FALLEN ANGEL SAGA

Demon Winter

Matthew Jones
Chris Sauerwald
Shaun West

Table of Contents

Overview

      "Fallen Angel Saga: Demon Winter" is a 2D side scrolling adventure game. The plot is a tale of a love gone wrong, and it follows the adventures of our hero as he struggles to save the world. Along the way, he will meet up with a number of different characters, obtain and use magical abilities, and acquire more powerful weapons and items. A large variety of evildoers will be out in full force to make sure that the hero does not succeed in his quest, and they will all need to be fought off to continue the game.

System Requirements

      This game will require a powerful computer and a fast video card. It worked excellent on our tests on and Athlon XP1800 with Radeon 8500, it work acceptably on a Pentium 3 with a Geforce 2 GTS. It performed poorly on Radeon VE video cards. It is very video card dependant. Otherwise it should operate fine on any version of windows and easily portable to linux in the near future.

Game Mechanics

OverView:
        Demon Winter (DW) is a side scrolling 2D adventure game. The interface and camera are fixed on the main character at all times at he travels through the world. It follows many of the pre-established mechanics for this genre.

- Camera Position:

        The camera will track so that the main character is always in view. It will not move until he starts to get near the edges of the screen, but will move along with him as he walks. In the images below, the character is running to the right as the camera follows him, which is reversed when he runs to the left.

- User Interface:

The display for the game shows the current gold and level of the character on the right of the screen, and the health as picture bars on the left. This is all of the information that the player needs when navigating through the world.

- Replaying and Saving

After the character dies, the game is over. You can replay from the beginning, but since the game is short, there are no saves. In later versions of the game, save points would be introduced, but a player should be able to complete the game in less than 10 minutes. Health does recharge as the player plays, so he should always be able to finish the adventure.

- Control Summary

The player is controlled by of 4 directional controls (Up/Down/Left and Right) and 2 attack buttons. The left and right buttons will move the character left and right across the screen. The up button will make the character jump and the down button will make him duck. These are controlled with the arrow keys on a keyboard.

One of the action buttons(Z) will swing the main weapon that is a sword, and the secondary buttons (A) will shoot off a magic spell if the character has enough magic points. The action button can also be used to talk to NPC's in the world.

As a secondary usage, a game pad would provide even greater control and better immersion and the game fully supports a two button game pad.
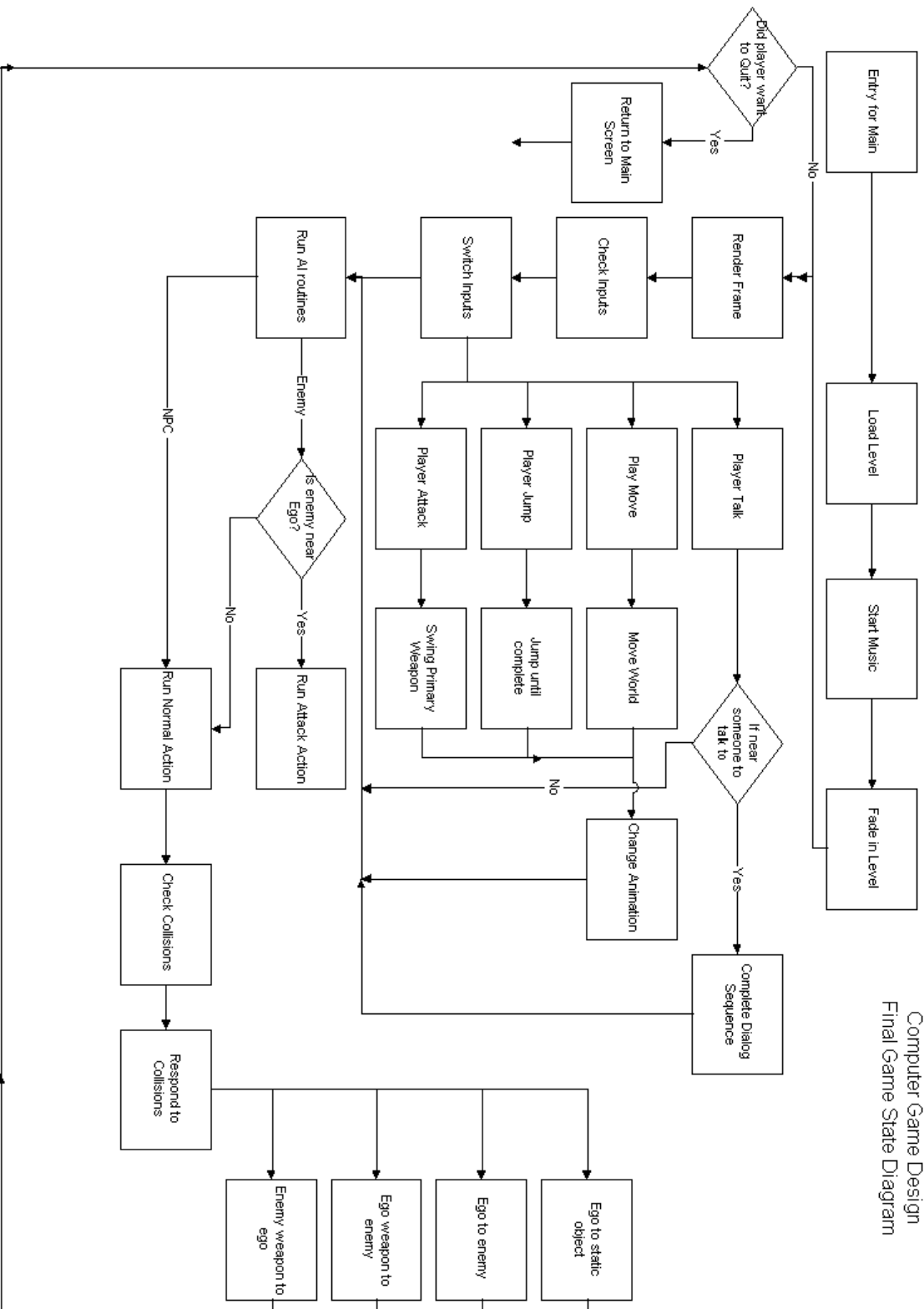
- Game Play Details

