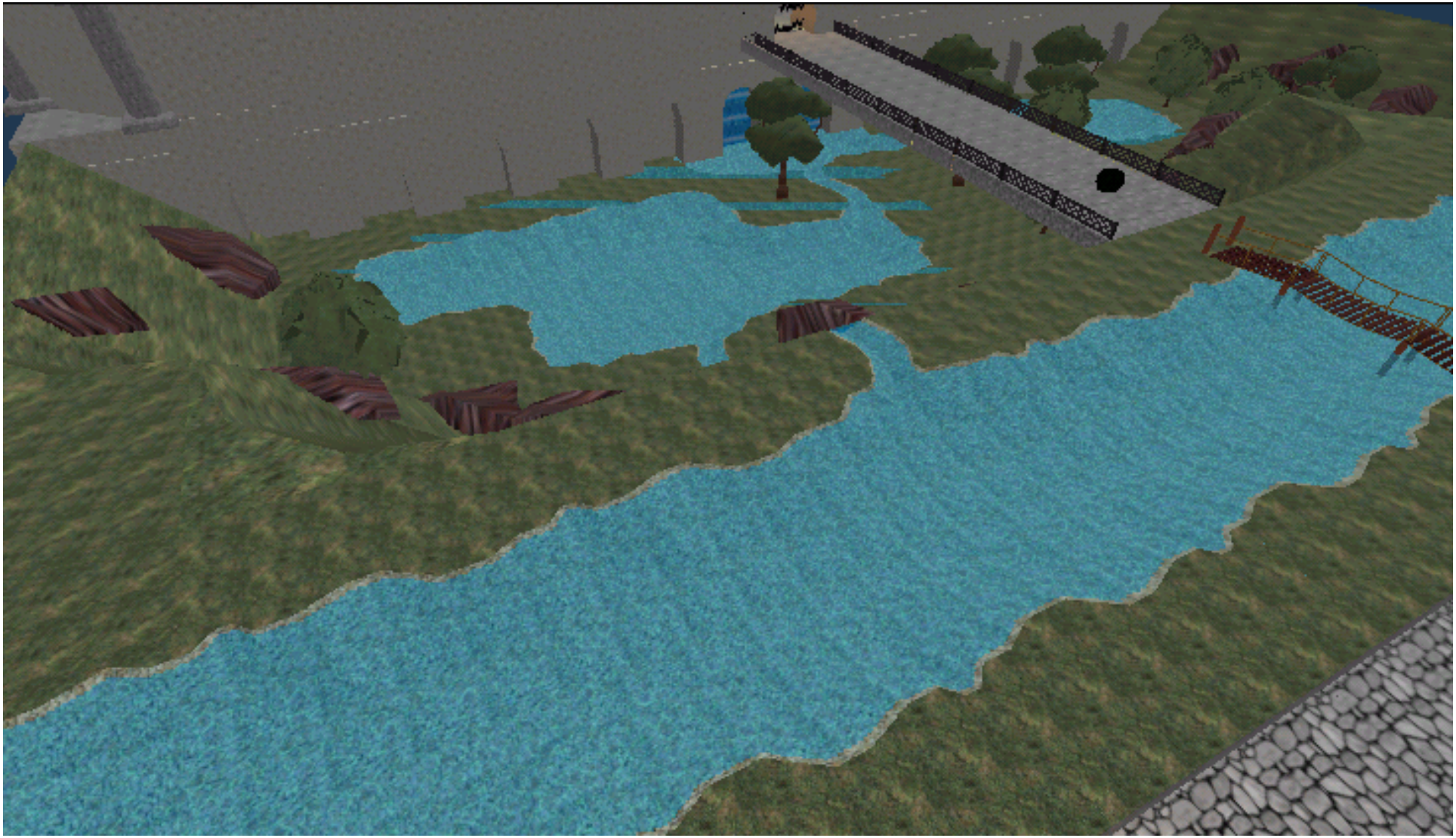
A fire beneath a tall window

gulakon42401-2.bmp



Ladders along a stone passage

gulakon42401-3.bmp



Waterways and a bridge seen from above

gulakon042401-4.bmp



A fountain between the trees

gulakon042401-5.bmp



