

Aperfeiçoamento de reações comportamentais de *non-player character* (NPC) no jogo Doom

Acadêmico: Filipe Albano Mattedi

Orientador: Mauro Marcelo Mattos

Roteiro

- Introdução
- Fundamentação teórica
- Desenvolvimento
- Implementação
- Resultados
- Conclusão



Introdução

- Jogos eletrônicos
- Oponentes humanos x oponentes virtuais



Objetivos do trabalho

- Aperfeiçoar as respostas comportamentais dos NPCs do jogo Doom.
 - Engenharia reversa do código fonte
 - Manipulação do código fonte para afetar o comportamento dos NPCs
 - Comparar o jogo alterado com o original



Fundamentação teórica

- *Game AI*
 - Comportamento
 - Máquina de estados finita
- Engenharia reversa
- Doom
 - Doom Engine
 - WADs



Game Artificial Intelligence

- Aplicação prática, com menos teoria.
- Evolução rápida e constante.
- Controla o NPC, o ambiente, a lógica geral do jogo.
- Diversão proporcionada vem antes do realismo.



Game Artificial Intelligence

- Comportamento
- Máquina de estados finita
 - Exige pouco poder de processamento
 - Comportamentos repetitivos

Engenharia Reversa

- Descompilação de um sistema para entender seu funcionamento.
 - Copiar
 - Aperfeiçoar
 - “Crackear”

Doom

- *First Person Shooter*
- Publicado pela idSoftware em 1993
- Código fonte (Doom Engine) lançado ao público em 1997, exclusivamente para Linux
- Máquina de estados finita



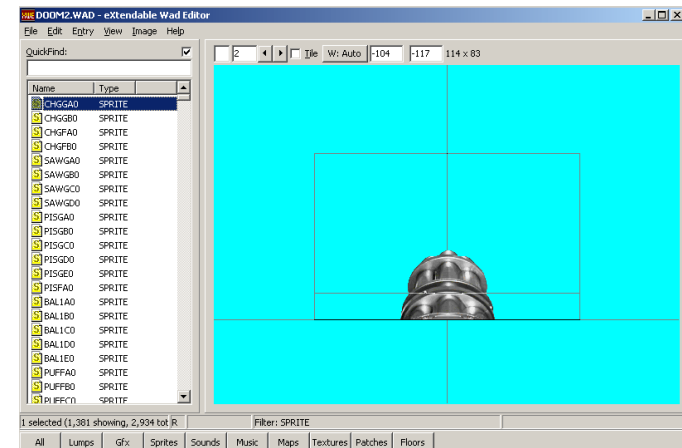
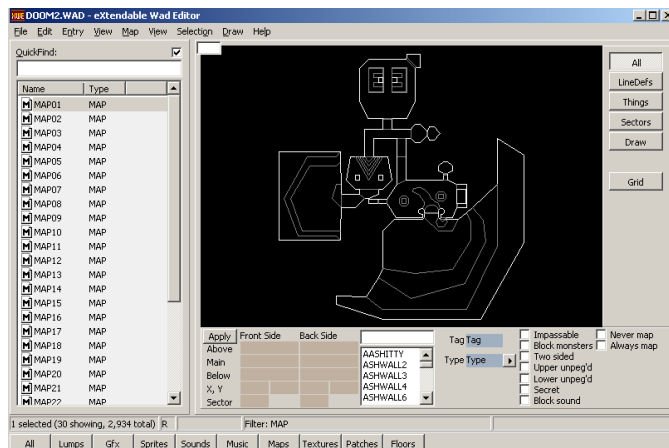
Doom Engine

- Engine de Doom, Doom 2, Hexen e Heretic.
- C++
- Feita para ser facilmente modificável.
 - Maleabilidade permitiu o surgimento de *source ports*, versões do código fonte modificadas para que funcionem em outros sistemas operacionais.
 - Desenvolvedores lançam versões com correções e adições ao código original.



WADs

- *Where's All the Data?*
- Armazenamento de informações
- Jogadores podem construir seus próprios *WADs* ou alterar os já existentes, com o uso de editores