The game play will be 2D side scrolling console style game play. The primary goal will be for the character to advance the story by:

- Communicating with Non-Player Characters
- Defeating minor enemies
- Defeating bosses
- Solving puzzles (Primarily involving bosses)

The player can converse will all of the NPCs in the game world who all have something different to say.

CIS 587
Computer Game Design
Final Game State Diagram

Entry for Main

Did player want to Quit?
- Yes → Return to Main Screen
- No

Load Level → Start Music → Fade in Level

Render Frame → Check Inputs → Switch Inputs → Run AI routines

Switch Inputs:
- Player Talk
- Play Move
- Player Jump
- Player Attack

Player Talk → If near someone to talk to
- No
- Yes → Complete Dialog Sequence

Play Move → Move World → Change Animation

Player Jump → Jump until complete

Player Attack → Swing Primary Weapon

Run AI routines:
- Enemy → Is enemy near Ego?
  - No → Run Normal Action
  - Yes → Run Attack Action
- NPC → Run Normal Action

Run Normal Action → Check Collisions → Respond to Collisions

Respond to Collisions:
- Ego to static object
- Ego to enemy
- Ego weapon to enemy
- Enemy weapon to ego

- Cut-Scenes

Cut scenes are handled with pop-up dialog boxes at key points in the game, such as on a chat with the demon Gurog. After the conversation is done, the game will continue. There are multiple instances of dialogs being used by the game.

Level Information:

Level Loader:
We are using a plain text file with tab-delimited fields for a level description file. The file format supports //line comments that allow us to put in human readable information in the level files. This file holds most information needed to describe a level. Including
- Scenery data
  - Model filenames/skin filenames
  - Scaling factor
  - Position(x,y,layer)
- Enemy data
  - Model filenames/skin filenames
  - Scaling factor
  - Position(x,y,layer)
  - AI tag
  - Gold value
  - Exp value
  - Damage they do
  - Hit points
  - Armor
  - Velocity(x and y)
- NPC data
  - Model filenames/skin filenames
  - Scaling factor
  - Position(x,y,layer)
  - AI tag
  - Velocity(x and y)

Level file sample:
```
//<model file>   <skin>  <x>      <y>      <z-layer>        <scale>
//background large houses
data/l-house.md2  data/l-house.pcx  0        -1.5     SClayer3 1
```

Layers:
The following are tentative z coordinate positions in the world for different 'layers' mentioned in this section of the document.

Background: (Scenery area)
- Scenery layer 3(Sclayer3): z = -15.0
- Scenery layer 2(Sclayer2): z = -7.2
- Scenery layer 1(Sclayer1): z = -6.6

Foreground: (EGO area)
- EGO layer 3(Eglayer3): z = -6.5
- EGO layer 2(Eglayer2): z = -5.0
- EGO layer 1(Eglayer1): z = -3.5

Level Descriptions
The following section describes the general idea behind each level and the model files used in each level. The naming schemes for the models mentioned in this area correspond to those in the character bible.

First level
Opening:
Our main character is standing on a road near the town. . Trees line the streets and houses can be seen in the background. As the character approaches the town enemies confront him.

Background:
- Models
  - Sclayer3
    - L-house
    - Mansion1
    - Mansion2
  - Sclayer2
    - Tree1
    - Tree2
    - Tree3
    - Tree4
  - Sclayer3 -- empty

Foreground
- Models
  - Eglayer3 -- empty
  - Eglayer2
    - Player
    - Enemies
      - Lesser demon
      - Hell spider
  - Eglayer1
    - Tree1
    - Tree2
    - Tree3
    - Tree4

Street area:
As we enter town we see NPCs running about. We can interact with some of them; in fact some might run up to ask us for help.

Background
- Models:
  - Sclayer3
    - L-house
    - Mansion1
    - Mansion2
  - Sclayer2
    - Store-apartment
    - Downtown1
    - Apartment-building
  - Sclayer1
    - Occasional trashcan

Foreground:
- Models:
  - Eglayer3
    - Interactive NPC's
    We will be able to talk to them but they will not block our path.
  - Eglayer2
    - Player
  - Eglayer1 -- empty

Damaged area of town
NPC's are gone now and the town is damaged, burnt crumpled buildings in the downtown area. Enemies start to appear once again, in larger numbers and more powerful.