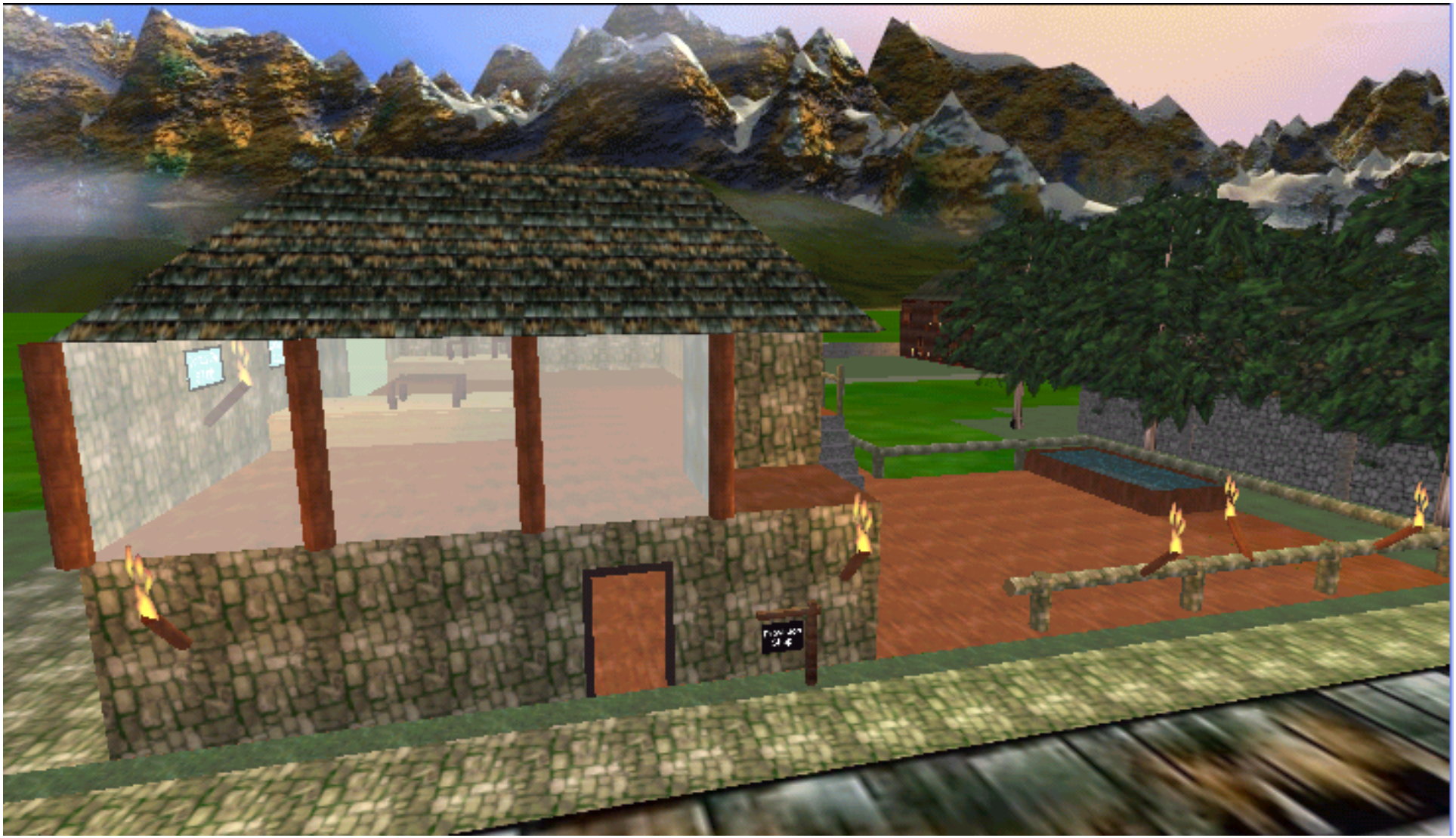
Shopfronts along a stone street

gulakon042401-6.bmp



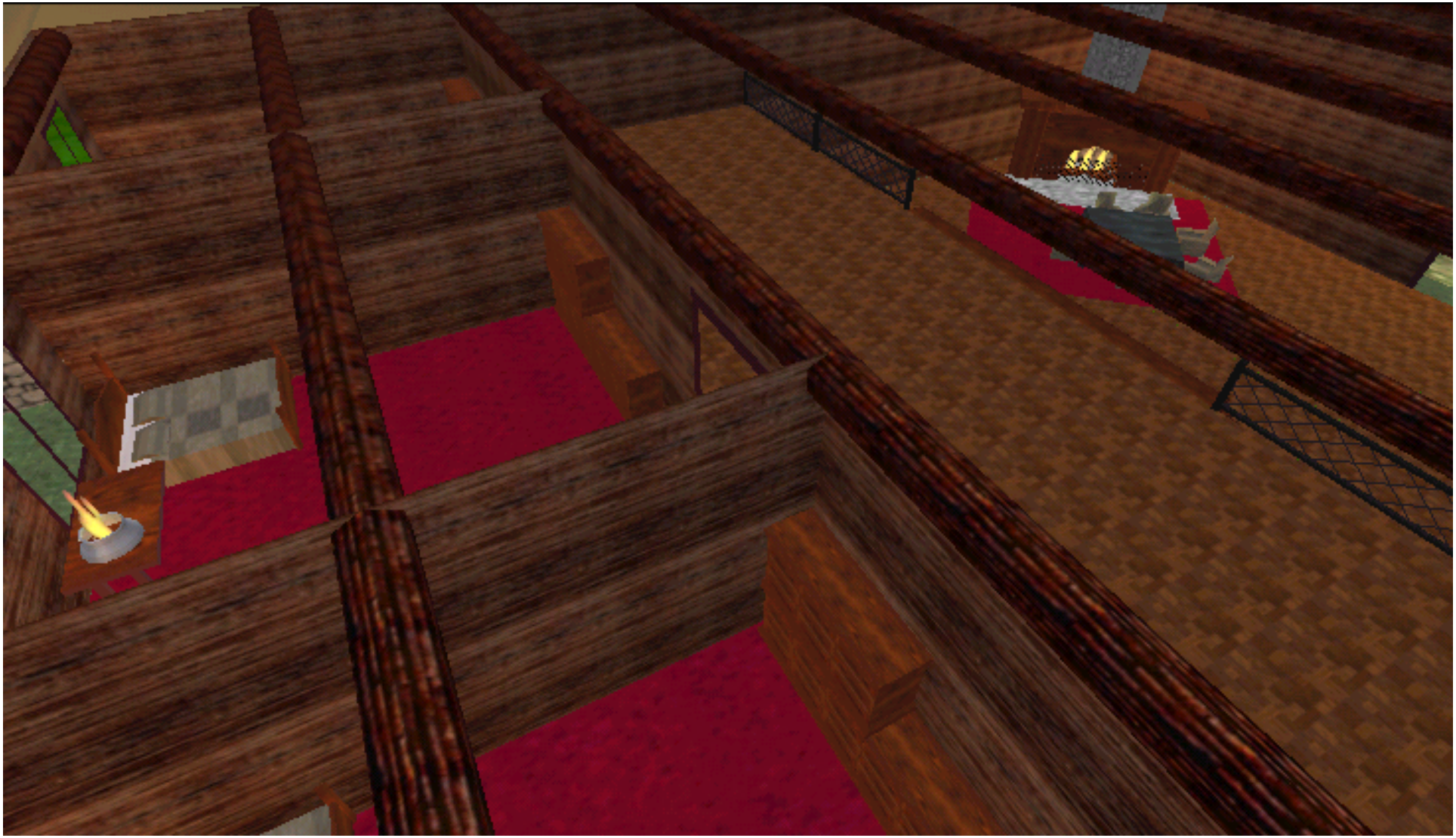
Evergreens beside a path

gulakon042401-7.bmp



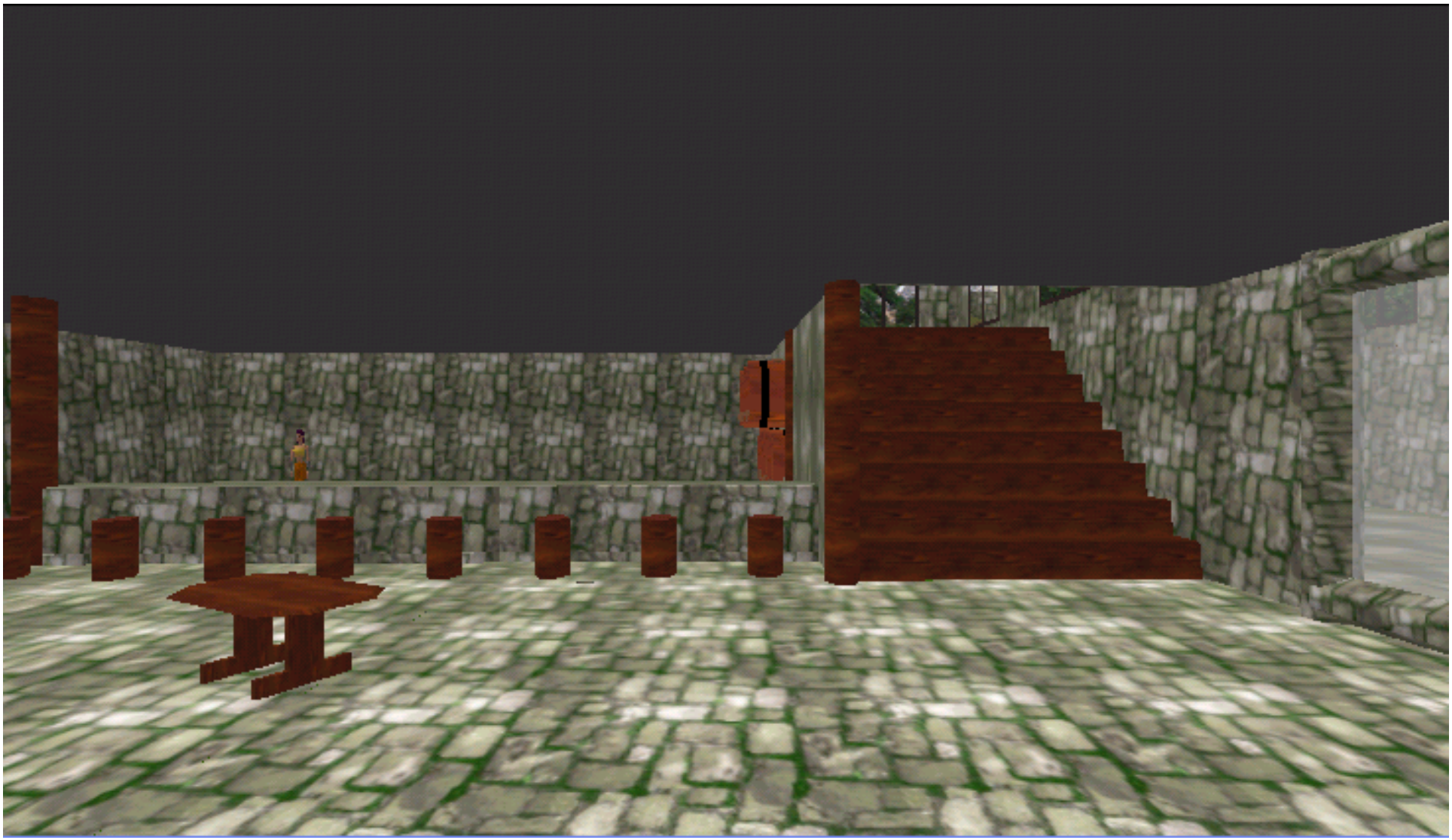
A timber-framed house beside a walled yard