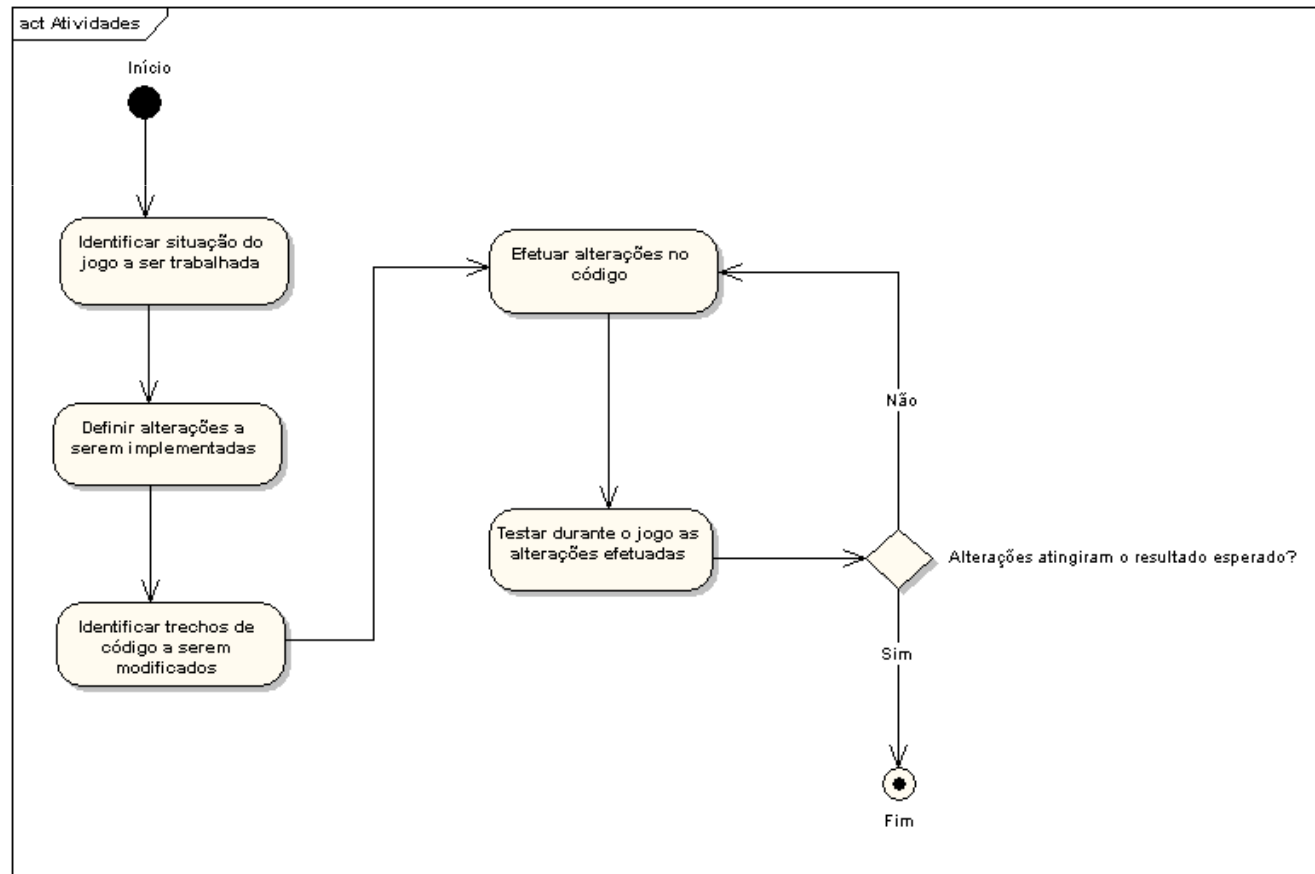
Desenvolvimento

- Ferramentas utilizadas
 - Gedit: manipulação do código fonte.
 - Enterprise Architect: especificação e engenharia reversa.
 - Microsoft Excel: confecção dos gráficos utilizados na comprovação de resultados.