Background:
- Models:
  - Sclayer3
    - Downtown-damage
    - Store-apartment-damage
    - Apartment-building-damage
  - Sclayer2
    - Downtown-damage
    - Store-apartment-damage
    - Apartment-building-damage
  - Sclayer1
    - Trashcans

Foreground:
- Models:
  - Eglayer3 -- empty
  - Eglayer2
    - Player
    - Enemies

- Hell spiders
- Skeletons
- Lesser demon
- Winged demon
  - o Eglayer1 -- empty

Boss fight

The character finds themselves in an alley type area. Large demon 'Gurog' is on the screen when they enter and they begin fighting.

Background:
- Models:
  - o Sclayer3
    - ▪ L-house
    - ▪ Mansion1
    - ▪ Mansion2
  - o Sclayer2
    - ▪ Apartment-building-damage
    - ▪ Store-apartment-damage
    - ▪ Downtown-damage
  - o Sclayer1 -- empty

Foreground
- Models
  - o Eglayer3 -- empty
  - o Eglayer2
    - ▪ Player
    - ▪ Boss – Gurog
  - o Eglayer1 -- empty

Return to town

After successfully defeating the boss we return to town to be congratulated, thus ending the game. The player cannot leave this level.
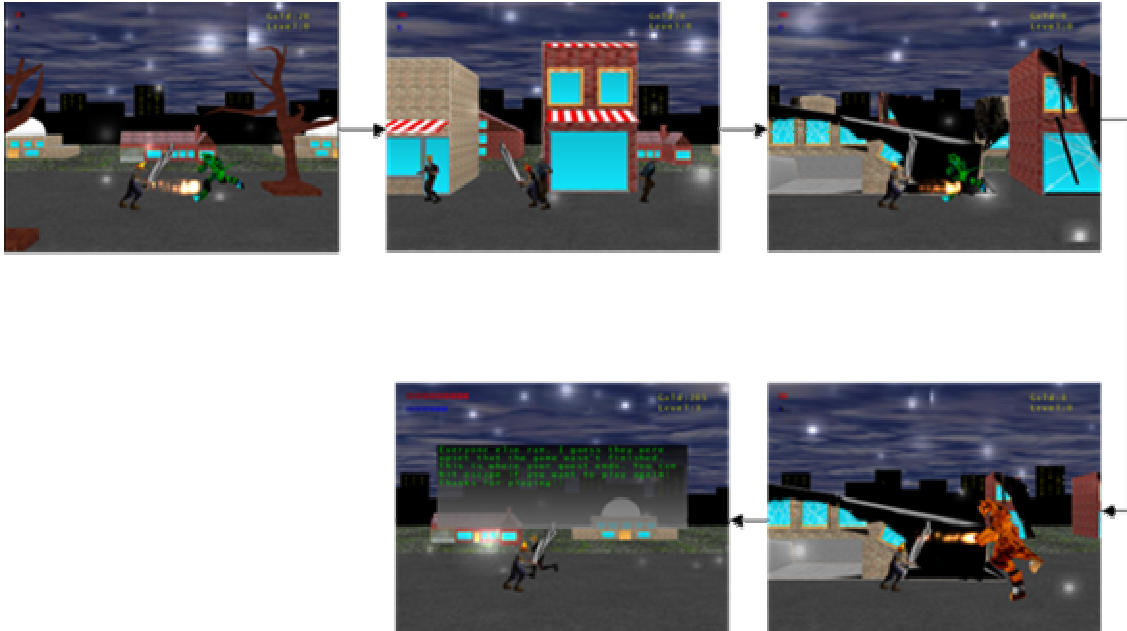
Background
- Models
  - o Sclayer3
    - ▪ L-house
    - ▪ Mansion1
    - ▪ Mansion2
  - o Sclayer2
    - ▪ Apartment-building
    - ▪ Store-apartment
    - ▪ Downtown1
  - o Sclayer1
    - ▪ Trashcans

Foreground
- Models
  - Eglayer3
    - NPC's
  - Eglayer2
    - Player
  - Eglayer1 – empty

Level Progression

AI

The game's AI system consists of a single class, which runs the appropriate algorithms on NPCs and enemies. An NPC or enemy on which the AI system is running is referred to as the *subject* and the player is referred to as *ego*.

Player Detection

The AI determines if the player is near by calling the *near_ego* function. This function compares the location of the subject to the location of ego and determines if they are close enough to run AI routines. Ego and the subject are considered to be near each other if they fall within a set x and y range. If they are not within range of each other, no AI operations are executed.

Motion

The subject's movement is dependent on location and AI algorithm. The *near_gamewindow* function determines whether the subject is near or within the viewing area. If near the viewing area, AI algorithms execute on the subject. Depending on the algorithm, the subject will walk randomly, approach ego, or fly about in a pattern.

Special Actions

Some special actions controlled by AI are: attacking, ranged attacking, swooping, and stomping. When these functions execute depends on AI algorithm and location. Attacking normally does not commence until the *near_ego* check succeeds. Ranged attacking, swooping, and stomping execute at random times.

