

Projeto Interativo III

Angry Robots



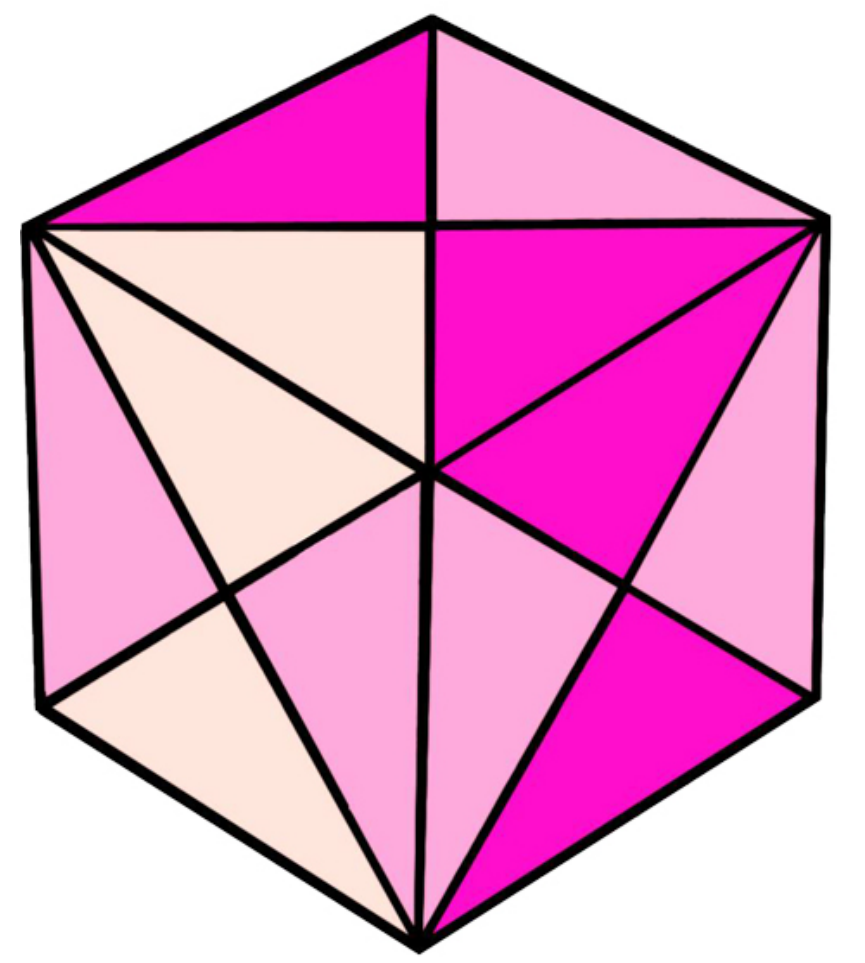
Caroline Bomfim do Espirito Santo, Mahaira Soares de Souza, Rafael da Silva Santos, Thiago de Sousa Messias

Bacharelado em Ciência da Computação

Centro Universitário SENAC - Campus Santo Amaro (SENAC-SP)

Av. Engenheiro Eusébio Stevaux, 823 – Santo Amaro, São Paulo – CEP 04696-000 – SP – Brasil

caroline.bomfim@hotmail.com.br, mahaira.souza@hotmail.com,
rafa-silva.santos@hotmail.com, messiasth@gmail.com



1. Resumo

Angry Robots é um jogo desenvolvido em linguagem C[1], utilizando a biblioteca gráfica allegro 5[2], fundamentado em visão computacional aplicando inderetamente o "OpenCV"[3]. Um de seus principais objetivos é induzir o usuário a trabalhar sua mente e desenvolver seu raciocínio lógico, com estratégia e agilidade para derrotar o robô. A parte de visão computacional foi construída com algoritmos em função do cálculo da centróide e o sistema de cores HSV (formado pelas componentes Hue (tonalidade), Saturation (saturação) e Value (valor)).