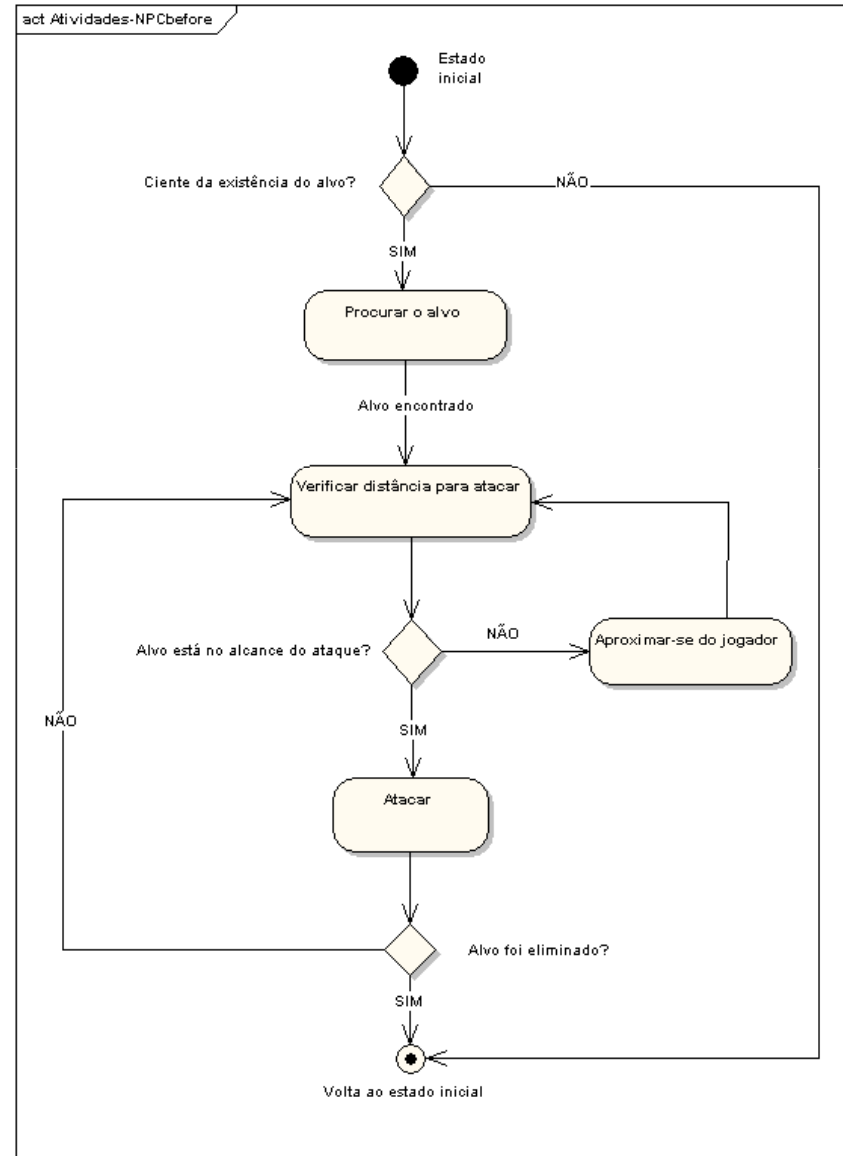
Desenvolvimento

- Diagrama de atividades que descreve as etapas do desenvolvimento:



Desenvolvimento

- Diagrama que demonstra as ações possíveis do NPC em relação ao jogador, no jogo original:



Desenvolvimento

- Situações selecionadas para se trabalhar:
 - O NPC não se afasta do jogador quando sob ataque; não preserva a própria vida.
 - O NPC não verifica se há um aliado em sua linha de fogo, ocasionando ataques indevidos a seus aliados.