Combat

There are various forms of combat. Only enemies execute combat routines. The AI will run the normal melee attack when the *near_ego* check succeeds, if the routine supports melee attacks. Flying enemies can use a swoop attack, which runs at random times when the *near_gamewindow* check succeeds. Enemies that make use of ranged attacks and stomp attacks will also only execute their attacks when *near_gamewindow* and a random check both succeed.

AI State Diagrams:
Standard Walking/Ranged Attack

Standard Walking

```
                    ┌──────────────────┐
                    │                  │
                    └─> Start <─────────────────────────┐
                         │                               │
                         ▼                               │
                      ╱Subject╲                          │
                     ╱─────────╲                         │
          ┌──────────┼────────────────┬──────────┐       │
       is │       is │          is    │     else │       │
       Dead│      Damaged│     near window│       │       │
          ▼          ▼                ▼          ▼       │
    ┌─────────┐ ┌──────────┐    ┌─────────┐ ┌──────────┐ │
    │DeathAction│ │RecoilAction│  │ Face Ego │ │Stop Action├─┤
    └─────────┘ └──────────┘    └─────────┘ └──────────┘ │
                                     │                    │
                                     ▼                    │
                                  ╱  is  ╲    NO          │
                                 ╱ near   ╲──────> ┌──────────┐
                                 ╲ ego    ╱        │  Move    ├──
                                  ╲──────╱         │ Action   │
                                     │             └──────────┘
                                     ▼ yes
                                ┌─────────┐
                                │ Attack  │
                                │ Action  │