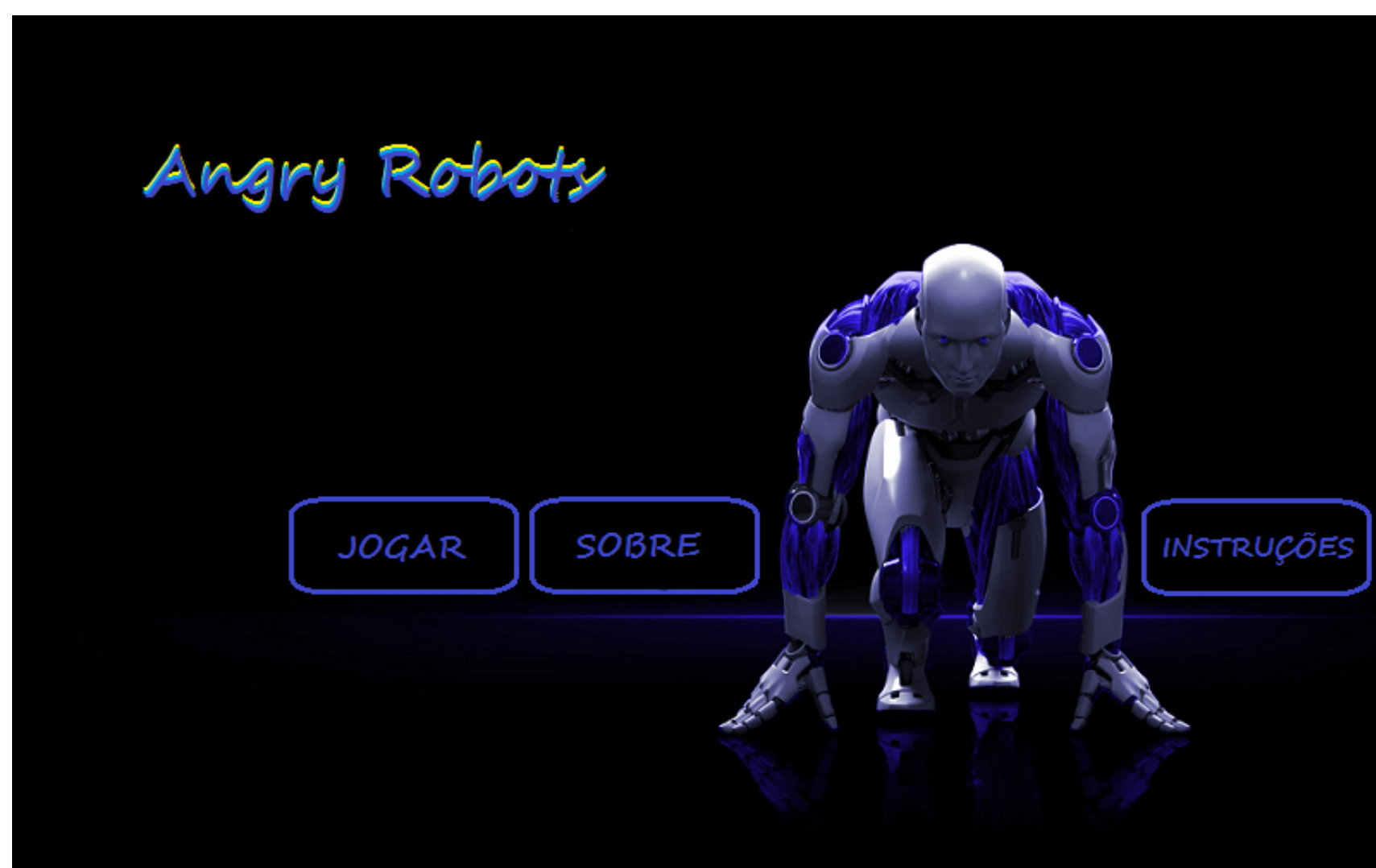


Figura 1: Menu do jogo.

2. Introdução

Nos últimos anos a indústria de jogos cresceu de forma constante, e com ela a importância dos mesmos para o desenvolvimento de muitas habilidades humanas. Sabemos que a maioria dos jogos são desenvolvidos com intuito de estabelecer uma conexão entre o mundo virtual e o real, para que desta forma possa proporcionar ao jogador tanto uma diversão quanto o desenvolvimento de suas habilidade pessoais, como agilidades, raciocínio lógico, estratégia, entre outros, assim lhe ocasionando um exercício mental.

Angry Robots, faz com que o jogador se movimente de forma rápida e lógica para que consiga chegar ao objetivo final, ganhar a partida e derrotar o robô.

Em relação a visão computacional, quando tratada a imagem, utiliza-se algoritmos e fórmulas matemáticas, e principalmente a física, fundamental para calcular a frequência e tonalidade de cores, a iluminação e luminosidade do local, de maneira que possamos descobrir em qual escala de cor se encontra e melhorar a qualidade visual do jogo.



Figura 2: Versão beta do jogo, com design teste.

3. Revisão de Literatura

Angry Robots é um jogo limitado, pois dentro das exigências foram permitidos somente as bibliotecas, allegro 5 graficamente e a multiplataforma OpenCV utilizada para desenvolvimento de aplicativos na área de visão computacional.