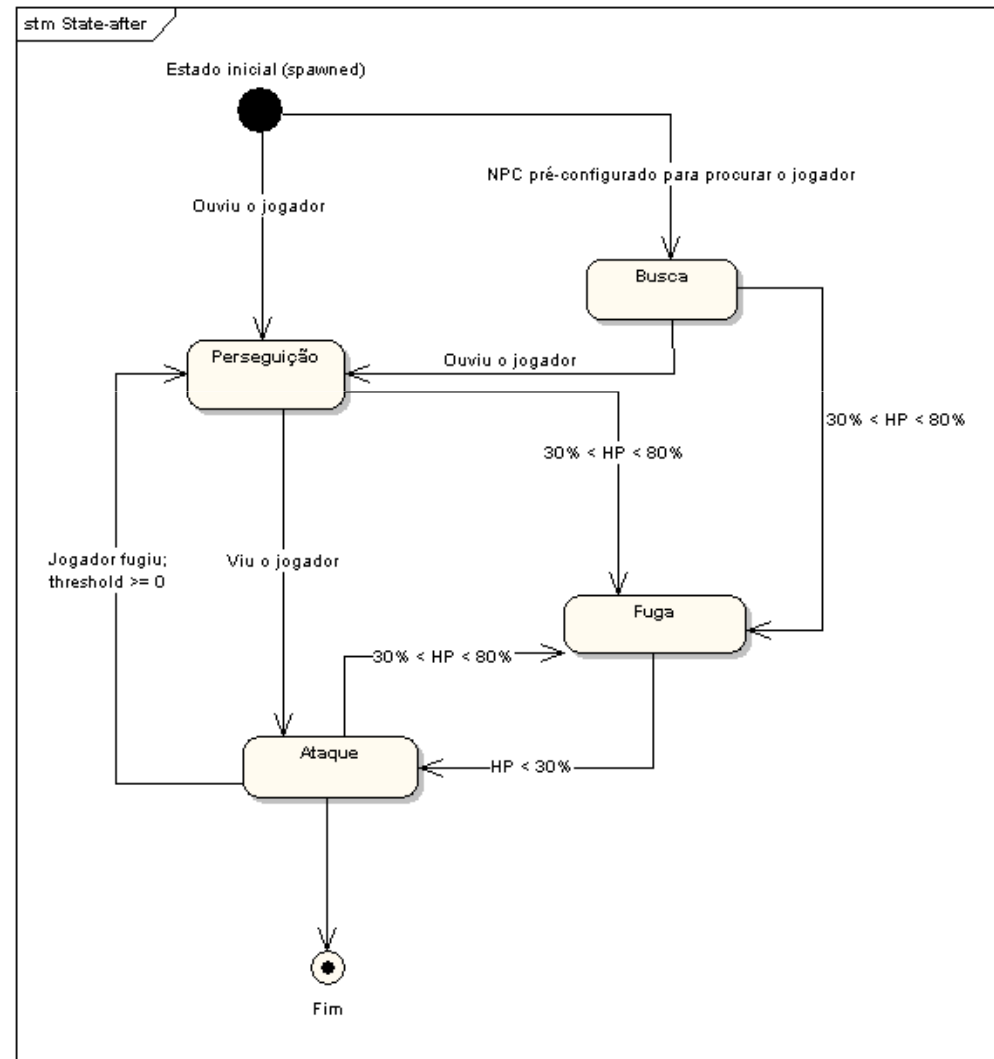
Desenvolvimento

- Soluções propostas:
 - Criação de uma rotina de fuga. O NPC se afasta do jogador quando está dentro de uma faixa do valor total de seus *hit points*.
 - Criação de rotina para que o NPC verifique se há outro NPC na linha de fogo, mudando sua posição em caso positivo.



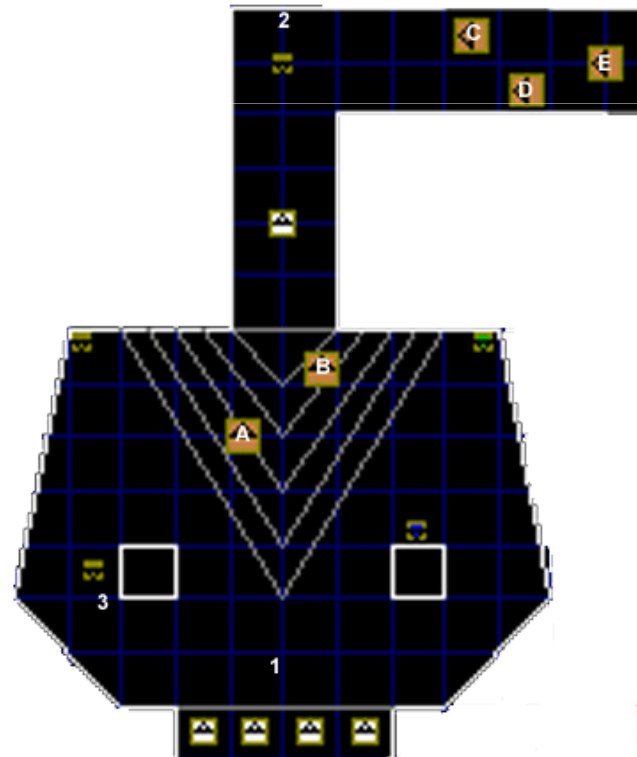
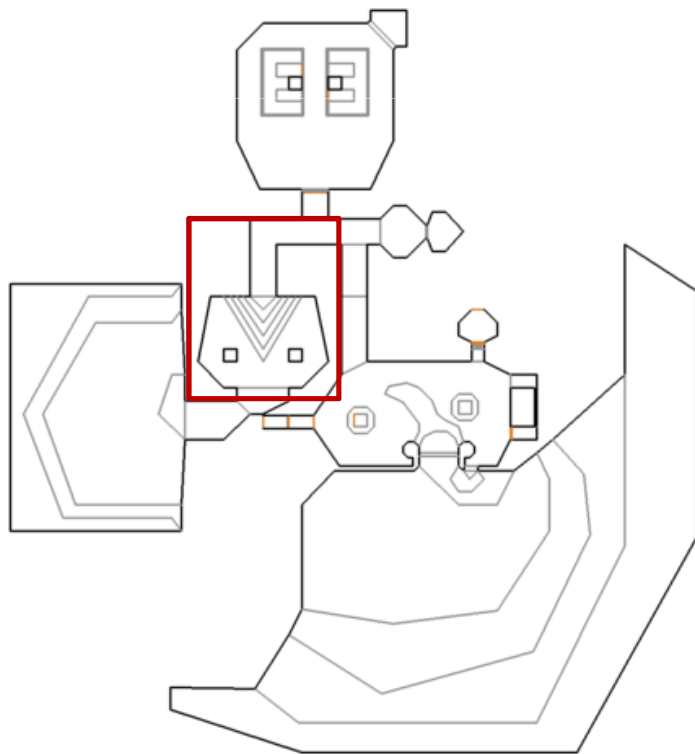
Desenvolvimento

- Gráfico que demonstra as ações possíveis do NPC em relação ao jogador, após as modificações:



Implementação

- Localidade selecionada para os testes: sala inicial do primeiro estágio de Doom 2.



Implementação

- Cenários elaborados para os testes:
 - Cenário 1: Jogador dá um tiro de alerta e fica parado no ponto 1.
 - Cenário 2: Jogador dá um tiro de alerta e anda até o ponto 2, desviando dos NPCs caso estes bloqueiem o caminho.
 - Cenário 3: Jogador dá um tiro de alerta e refugia-se no ponto 3, enfrentando os NPCs.