                                └─────────┘
```
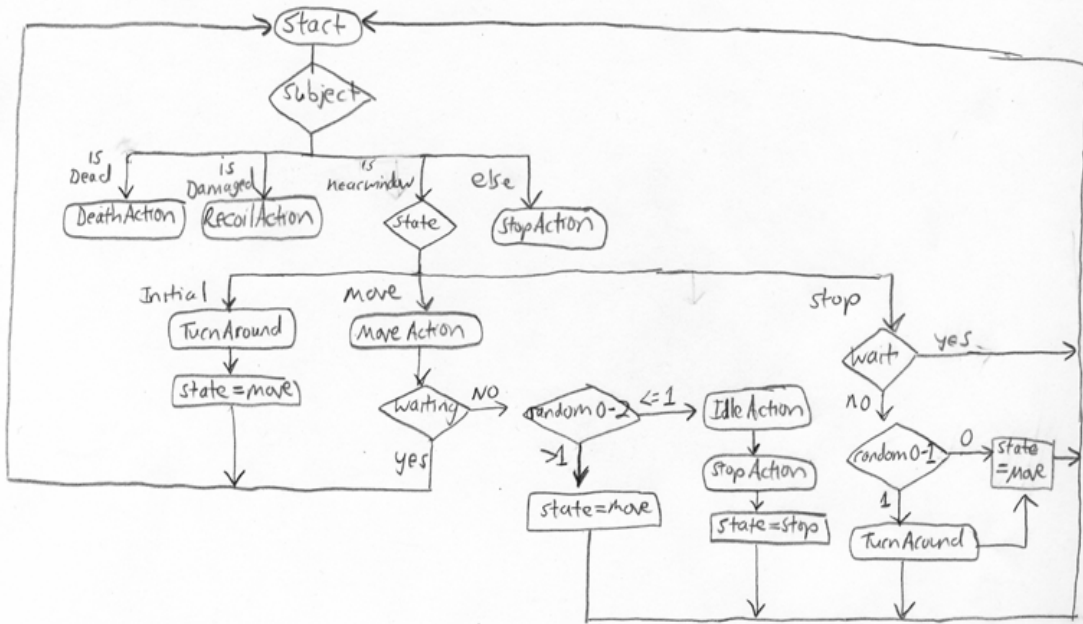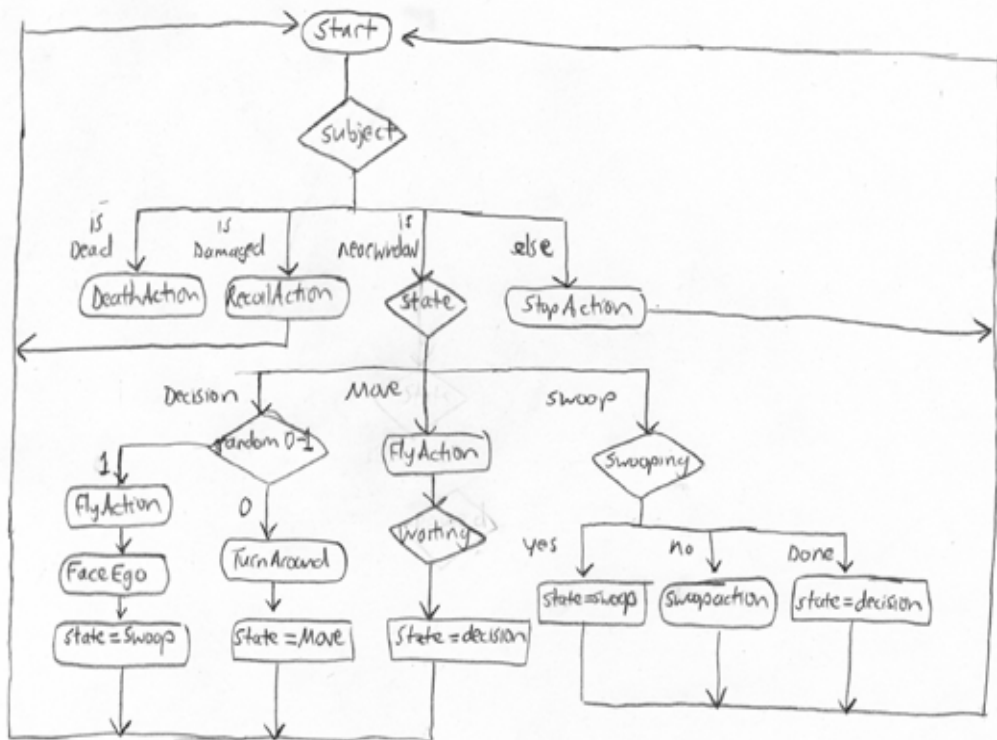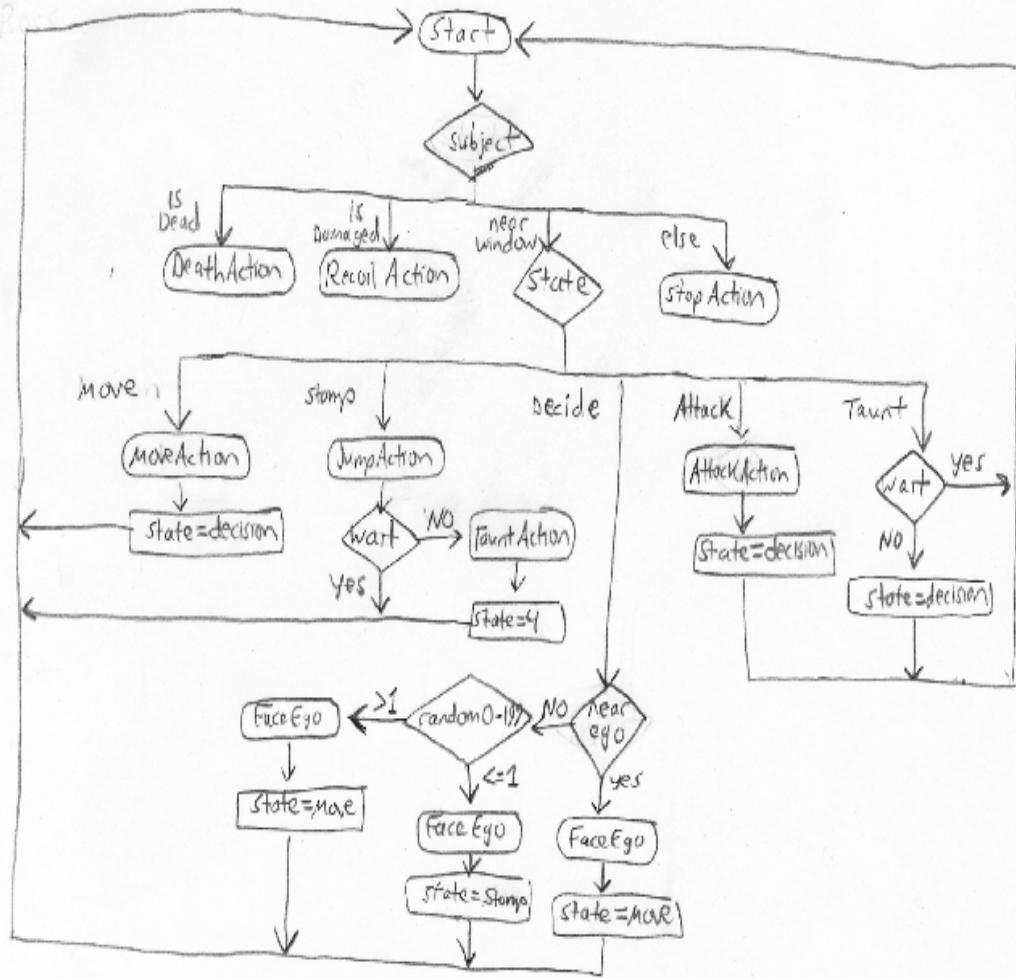
Ranged Attacking

```
                    ┌──────────────────┐
                    └─> Start <─────────────────────────────┐
                         │                                   │
                         ▼                                   │
                      ╱Subject╲                              │
          ┌───────────┼──────────┬──────────┐                │
       is │        is │      near window│  else│             │
       Dead│      Damaged│            │       │             │
          ▼           ▼          ▼          ▼             │
    ┌─────────┐ ┌──────────┐ ┌─────────┐ ┌──────────┐       │
    │DeathAction│ │RecoilAction│ │ Face Ego │ │Stop Action├──────┤
    └─────────┘ └──────────┘ └─────────┘ └──────────┘       │
         ▲            ▲           │                           │
         └────────────┴───────────┼──┐                        │
                                  ▼                           │
                               ╱Subject╲                      │
                    ┌──────────┼──────────┬──────────┐        │
               near │     Flag ON│    random│     else│        │
               ego  │           │         │         │        │
                ▼           ▼         ▼         ▼        │
         ┌──────────┐ ┌──────────┐ ┌──────────┐ ┌──────────┐ │
         │Attack Action│ │Increment flag│ │Increment flag│ │Move Action├─┤
         └──────────┘ │   count   │ │   count   │ └──────────┘ │
                       └──────────┘ └──────────┘               │
                              │    ┌─────────────┐   │         │
                              └──> │   Action    │ <─┘         │
                                   │Ranged Attack│             │
                                   └─────────────┘─────────────┘
```