Existem diversos jogos em que o Angry Robots foi baseado, um deles foi o Cube Slam, um jogo em que os usuários se enfrentam numa partida visual de air hockey, no qual o jogador luta contra um urso, já no Angry Robots o adversário é um robô rápido e ágil.

Há projetos no blog "Laboratório Garagem", entretanto todos fugiam do ambiente proposto, a maioria deles utilizavam Arduino, OpenCV e Python. Porém, mesmo com os obstáculos impostos no Angry Robots, foi de grande utilidade pois algumas ideias surgiram através de post's referentes à robôs.

Gran Slam Tennis 2, jogo que simula os grandes campeonatos de tênis, foi utilizado como base para os movimentos do jogo.

Os algoritmos desenvolvidos são autorais sem base em outros jogos, visto que mesmo com fundamento em tantos jogos, Angry Robots é um diferencial.

4. Desenvolvimentos

Centróide

É o ponto interior que define seu centro geométrico, caso a forma geométrica represente uma secção homogênea de um corpo, então o centroide coincide com o centro de massa. A mesma foi implementada no Angry Robots para captar o meio da tela, facilitando a detecção e visualização do restante do display para uma variação é identificação da cor de preferência, no caso o azul.

```
Centróide.h
73 int Centroid(unsigned char ***imagem, int altura, int largura, int *coordenada){
74     int px = coordenada[0];
75     int py = coordenada[1];
76     int markx = 0, marky = 0, cn = 0;
77     int h, s, v;
78     for(int a = 0; a < altura; a++){
79         for(int b = 0; b < largura; b++){
80             RGB2HSV(
81                 imagem[a][b][0],
82                 imagem[a][b][1],
83                 imagem[a][b][2],
84                 &h, &s, &v
85             );
86             if((h > 226 && h < 250) && s > 85 && v == 100){
87                 markx ++ a;
88                 marky ++ b;
89                 cn++;
90             }
91         }
92     }
93     if( cn > 0){
94         coordenada[0] = markx / cn;
95         coordenada[1] = marky / cn;
96     }
97     if(px == coordenada[0] && py == coordenada[1]){
98         return 0;
99     }
100     return 1;
101 }
102 }
103 #endif
104
```

Figura 3: Exemplo: Código Centróide.

Conversão para HSV

Calcula a intensidade da tonalidade, saturação e brilho da imagem, possibilitando aprimoramento no rastreamento do código, para resolver problemas com a iluminação do local, assim, não atrapalha a jogabilidade. A tonalidade permite distinguir as cores puras de 0 à 360 graus, a saturação verifica a intensidade da pureza da tonalidade, o brilho verifica a iluminação da imagem.

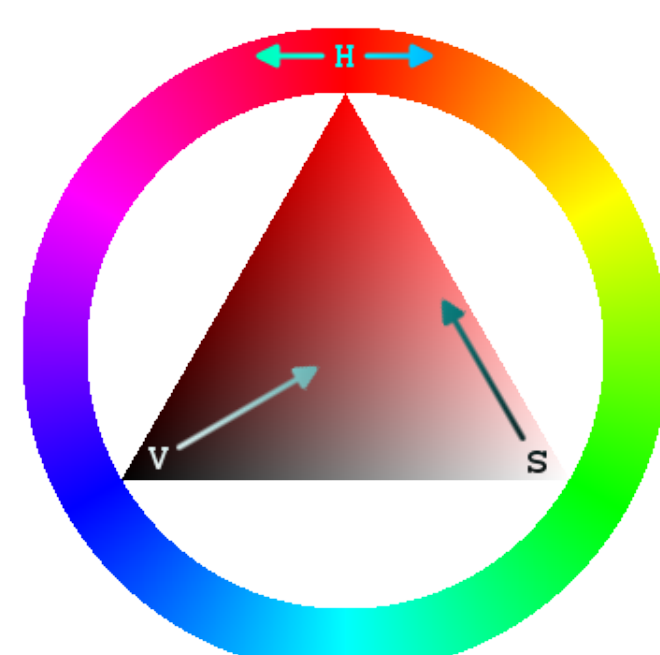


Figura 4: Exemplo: triângulo de explicação HSV

Histograma

Um histograma, conhecido também como distribuição de frequências. Aplicado no código para captar os tons de azul, onde todos os valores abaixo da media automaticamente são zerados e os maiores maximizados, para que se encaixem no tom rastreado, calculando assim o centro.