gulakon042401-8.bmp



A wooden balcony above a room with red chairs

gulakon042401-9.bmp



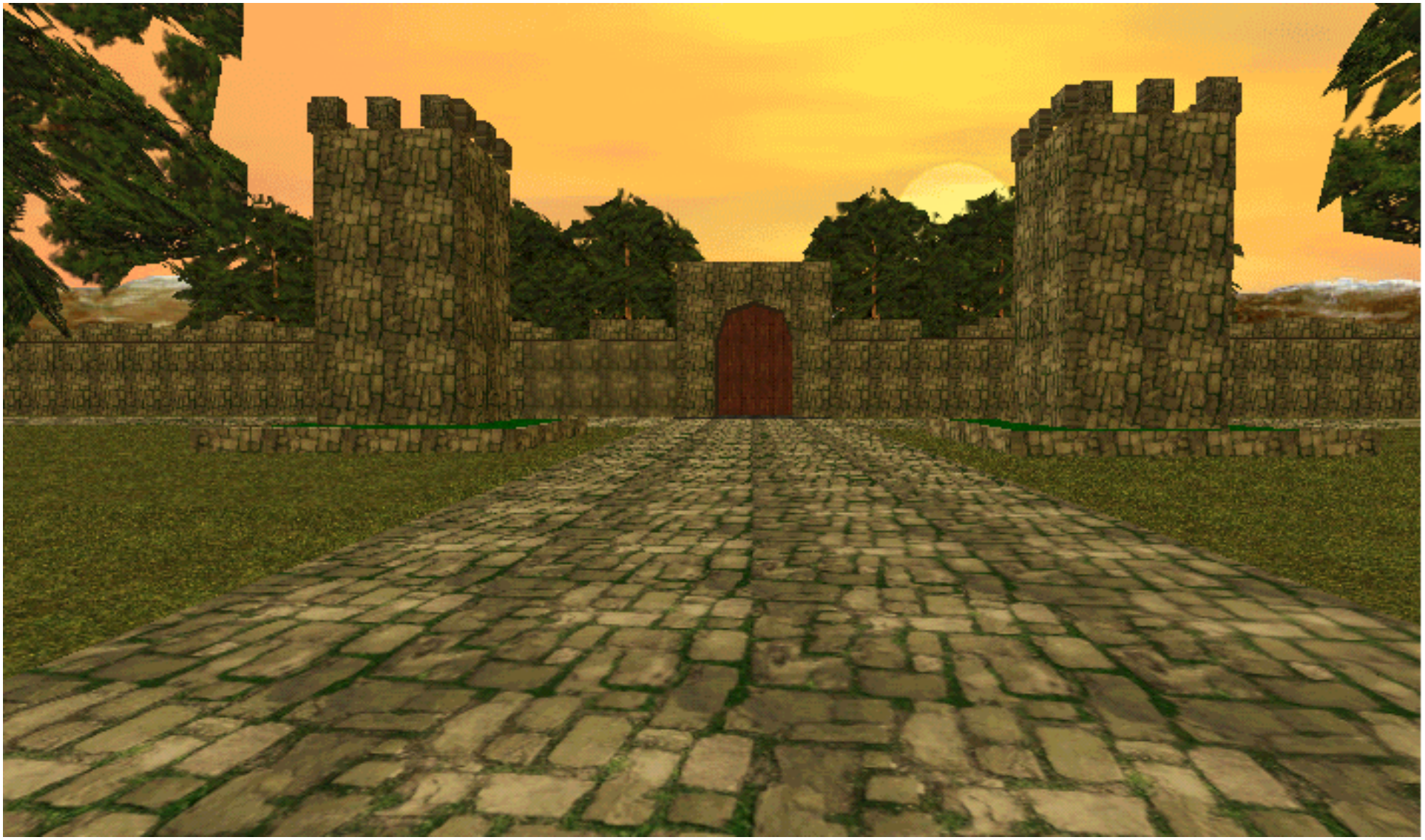
A stone chamber with a staircase

gulakon042401-11.bmp



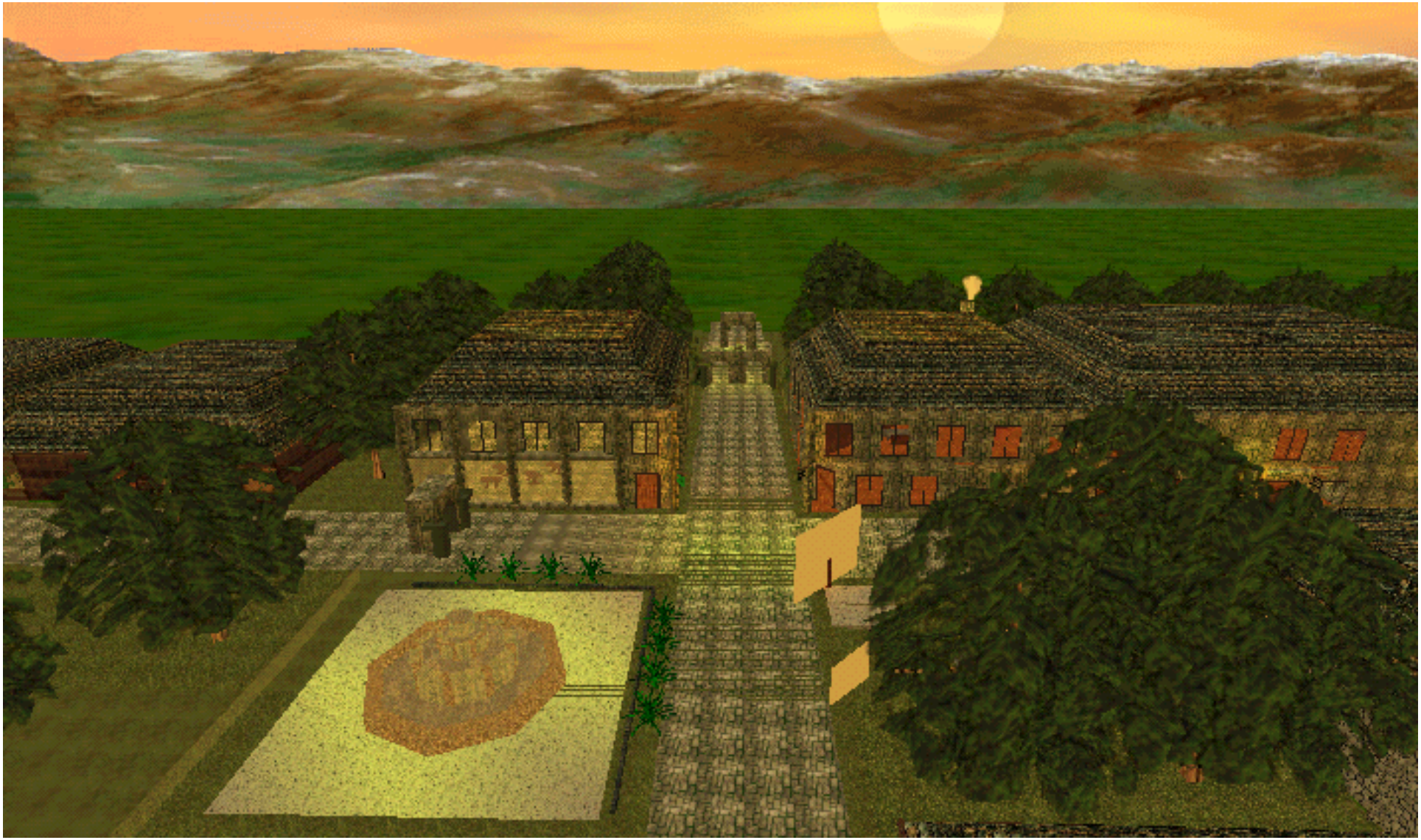
Benches facing three arched doorways

gulakon042501.bmp



A castle gate at sunset

gulakon042501-2.bmp



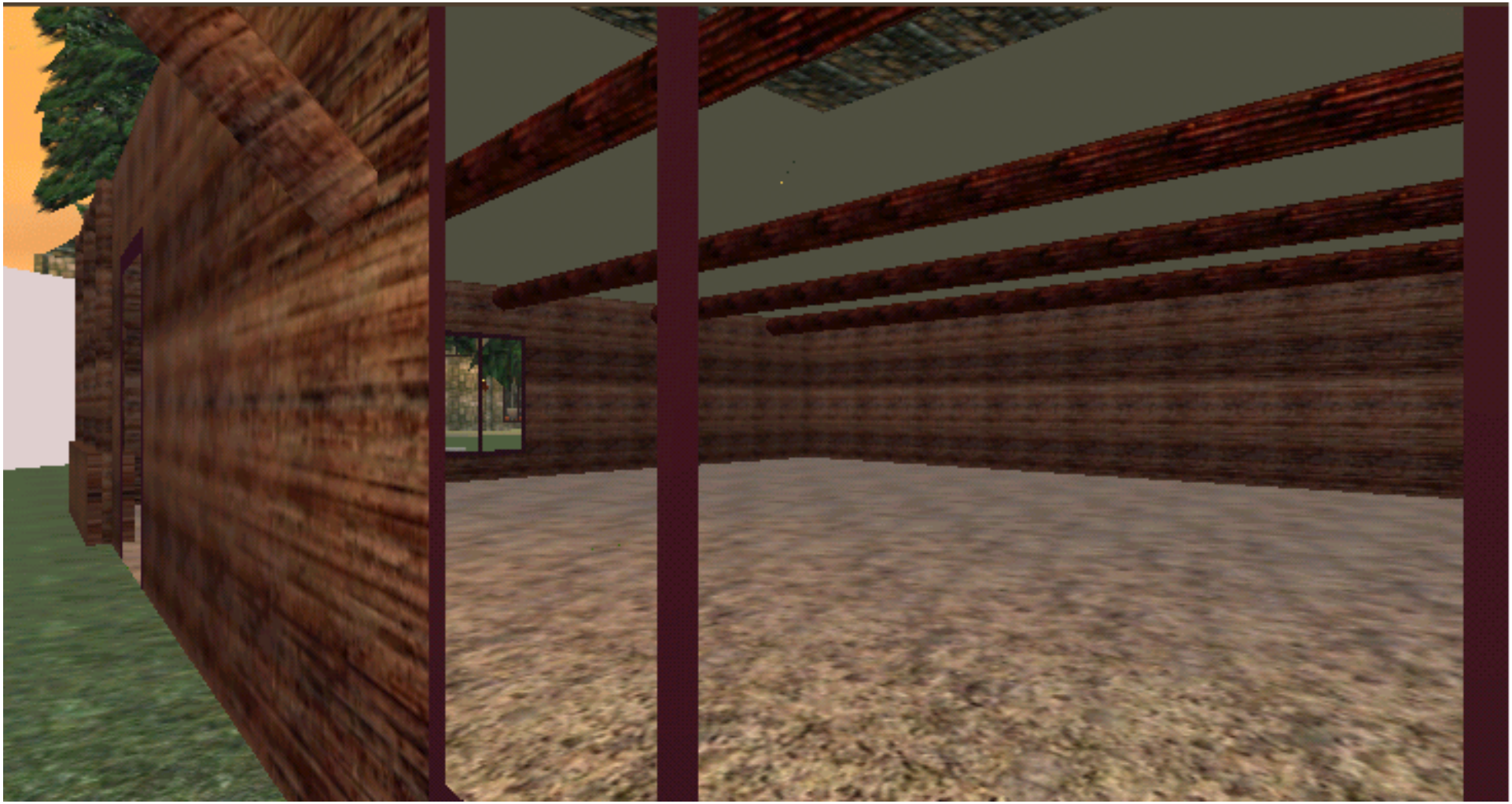
Rooftops and a square in the evening light

