O Duelo

Mario de Castro - Humberto Vieira - William de Alvelos
São Paulo - SP, Brasil

Abstract

Para a disciplina de Projeto Integrador III foi proposto o desenvolvimento de um jogo que utilize visão computacional como principal meio de interação do usuário. O jogo proposto, chamado "O Duelo", consiste de um jogo de ação para dois jogadores que, utilizando duas câmeras para captar o movimento dos jogadores separadamente, busca simular a tensão de um duelo ao estilo velho oeste. O jogo foi escrito na linguagem C e utiliza a biblioteca gráfica Allegro5 assim como a biblioteca de visão computacional OpenCV. O seguinte artigo busca detalhar o processo de desenvolvimento do projeto com ênfase nos algoritmos de visão computacional utilizados, assim como ressaltar dificuldades e desafios encontrados durante todas as etapas.

Palavras Chave: Visão Computacional. OpenCV. Velho Oeste. Duelo.

1 Introdução

O jogo aqui descrito foi elaborado para disciplina de Projeto Integrador III, ministrada pelo Prof. Marcelo Hashimoto no terceiro semestre do curso de Bacharelado em Ciência da Computação do Centro Universitário Senac.

Foi requerido aos alunos da disciplina o desenvolvimento de um jogo em C que utilizasse visão computacional, através da biblioteca OpenCV, como principal meio de interção do jogador.

Para o trabalho descrito neste artigo o grupo tinha como objetivo criar um jogo para dois jogadores onde a visão computacional fosse utilizada para captar o movimento de um jogador, utilizando este movimento de forma passiva, com o objetivo de simular a tensão de um duelo ao estilo velho oeste.

O jogo utiliza duas cameras para captar as interações dos dois jogadores separadamente. Ao iniciar o jogo o jogador deve permanecer imóvel de frente para o adversario até receber um aviso do jogo que ele pode se mover. As

duas câmeras devem estar entre os jogadores, cada uma apontada para um jogador.

Ao escutar o aviso os jogadores devem reagir o mais rapido possivel para apontar uma réplica de arma segurada por eles para o adversário e atirar. As replicas das armas possuem leds nas pontas que acenderem ao pressionar o gatilho e permitirem ao algoritmo do jogo detectar o tiro. O jogador que atirar primeiro após o aviso ganhará a rodada. O jogador que se mover antes do aviso perderá a rodada.

O jogo dará *feedbacks* visuais aos jogadores informando a rodada atual, quantas rodadas cada jogador ganhou, quem ganhou a ultima rodada, quem ganhou o duelo e quando é permitido se mover.

No jogo são utilizados dois algoritmos de visão computacional principais: um para detecção do movimento do jogador e outro para reconhecimento do led das replicas das armas. Os dois algoritmos serão descritos em maiores detalhes na próxima seção.

2 Desenvolvimento

2.1 Detecção de Movimento

Para realizar a detecção de movimento dos jogadores utilizou-se uma versão de um algoritmo que compara o frame atual da câmera com o frame imediatamente anterior em busca de diferenças entre seus pixels. caso a diferença seja grande o suficiente o movimento é registrado.

Ao iniciar o jogo o primeiro frame da câmera é salvo em uma matriz de vetores *matriz1*, onde cada vetor representa os valores RGB de cada pixel da imagem. No decorrer do jogo cada novo frame da câmera é então salvo em uma segunda matriz de vetores RGB *matriz2*.

Em seguida o algoritmo percorre todos os campos de cada matriz e compara os valores R (vermelho), G (verde) e B (azul) de cada vetor da *matriz1* com seu vetor correspondente da *matriz2*. Caso o valor de algum dos campos R, G ou B do vetor da primeira matriz defira do valor RGB da segunda matriz para mais ou para menos de que um certo *RANGE* previamente definido considera-se que houve mudança naquele pixel. Por exemplo, caso seja definido que o *RANGE* é igual a 20 e o valor R de um pixel da *matriz1* é igual a 100, só será identificado que houve mudanca no pixel se o valor do pixel correspondente na *matriz2* for maior que 120 ou menor que 80.

Para cada pixel definido como diferente é somado 1 a um contador de pixels diferentes q que é iniciado com valor 0, e ao final do algoritmo ter percorrido toda a matriz tera a quantidade de pixels diferentes entre as duas matrizes. Caso essa quantidade q seja maior que uma valor *sensibilidade* a ser definido, é considerado que houve movimento entre os frames. O valor da *sensibilidade* é determinado pela divisão entre quantidade de pixels *qpixels* da matriz por um valor *razão* pré definido. de forma que *sensibilidade* = *qpixels* / *razão*. quanto maior o valor de *razão*, menor será a quantidade de pixels diferentes necessários para identificação de movimento. Então temos que se q for maior do que *sensibilidade* o movimento é identificado, caso contrario não houve movimento.

A precisão do algoritmo pode ser facilmente controlada alterando-se o valor do *RANGE* que determina se houve mudança em um pixel ou o valor de *razão*, que determina a quantidade de pixels que precisam ser diferentes para que o movimento seja identificado.

2.2 Detecção de Luz

3 Considerações Finais