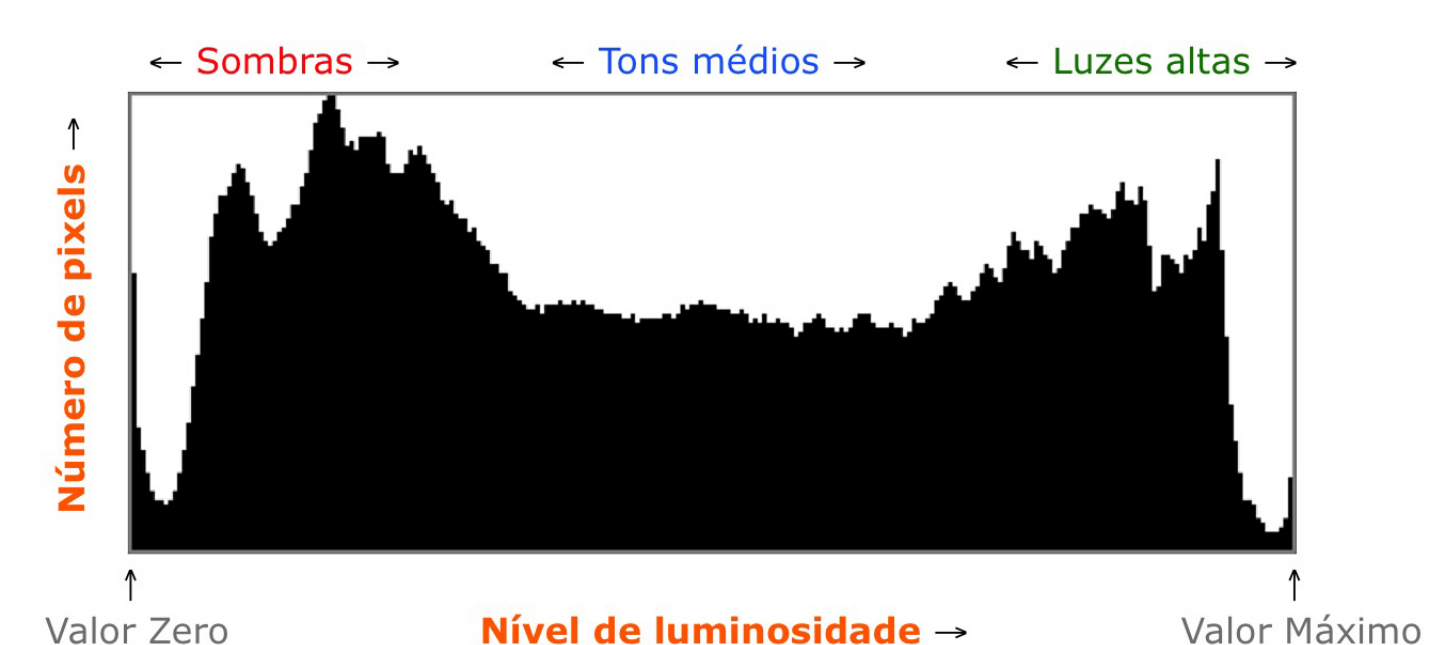


Figura 5: Exemplo: Nível de frequência da cor.

5. Resultados

Neste jogo o robô ira tentar fugir para que o jogador não consiga acertá-lo com pequenas bolas, isto será feito através de um laser azul, o objetivo do jogo é derrotar o robô jogando as bolas em sua direção, quanto mais rápido os movimentos do usuário, mais chances de vencer o jogo.



Figura 6: Imagem do jogo funcionando

6. Considerações Finais

O rastreamento é o ponto mais difícil, pois alguns fatores atrapalharam o desenvolvimento do mesmo, um deles foi a iluminação, pois devido a ela, a imagem pode ser ofuscada ou obscurecida demais, porém o HSV permitiu que a iluminação fosse ignorada, pois este algoritmo converte toda a imagem para cinza, e trata a variação da luminosidade, deste modo o usuário poderá jogar com uma camiseta azul por exemplo, sem afetar o rastreamento.

Referências

- [1] Portal R7 de notícias
<http://noticias.r7.com/tecnologia-e-ciencia/noticias/jovens-apelam-para-games-corporais-para-melhorar-condicionamento-fisico-20110527.html>
- [2] H. M. Deitel e P. J. Deitel. Como programar em C, 2ª edição.
- [3] <http://www.rafaeltoledo.net/tutoriais-allegro-5/> Allegro 5.
- [4] <http://opencv.org/> OpenCv
- [5] http://www.rapidtables.com/web/color/RGB_Color.htm RGB