Implementação

- Modificações efetuadas exclusivamente para auxiliar nos testes:
 - Redução dos *hit points* do NPC para 10;
 - Redução do dano da arma do jogador para 1;
 - Registro das coordenadas do jogador e do NPC cada vez que um NPC se move ou é atingido/morto.



Resultados

- As coordenadas capturadas dos NPCs e do jogador alimentaram tabelas, que originaram gráficos, representando a movimentação dos atores no cenário para cada uma das versões do jogo.

Os gráficos permitem comparar as duas versões quanto ao comportamento dos NPCs.

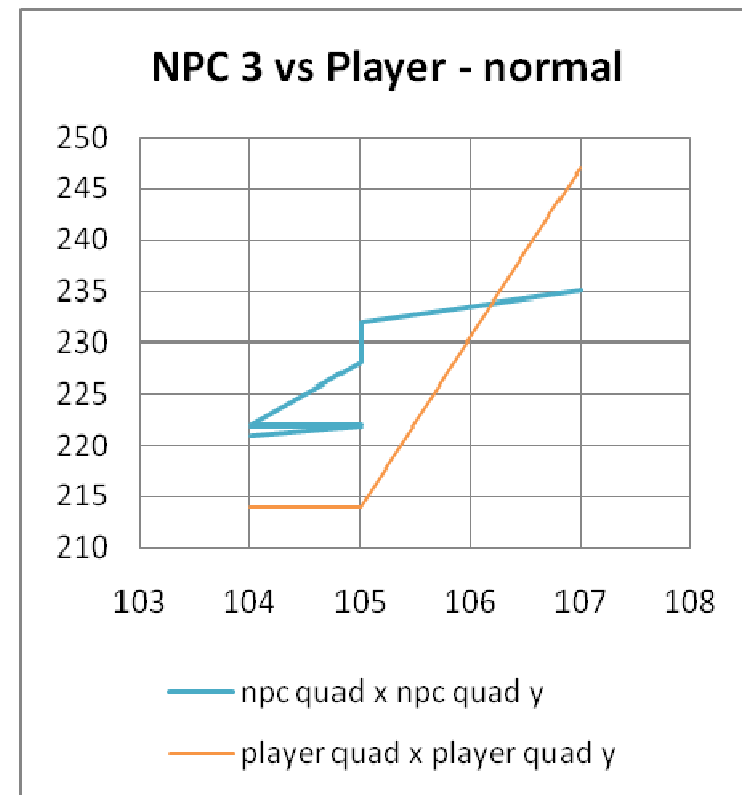
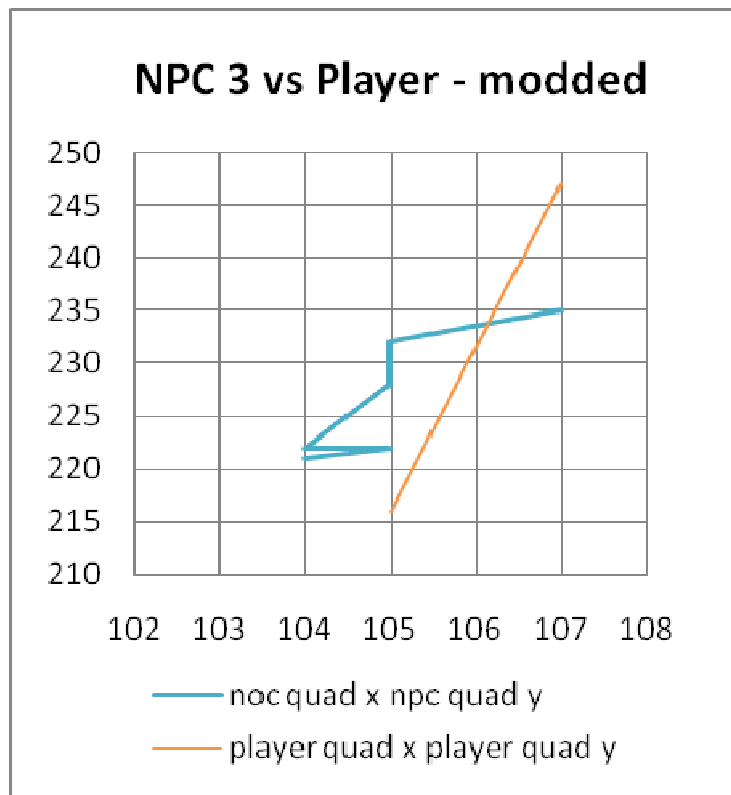


Resultados

- Cenário 1:
 - Com o jogador imóvel, os NPCs ficam constantemente se movendo e atacando o jogador.
 - Um NPC só morreria caso fosse atacado por outro NPC.
 - Em todos os testes efetuados para o cenário, os NPCs levaram mais tempo para morrer na versão modificada.
 - Não percebeu-se diferença significativa na movimentação do NPC.