Wander

Wander

Start

Subject

is Dead → DeathAction
is Damaged → RecoilAction
is nearwindow → state
else → StopAction

Initial → TurnAround → state=move

Move → Move Action → waiting
  - NO → random 0-2
    - >1 → state=move
    - <=1 → Idle Action → Stop Action → state=stop
  - yes

stop → wait
  - yes →
  - no → random 0-1
    - 0 → state=move
    - 1 → Turn Around

Fly

Flying

Start

Subject

is Dead → DeathAction
is Damaged → RecoilAction
is nearwindow → State
else → Stop Action

Decision → random 0-1
  - 1 → FlyAction → Face Ego → state=Swoop
  - 0 → TurnAround → state=Move

Move → FlyAction → waiting → State=decision

swoop → Swooping
  - yes → state=swoop
  - no → Swoopaction
  - Done → state=decision

Boss

Boss

Character Bibles

Cast

Name: Jay
Age: 23
Occupation: Student of the lost Martial-Arts
Description:
Jay is the hero of the game. He and his old friend Paul attended high school together, but while Paul and all of the other kids went off to find jobs, Jay sought something more. He sought the skills of the world's greatest warriors and the achievement of inner peace. To reach his goal Jay earned his way into the Temple of the Dragon to learn from the masters there. Jake cares nothing for power or wealth; he seeks only the betterment of himself.

Name: Paul
Age: 23
Occupation: Entrepreneur
Description:
Paul is a lover of money and women. In high school he was always the opposite of his good friend Jay. Which is part of the reason they drifted apart. After High School, Jay left and Paul went on to make a large sum of money in business. For a long time Paul was dating Jamie, until she went west in search of a career in acting.

Name: Jamie
Age: 22
Occupation: Waitress/Aspiring Actress
Description:
Jamie is a ditz. She's thin and blonde, every man's dream, but she hasn't a shred of brains in her body. She met Paul in High School and dated him for some time before the acting bug bit her. Suddenly she dropped everything, including Paul and went out west to seek employment.

Name: Lisa
Age: ??
Occupation: Fallen Angel
Description:
Lisa was an Angel who spent an eternity in the heavens doing good deeds and whatnot until she grew tired of that life. Looking down from the clouds, she spotted Paul and knew right then that she wanted the love of a mortal. So she fell from the heavens and in no time at all she and Paul were together.

NPCs

Assorted NPCs can be found scattered about the city. Most will talk to the player and some give helpful clues to help the player out.

Enemies

Enemies will attack the player in an attempt to stop him from reaching his goal. Each has various statistics corresponding to it including experience, gold, health, armor, and damage.

Experience (Exp.): The amount of experience points the player receives after defeating the enemy.

Gold: The amount of gold the player receives after defeating the enemy.

Health: Determines how many times the enemy must be hit before dying.

Armor: The enemy's resistance to attacks.

Damage: The amount of damage dealt to the player when hit by the enemy.

Name: Gurog
Exp: 200   **Gold:** 75   **Health:** 150
Armor: 5   **Damage:** 3
Description:
Gurog is a powerful demon with a mean hammer. Being so big, he's quite slow, and he's not too smart. But he is incredibly strong and excellent at breaking stuff.

Name: Lesser Demon
Exp: 10   **Gold:** 1   **Health:** 2   **Armor:** 0   **Damage:** 1
Description:
Though not as strong as other demons, when unleashed upon the earth, Lessers can cause a great deal of havoc and destruction. They cannot fly and attack with claws.

Name: Skeleton Minion
Exp: 20   **Gold:** 2   **Health:** 5   **Armor:** 1   **Damage:** 2
Description:
Like their cousins, these demons aren't incredibly strong but are good at causing trouble. These demons have the ability to blow fireballs from their mouths.

Name: Hell Spider
Exp: 10   **Gold:** 3   **Health:** 1
Armor: 1 **Damage:** 1
Description:
These spiders are fast and deadly. They attack by slashing and biting victims.

Name: Winged Demon
Exp: 25   **Gold:** 4   **Health:** 6
Armor: 2   **Damage:** 2
Description:
These flying demons swoop down to attack with sharp claws.