gulakon042501-3.bmp



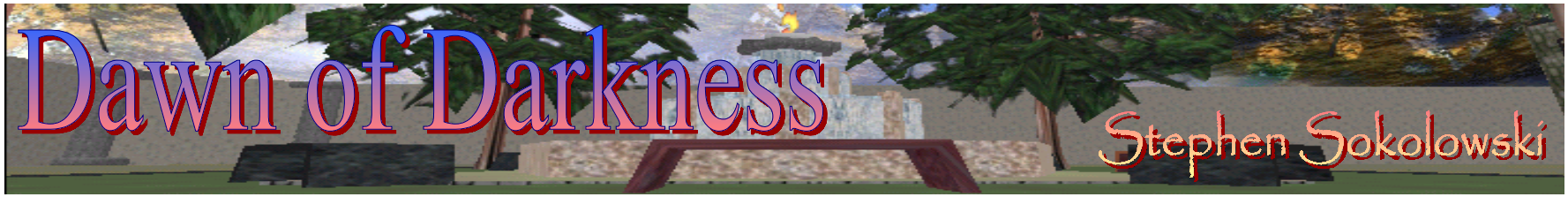
A central aisle between rows of pews

gulakon042801.bmp



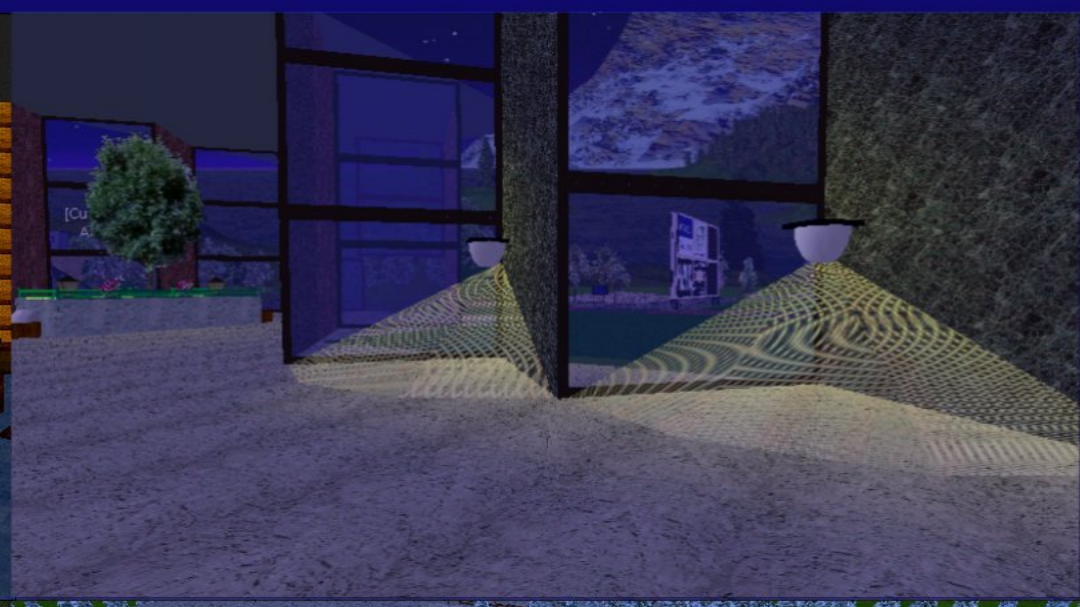
An open timber-framed interior

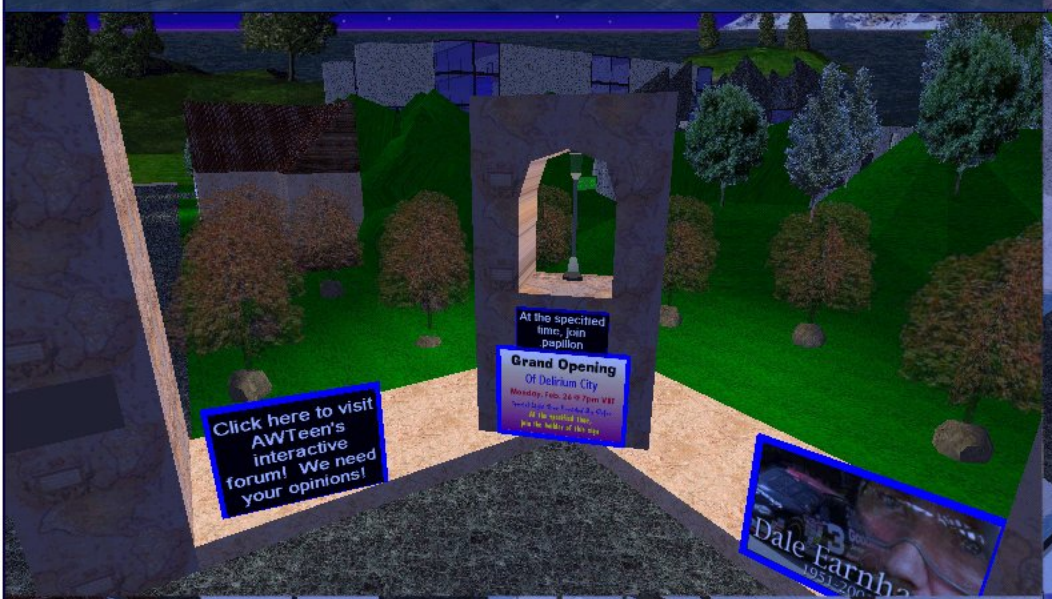
gulakon042801-2.bmp

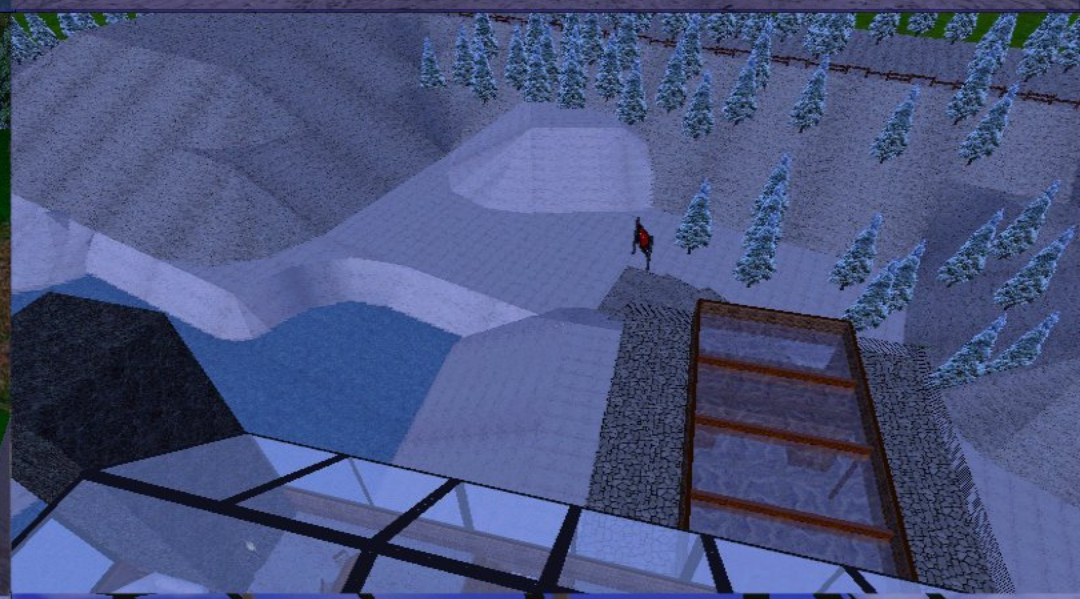

Dawn of Darkness
Stephen Sokolowski

three dimensional role playing environment

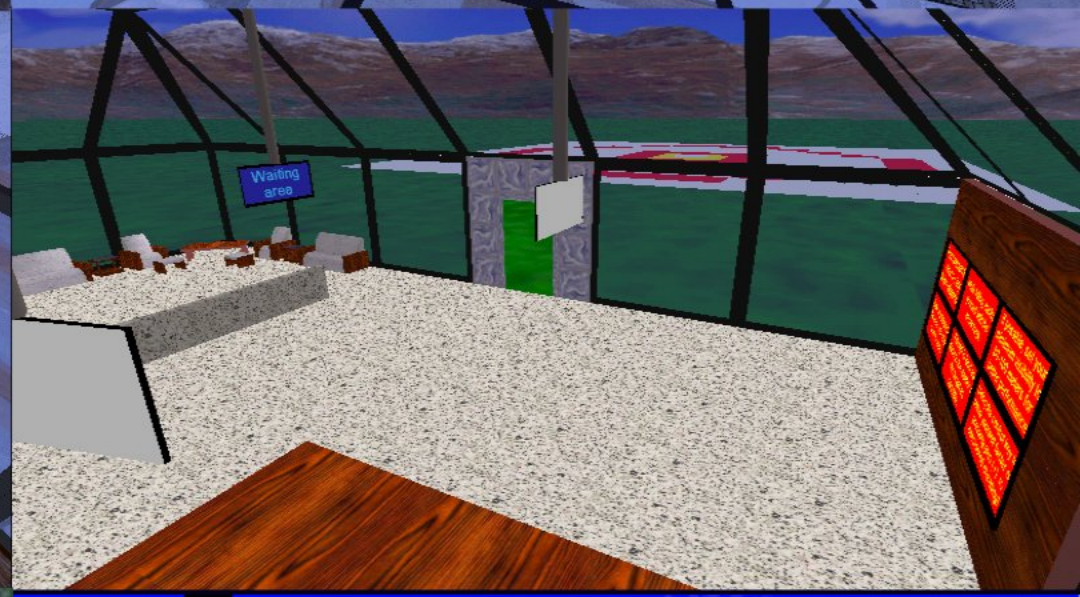




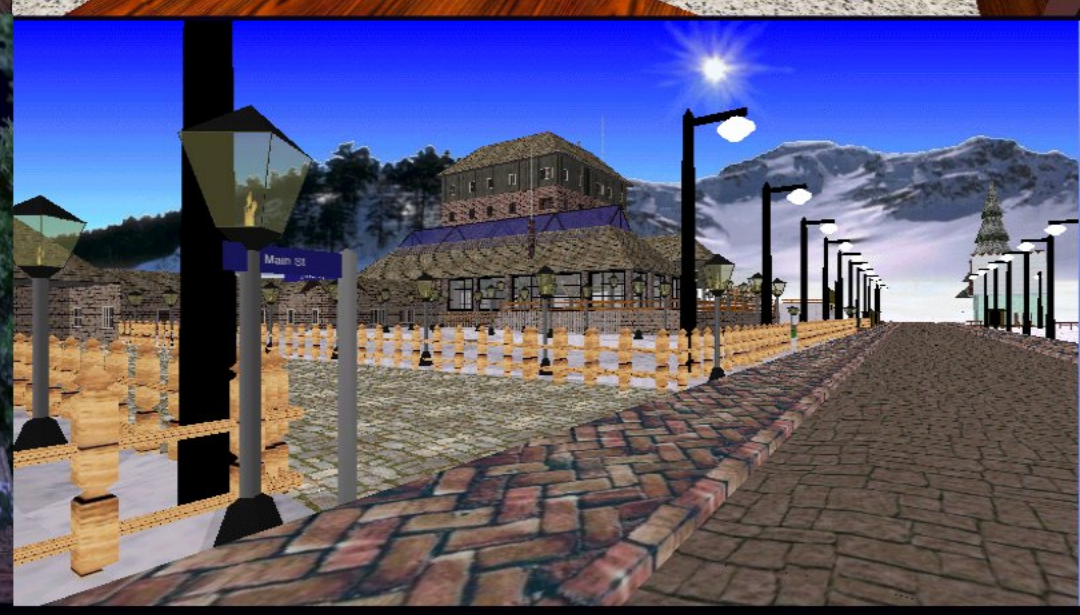












stephen sokolowski


Dawn of Darkness
Stephen Sokolowski

Overview

- Created a 3D game that can be played by Internet users around the world
- Programmed computer-controlled characters and guides (robots)
- Directed and developed the visual design
- Coordinated group efforts of helpers

In This Presentation…

- How the technology works
- Steps involved
- The product
- Difficulties encountered
- The Learning Stretch

Origins of the Project

- Discovered Active Worlds in 1998
- Previous attempt at creating a game in 1996 failed due to lack of knowledge and state of technology
- In summer 2000, started to write the game, eventually stopped because of a lack of time
- Started over from scratch

Terminology

- Server: a computer which runs an application to interact with other computers
- World: an area of virtual land, operated by a server run by a common person