Resultados

- Cenário 1 - gráficos

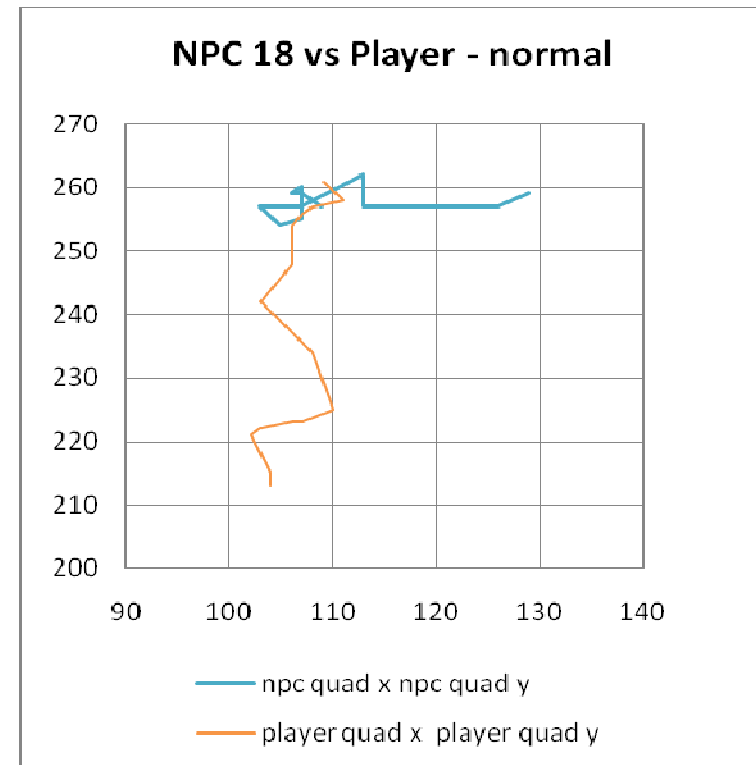
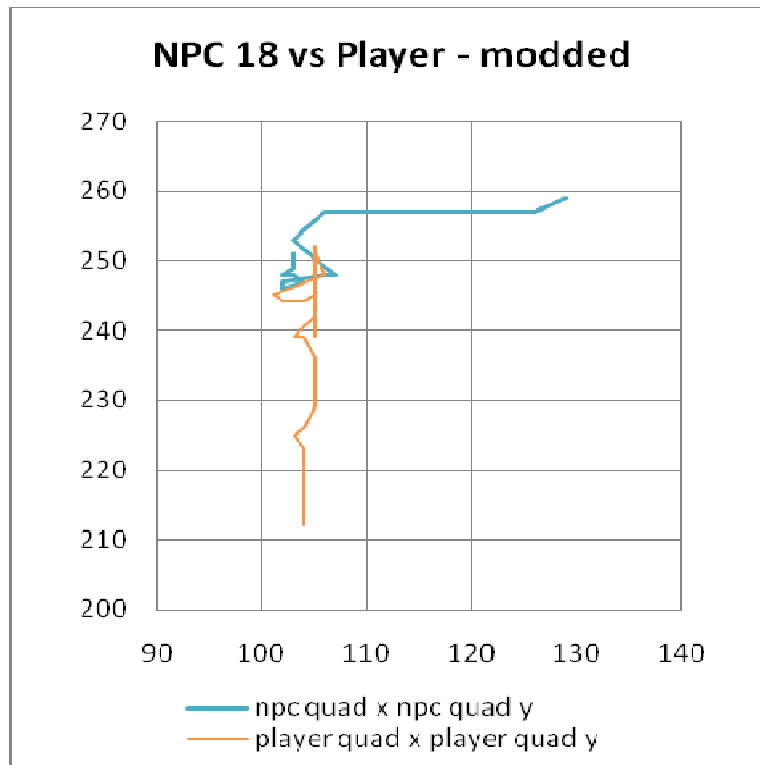


Resultados

- Cenário 2
 - Jogador se movimentava, mas não atacava. Trajetória do jogador dificilmente se repetia, pois o caminho ocasionalmente era bloqueado pelos NPCs.
 - Como o movimento do jogador não seguia um padrão, os NPCs se ajustavam diferentemente a cada teste para atacar o jogador.
 - NPCs mataram menos de seus aliados na versão modificada.