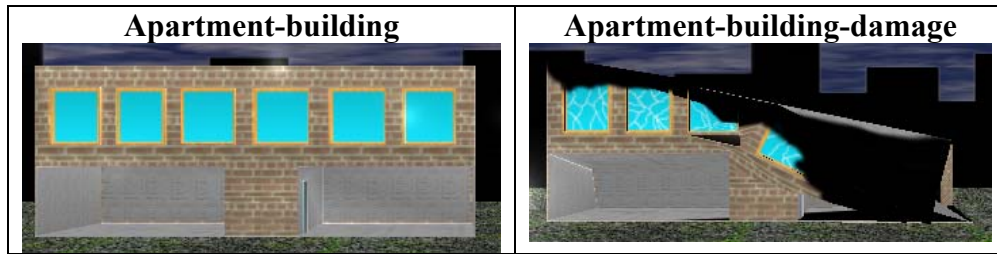## Scenery Model creation process:
Models were created initially in "Wings 3D", a simple 3D modeler. From there they were exported into "LithUnwrap". UV maps for texturing the models were made in "LithUnwrap". Using the BMP file representation of the UV map, GIMP was used to skin the models. After initial skinning was complete the models were exported into MilkShape3D. In "MilkShape 3D" they were rotated, scaled, and translated as necessary and then exported to Quake2 .md2 format.

Scenery Models:
This section shows what the scenery models mentioned in the next section look like in the game.

l-house	Mansion1

Mansion2	Tree1

Tree2	Tree3	Tree 4	Trashcan

Downtown1	Downtown-damage

Store-apartment	Store-apartment-damage

Apartment-building	Apartment-building-damage
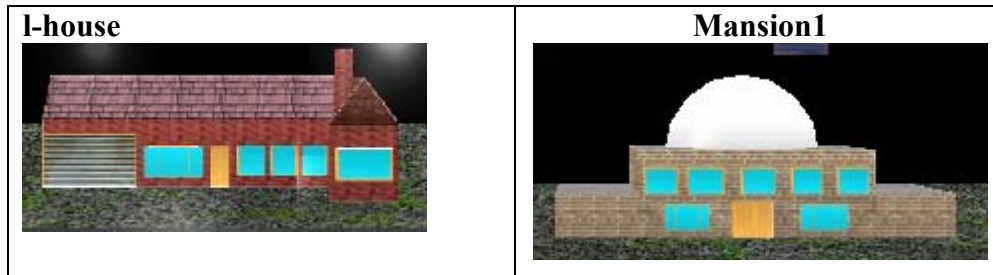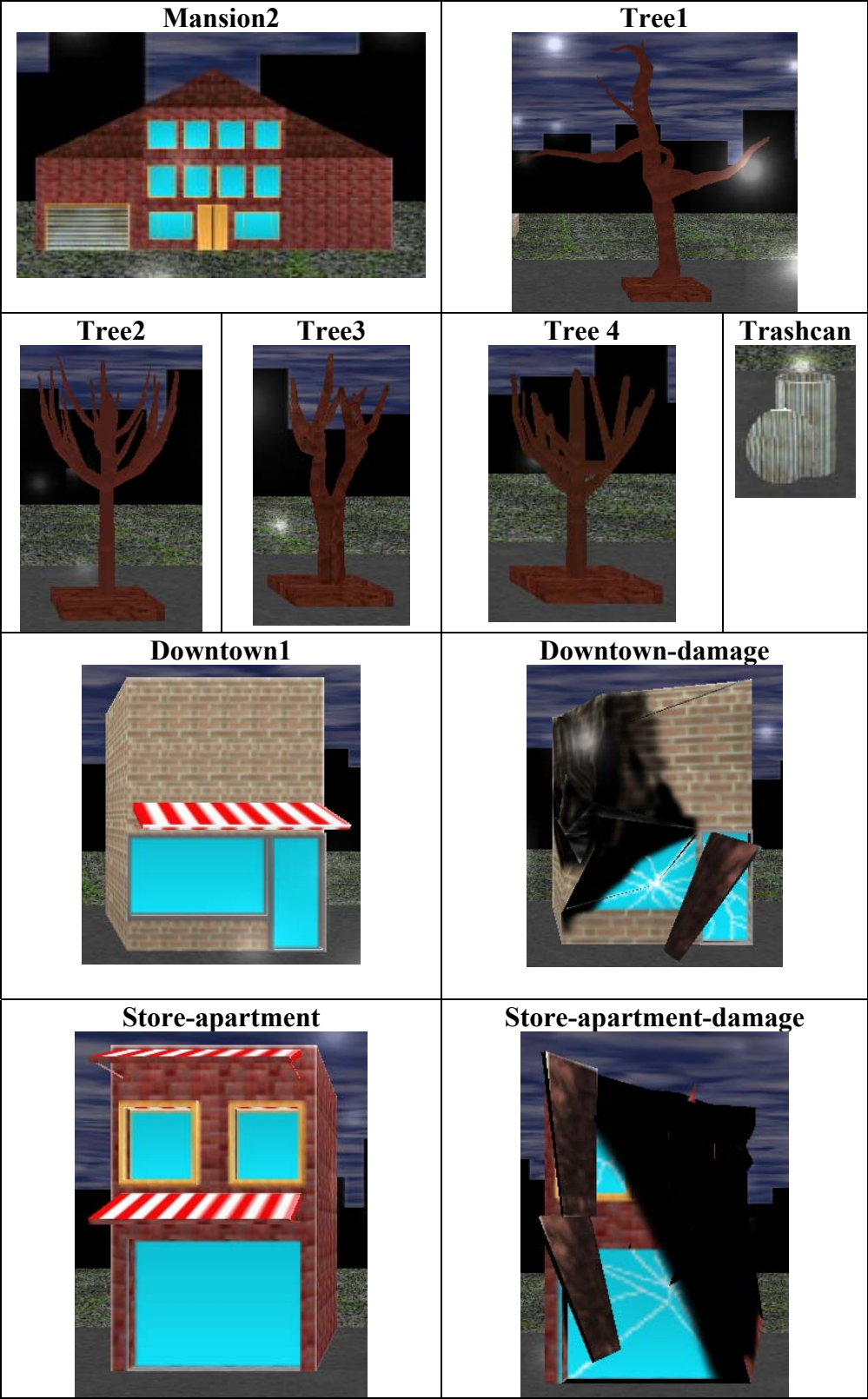
Story