Terminology (Part II)

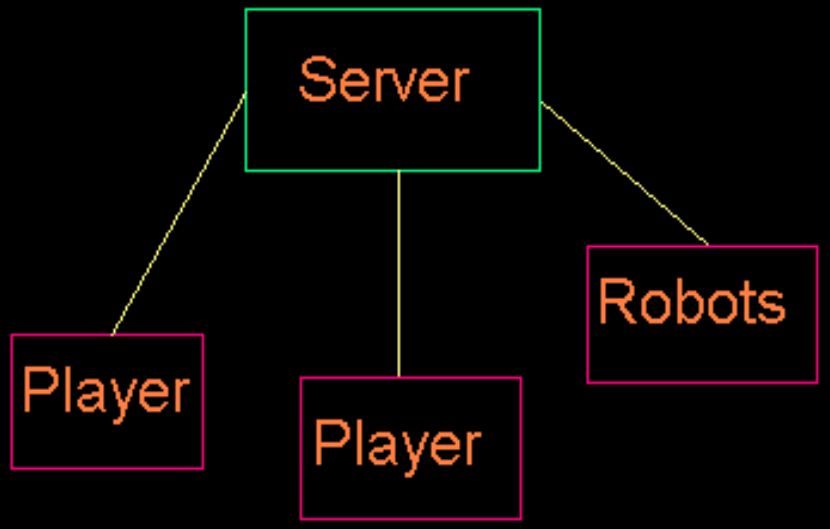
- Class: a method of programming where code is written for use in multiple applications; reduces common programming errors, usually written first
- C++: the programming language in which most commercial software packages are written
- Visual Basic: another programming language

Terminology (Part III)

- Object Path: a website where objects (trees, rocks, walls) are downloaded from and displayed on the screen
- Asynchronous: a method of computer communications where control is returned to the computer between the sending of data and a response so that other tasks can be accomplished

How the Technology Works

- Server is the program run at home to coordinate game
- Players connect to server along with robots
- Players' computers interpret commands and interact with robots




March 8, 2001
Began development of graphics

March 9, 2001



Construction of main town begins

March 12, 2001

Microsoft Visual C++ - [Gulakon.cpp]

File Edit View Insert Project Build Tools Window Help

```
#include "gulakon.h"
#include "aw.h"

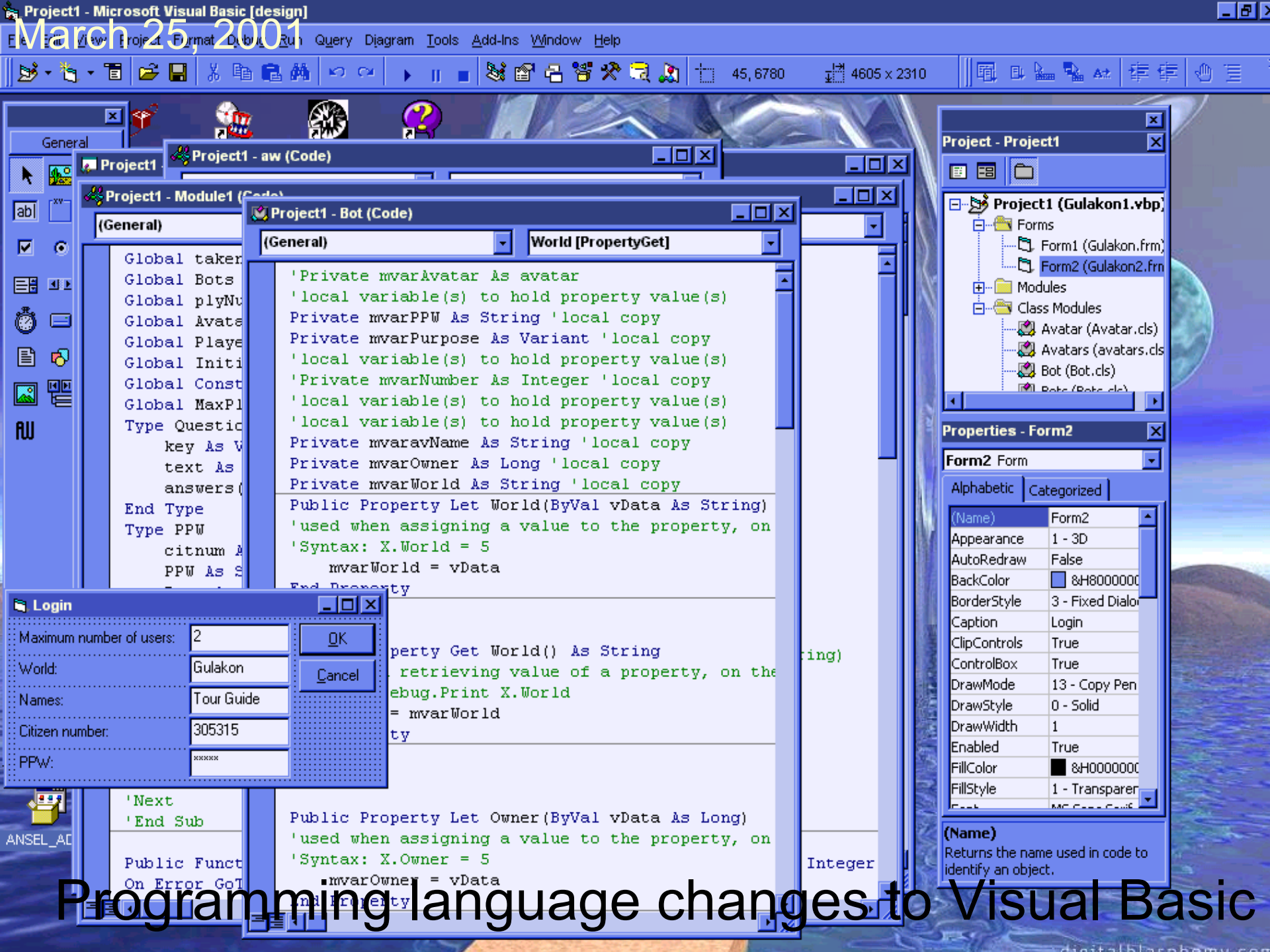
char* location::getLocation( void ) const //gets the location of an avatar in string format
{
   int x, z;
   char *c1 = "N";
   if ( x < 0 ) c1 = "S";
   char *c2 = "W";
   if ( z < 0 ) c2 = "E";
   return "Not finished";  //this function isn't finished yet, nor is it needed right now
}

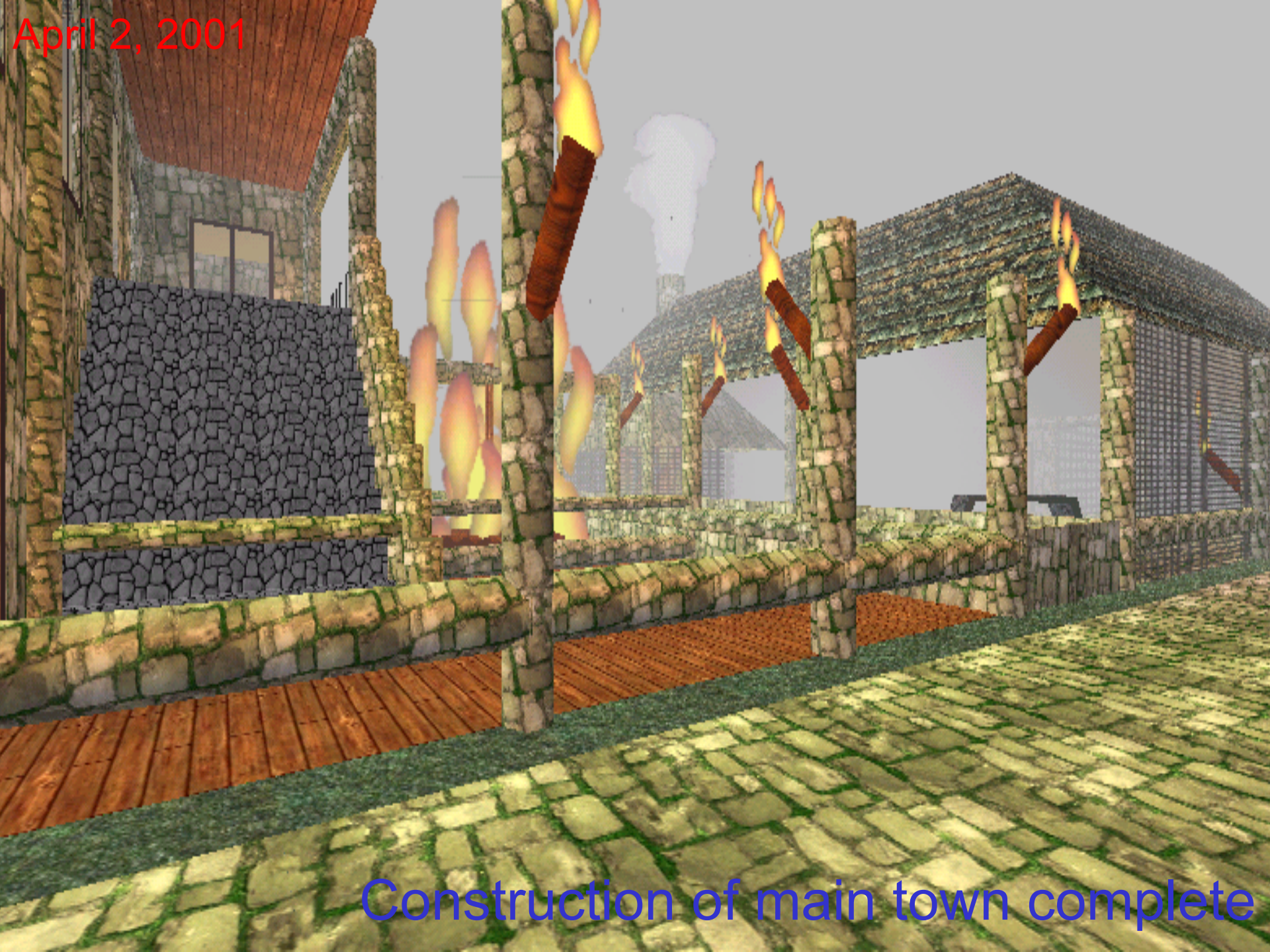
void avatar::getData ()
{
   citnum = aw_int( AW_AVATAR_CITIZEN );
   privs = aw_int ( AW_AVATAR_PRIVILEGE );
   version = aw_int ( AW_AVATAR_VERSION );
   loc.x = aw_int ( AW_AVATAR_X );
   name = aw_string (AW_AVATAR_NAME);
   loc.y = aw_int ( AW_AVATAR_Y );
   loc.z = aw_int ( AW_AVATAR_Z );
   loc.yaw = aw_int ( AW_AVATAR_YAW );
   gesture = aw_int ( AW_AVATAR_GESTURE );
   type = aw_int ( AW_AVATAR_TYPE );
   session = aw_int ( AW_AVATAR_SESSION );
   instance = aw_instance();
}

avatar::avatar()
{
   type = 0;
   name = "";
   session = 0;
   gesture = 0;
   citnum = 0;
   privs = 0;
   version = 0;
}
```