Resultados

- Cenário 2 - gráficos

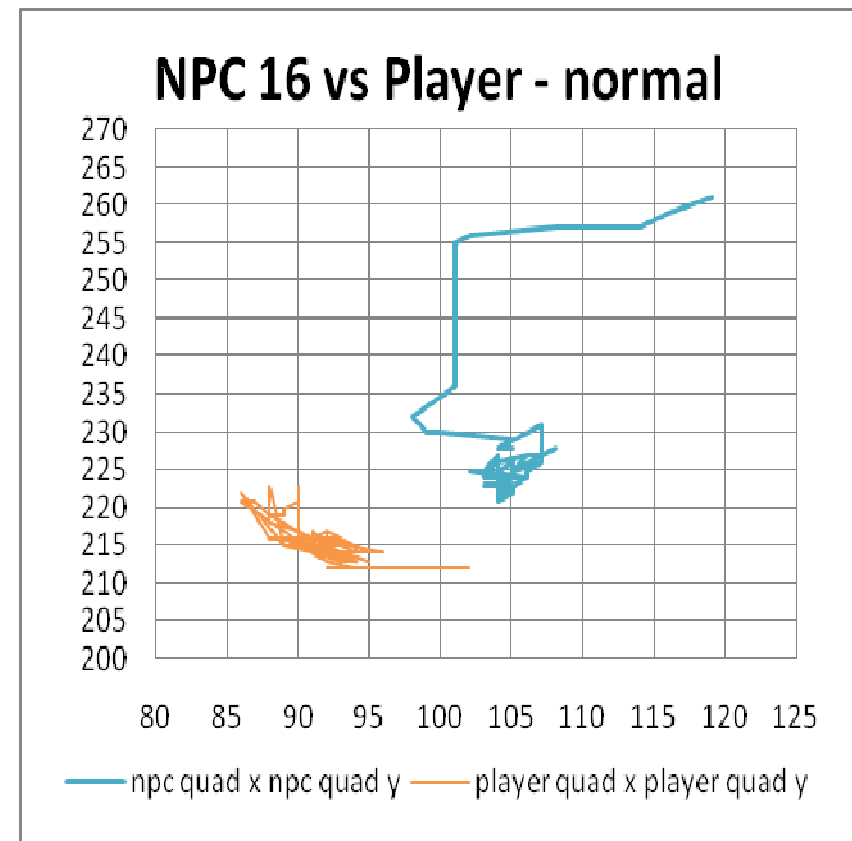
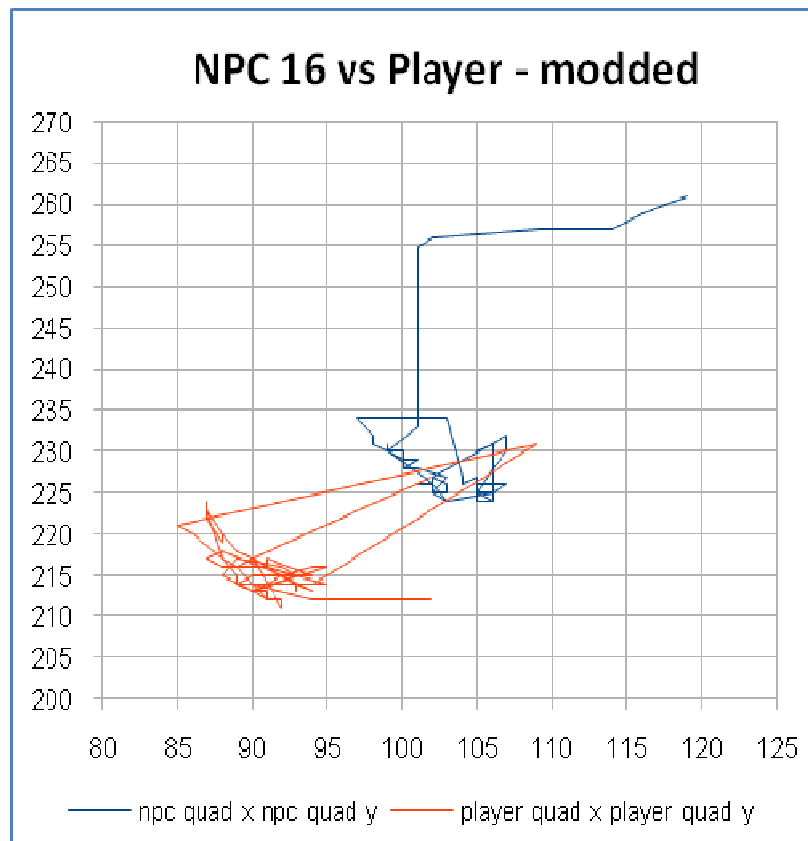


Resultados

- Cenário 3
 - Jogador toma postura ofensiva, atirando e buscando abrigo.
 - Rotina de fuga demonstra influência no comportamento do NPC.
 - Não houve diferença conclusiva quanto à *team kills* por parte dos NPCs nas duas versões.