The crisp autumn breeze was slowly beginning to give away to the harsh cold of winter when Paul walked the dark city street. Without warning, as if the gently falling snow had coalesced and took form, the most beautiful girl he had ever seen appeared before him. She said her name was Lisa and she sought only one thing: His love.

Paul could not resist the girl and the two quickly formed a relationship. Through the dark of winter and the light of spring, they were always together and never grew tired of one another. When summer came, they spent every moment together on the beach or at the park. By the arrival of fall, the two were thinking about marriage. Paul bought Lisa a ring and they made plans to marry on the anniversary of their first meeting.

Up until the wedding, everything was going well for the two. But only a few days before the anticipated event, a secret from Paul's past returned. Paul's old girlfriend, Jamie had come back to him and his eyes fell away from Lisa. Lisa begged him to stay with her, but Paul could not, for his heart still truly belonged to Jamie. Lisa, overcome with sorrow and seething with anger, departed just as she arrived. It seemed to Paul as if she faded into nothingness.

Little did Paul know, Lisa also had a secret. The winter before she appeared to him, she was watching from the clouds, high above the world. The moment she saw him, she knew that she had to be with him. So she took the fall. She gave up her Angelhood and left the heavens forever. Now she was alone, without purpose. The only thing left was vengeance. If she were to be without a home, then everyone would be without home! If she were to be without Paul, then no one would have him!

With all of the power within her, she summoned the dark Minions of the Underworld. They heeded her call and rose up from that place of fire and ash to the surface of the Earth. Lisa commanded them to wreak havoc upon the city where Paul lived. And they did as she said. Like a black cloud of evil they descended upon the city and infected everyone and everything with the disease of chaos. Fire ravaged shops and homes. The plague spread through the populace. Creatures of the Underworld stalked the streets and the people fled in terror.

Paul was shocked by the sudden turn of events but did not realize that Lisa was behind it until she once again appeared before him. She looked quite different now, no longer, glowing with beauty, she was dark and twisted. She revealed to him that she was a

Fallen Angel and yelled, "Look what you have caused! You were a fool to leave me and now you will pay!" And with that, she engulfed Jamie in a beam of light. "You'll be next, Paul!" she said and they both vanished.

Overwhelmed, Paul fell to the ground, helpless and distraught. He knew that this was partly his fault but there was nothing he could do. He was too rich and too handsome for fighting! How could he ever get Jamie back and save the city?

Far away from the city, deep in the wilderness at the Temple of the Dragon, Jay trained in the ways of the world's greatest warriors and most powerful wizards. He desired, through his training to achieve peace and enlightenment. Everything else meant nothing to him. Wealth, power, love. They were for the weak of mind. And he was well on his way to achieving his goal when he received the call for help from his old friend, Paul.

Jay knew he still had a long way to go in his training when he exploded with anger at the thought of having to leave the Temple. But his master convinced him that loyalty was a virtue he could not be without and that getting out a little wasn't a bad thing. So Jay grabbed his sword and his fighting uniform and left the peaceful countryside for the chaotic city.

Production Tools:

Visual C++ 6.0	(http://msdn.microsoft.com/vstudio/)
Gcc	(http://gcc.gnu.org/)
SDL	(www.libsdl.org)
(SDLMixer module)	
MilkShape 3d	(http://gcc.gnu.org/)
Quake 2 Modeler	
Wings 3D	(http://www.wings3d.com/)
Adobe Photoshop	(www.adobe.com)

OpenIL/Devil Image Library (http://openil.sourceforge.net)

ID Software, Various other quake 2 resources and tools

Production Team:

Matthew Jones: (jonespm@umich.edu)
Project lead, timeline organization
Main engine code and optimization
Special Effects, Physics
Music Mixing/Selection
Model Loader

Shaun West: (shaunwes@umich.edu)
Story and Plot
Cinematic Presentations and artwork
Artificial Intelligence
Sound Effects Programming/Timing

Chris Sauerwald: (csauerwa@umich.edu)
Background art/models/scenes
Level design, loading
Level layers and display
Foreground art/obstacles

Bibliography:
      The following OpenGL reference sites provided very valuable resources to the creation of this game:

- http://nehe.gamedev.net
- http://cone3d.gamedev.net
- http://msdn.microsoft.com/
- http://www.polycount.com/

Mason Woo, Jackie Neider et al. OpenGL(R) Programming Guide: The Official Guide to Learning OpenGL, Version 1.2. Addison-Wesley 1999.