Build | Debug | Find in Files 1 | Find in Files 2 | Results

Began programming in C++


March 25, 2001
Project1 - Microsoft Visual Basic [design]
Project1 - aw (Code)
Project1 - Module1 (Code)
Project1 - Bot (Code)
(General)
World [PropertyGet]
'Private mvarAvatar As avatar
'local variable(s) to hold property value(s)
Private mvarPPW As String 'local copy
Private mvarPurpose As Variant 'local copy
'local variable(s) to hold property value(s)
'Private mvarNumber As Integer 'local copy
'local variable(s) to hold property value(s)
'local variable(s) to hold property value(s)
Private mvaravName As String 'local copy
Private mvarOwner As Long 'local copy
Private mvarWorld As String 'local copy
Public Property Let World(ByVal vData As String)
'used when assigning a value to the property, on
'Syntax: X.World = 5
mvarWorld = vData
Public Property Get World() As String
'Next
'End Sub
Public Property Let Owner(ByVal vData As Long)
'used when assigning a value to the property, on
'Syntax: X.Owner = 5
Login
Maximum number of users:
2
World:
Gulakon
Names:
Tour Guide
Citizen number:
305315
PPW:
OK
Cancel
Project - Project1
Project1 (Gulakon1.vbp)
Forms
Form1 (Gulakon.frm)
Form2 (Gulakon2.frm)
Modules
Class Modules
Avatar (Avatar.cls)
Avatars (avatars.cls)
Bot (Bot.cls)
Properties - Form2
Form2 Form
Alphabetic
Categorized
(Name) Form2
Appearance 1 - 3D
AutoRedraw False
BackColor &H8000000
BorderStyle 3 - Fixed Dialo
Caption Login
ClipControls True
ControlBox True
DrawMode 13 - Copy Pen
DrawStyle 0 - Solid
DrawWidth 1
Enabled True
FillColor &H0000000
FillStyle 1 - Transparer
(Name)
Returns the name used in code to identify an object.
Programming language changes to Visual Basic


April 2, 2001
Construction of main town complete


April 3, 2001
Interiors of buildings begun


April 13, 2001
Construction of castle finished


April 25, 2001
Sky city begun

Steps Involved

- Started out with ground and sky, and a blank C++ project
- Organized helpers to create visual environment
- Started programming of robots after sufficient graphics completed
- Solved problems as they were encountered

Unregistered HyperCam



[Tour Guide 1]: *Welcome to Dawn of Darkness, "Quintin"! I will be your guide today. Say 'begin' to create a new character and start on the quest!*

[Tour Guide 2]: Dawn of Darkness version 1.00

Whisper To

The Product

What I accomplished

- Logging in
- Tracking of players
- Storytelling/Chat
- Music
- Graphics
- Weather

What I plan in the future

- Battle system
- Expanded storyline
- Quests/missions
- Shopkeepers

Difficulties Encountered

- Deletion of object paths
- Problems with asynchronous communications
- Change of programming languages
- Pace of class development
- Viruses and lost time
- Other technical problems

The Learning Stretch

- Had never attempted to create a game of this magnitude
- Became proficient in programming in both Visual Basic and C++
- Interacting with people taught me the value of synergy
- Learned how to overcome difficulties

Thank you!

The recorded tour

The film used in slide 18 runs 3 minutes 5 seconds. These frames follow the tour from the town street to the castle and the guide's dialogue.



0:00 The guide welcomes the player.



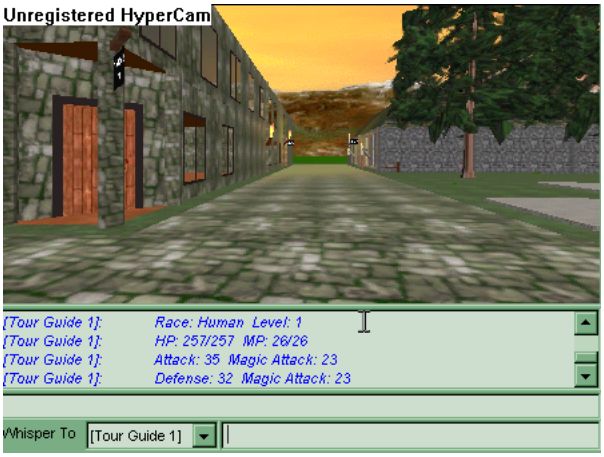
0:55 A patterned disc in the square.



1:14 The gate across the road.



1:38 The bridge to the castle.



2:16 The guide gives the player's status.



2:59 The guide continues the story.

Save clip0027.avi from the PDF's Attachments panel to play the original recording.