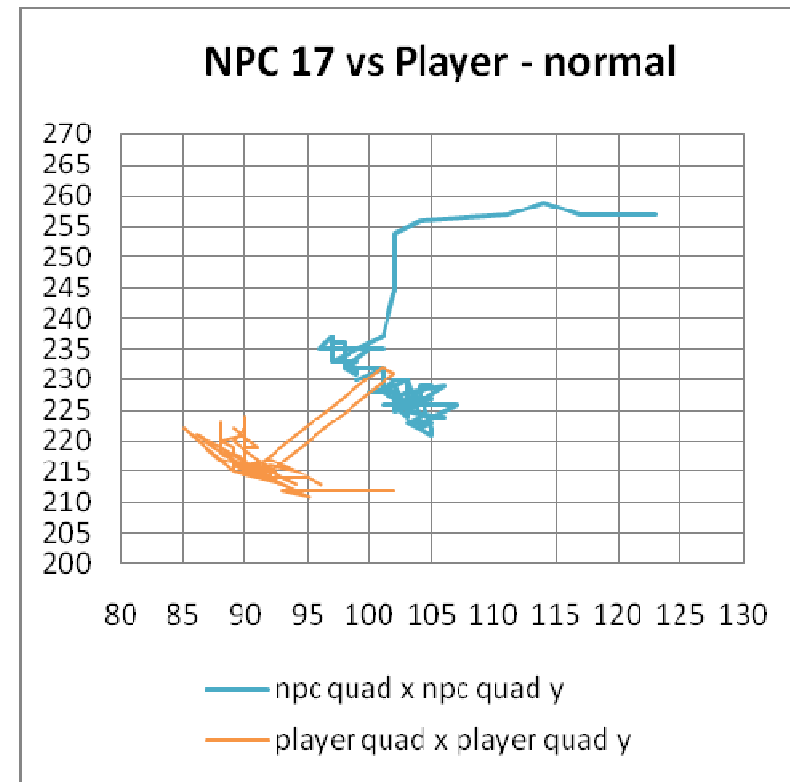
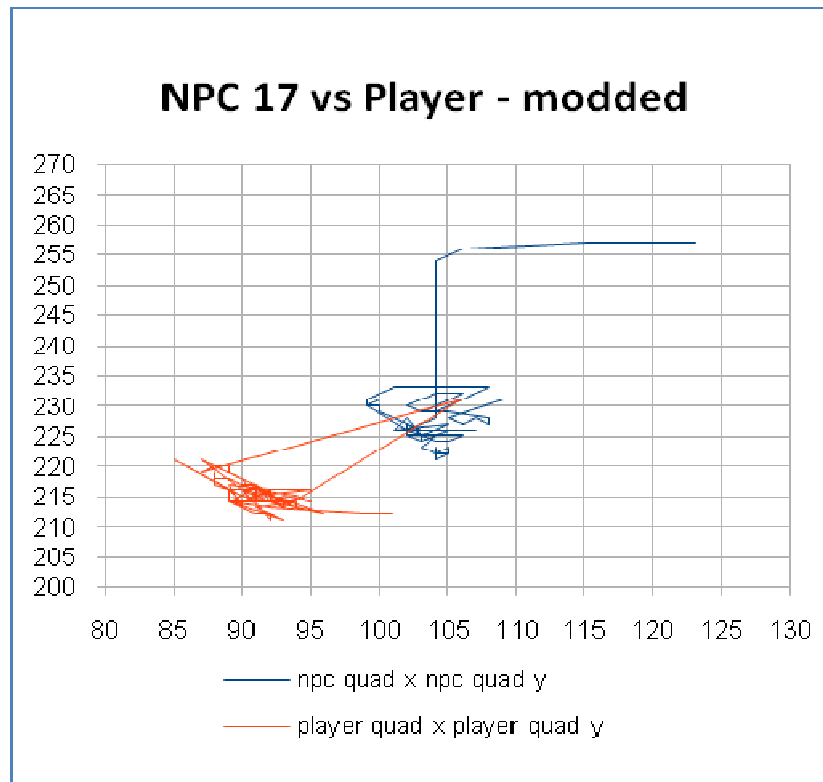
Resultados

- Cenário 3 - gráficos



Resultados

- Cenário 3 - gráficos



Resultados

- Objetivos alcançados
 - Efetuada engenharia reversa do código fonte.
 - Elaborado método comparativo entre a versão modificada e a original.
 - Conferiu-se, visualmente e pelos gráficos, alteração no comportamento dos NPCs quanto às situações selecionadas para se trabalhar.



Conclusão

- Objetivos atingidos.
- A alteração de jogabilidade proporcionada pode ser melhor utilizada com o uso de um WAD específico.
- Trabalho exigiu aprendizado em outro sistema operacional.
- Dificuldades:
 - Código mal documentado/organizado.
 - Configuração do ambiente de trabalho.



Extensão

- Aprimorar as rotinas implementadas;
- utilização de WAD concebido para melhor aproveitar as modificações;
- implementação de outras técnicas de IA.