Watch on YouTube

http://www.eingang.org/Lecture/pascaline.html
-first mechanical adding machine 1642 – had wheels that, when turned in the proper order, would give the result of a calculation – technophobia emerges
-charles babbage realizes in 1812 that many mathematical operations are repeatedly performed, sets out to design a completely automated machine to solve complex problems by repeating simple steps – difference engine
-Herman Hollerman – 1890 – develops device that could automatically read census info from cards – founds corp called IBM, huge success – only could add
-1941- Konrad Zuse develops first machine that could perform complex calculations using binary system
-Howard Aiken develops Mark I computer at IBM in 30s – mechanical computer that can perform all four operations
-ENIAC in Nov 1945 - based on vacuum tubes and used too many tubes to have much storage space - used decimal numbers instead of binary
-software was nearly impossible to develop because nothing of the sort had been attempted before - a small bug could bring down an important program, so new languages were developed
-EDSAC is the next computer - has mercury delay lines that hold 1kb of data each - 1/6 the size of eniac - may 1949
-in 1951, computer begins to shift toward a business machine instead of a mathematical tool - Ferranti Mark I - first computer on market in Feb 1951 by british company
-in late 40s, IBM rejects the computer as a machine that will only be used as a mathematical instrument
-Card Programmed Calculators (models 603 and 604 developed by IBM in the 50s) appear, sell up to 5600 units
-stored program computers become a large priority in the late 40s, don't want to have to put in punch cards all the time
-ibm concentrates on defense contracts and falls behind in the business market
-1960 to 1962 ibm goes from punchcard supplier to computer supplier
-UNIVAC used to predict election in 1952
-IBM 1401 was the first transistorized computer in 1960
-Report program generator allowed programmers who were familiar with wiring electronic boards to make programs
-Ibm and other companies plagued with incompatible products, and develop the New Product Line (NPL) of compatible computers
-Altair 8800 first PC in 1975 - called a mini computer because PC had not been coined

-software
-subroutine libraries were developed in the middle part of the century because programming was becoming nearly impossible
-Grace Hopper puts together the code for an automated compiler to make programming easier
-FORTRAN is the first language that came into being - allowed words to simulate many lines of machine code
-COBOL was developed as a standard by the US government around 1960, to make all future computer systems compatible (they hoped) - used english words
-software costs were the single most important cost involved with the creation of a computer system in the 50s, and software contractors and public libraries came into existence
-software crisis of the 1960s erupts when computers developed much more fastly than software could take advantage of
-operating systems come into existence in the late 1950s
-software was frequently buggy and million-line programs were impossible to create, such as an operating system
-OS 360 - biggest software debauchle ever - half a billion dollars and 5000 man-years of labor resulted in buggy software that couldn't be repaired
-turns out that small problems contributed to the trouble, too many programmers
-Mariner I spacecraft on July 22, 1962 had to be destroyed because of a single incorrect FORTRAN character
-structured programming design allows programmers to concentrate on a small part of a program without worrying about the larger program

-December 1968 - IBM decides to publish its software and hardware separately, leads to "packaged" programs as opposed to contracting

BASIC develops when FORTRAN is too complicated for students to use - spreads first to other universities around 1970, opens up a new field of users
-time-sharing eliminates the problems caused by wasted processing time, computer utility (many users interacting) leads to the development of the Internet, but collapses first and is then rebuilt
-Unix developed in 1974 - simple approach allowed it to succeed over systems such as OS/360
-C is a language developed to make Unix portable to be used on any computer system Ken Thompson and Dennis M. Ritchie
-minicomputer is a term that came into existence to describe lower-end systems
-digital watches and calculators were coming into their own around 1975
-Atari develops a home version of Pong in 1974 - several years later video games had become a new industry
-microprocessor was independently invented in two places, but Intel was the primary focus - Robert Noyce and Gordon Moore
-Altair 8080 was the first PC - true only in the fact that its price was low enough to buy by a normal person
-256 bytes of memory, programs entered only by binary code on front
-Gates is one of the people who discovers personal computers and writes his programs in the 1970s - he developed BASIC for the Altair computer system
-Apple II and Commodore PET both released simultaneously in 1977 - beginning of the PC revolution
-games were the impetus in the PC software revolution
-in 1980, IBM contracts Intel to develop chips
-Microsoft makes a contract with IBM to develop software for its computers - so in essence IBM started the revolution
-runaway success, beats out apple and Tandy computers - by 1983 most software packages converted
-January 1983 Man of the Year is a PC
-first too little software, now too much - 35,000 competing for 200 spots in a computer store
-GUI developed to eliminate the problems of the 20-hour estimated time to learn a software package
-in the 1980s, market became flooded with PC's and the average user couldn't understand them, so didn't grow as quickly.
-in early 1990s, three companies – Microsoft, Apple, and IBM compete for software dominance – Microsoft wins on advertising

development of the internet

-Denis Diderot – French philosopher, creates encyclopedia in the 18th century, wanted to make the world's store of knowledge accessible to ordinary citizens
-before the early 1800s, someone could learn everything there was to know easily – after, people couldn't learn both the arts and sciences because knowledge increases
-so much knowledge by 1900 makes specialization a trend, but some people want a systematic store of information so that this trend could be reversed
-HG Wells in 1937 proposes a World Encyclopedia – realizes that it might not be necessary for all the information to be concentrated in one place
-Vannever Bush, head of the Office of Scientific Research and Development during WWII, revives the project after Wells' death – making use of the information was the problem now, not containing it – microfilm was possible – a memex that was a desk where someone sat and went through information
-in mid 1960s Ted Nelson coins the term Hypertext
-JCR Licklider makes a proposal to develop ARPANET, which would link computers across the country to make use of idle processing time to save money
-Larry Roberts takes over the project, develops "packet switching" – telegraph example
-"store and forward principle"
-nodes would break up long messages and transmit them over the connections
-software problem was solved by putting a single operating system at nodes (the information message processor), which sends the data to computers

-graduate students worked on the first networks, which is why colleges are the mainstead of the internet today
-International Conference on Computer Communications in 1972 – showcases the network of computers
-Aarpanet introduces an E-Mail system in 1971, but nobody thought it would become as popular as quickly as it did
-email was the driving force behind just about all network innovations
-a “netiquette” developed because it was impossible to determine what was meant emotionally in email, and newbies quickly became confused
-there were separate networks that required proprietary software to run, so a great benefit was gained by Inter-networking, where networks were connected
-grew slowly without the help of personal computer users
-people put up entire documents, not just email on the internet, but cataloguing and searching through this imformation was impossible, so “gopher” was developed at the University of Minnesota – first search engine
-National Information Infrastructure proposed by Al Gore in 1991 – press brings people to connect to the Internet in record numbers

-in 1700s and 1800s, logarithm tables revolutionized the way mathematicians did their work - they saved lots of time
-nautical almanac – 1766, yearly issues of tables for nautical navigation

Where the pages came from

Paper documents

The cover and spine use the Word files prepared for printing. The panel letter carries the date saved in its original Word fields, March 27, 2001. The later journal ends on May 9 with a total of 95.8 hours.

Program listings

The Visual Basic screenshot names Avatar.cls, avatars.cls, and Bot.cls. Several declarations and the World property match the recovered Bot.cls line for line. Two companion files describe the game player and the collection of players. All five are identical in two archived ZIPs. Their dates are April 12 and May 8, 2001. The exact May 7 printout is still unverified.

Pictures and poster

The first picture sheet comes from the surviving Word template. The following 22 screenshots use that sheet’s landscape format, ordered by the dates in their filenames. Their captions were supplied for this reconstruction. The sign uses the full-size image saved in the layered Paint Shop Pro file.

Presentation film

Slide 18 calls for clip0027.avi. An AVI with that name survives in the AWTeen collection of bots and source code. It opens with a guide welcoming the player to Dawn of Darkness, version 1.00.

Digital originals

The PDF’s Attachments panel contains clip0027.avi and a ZIP of the source materials. The ZIP includes the original presentation, word-processing files, layered poster, bitmaps and five Visual Basic files.

From the presenter’s notes

Slide 2 *Explain what a role playing game is . Give example N 64 Zelda , or Paint ball*

Slide 20 *-These difficulties are why I never had said that I would actually be able to complete the project – too much time*

Slide 21 *Learned more with the difficulties I encountered than if I had accomplished more on the project*