

Projeto Interativo III

Angry Robots



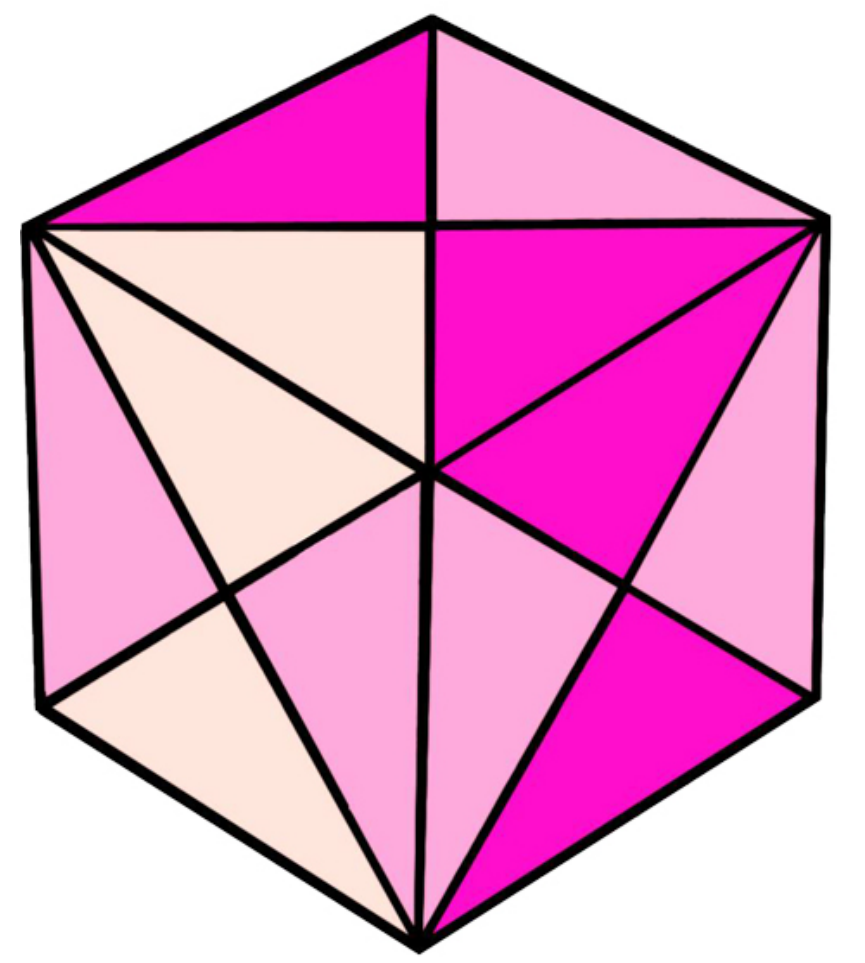
Caroline Bomfim do Espirito Santo, Mahaira Soares de Souza, Rafael da Silva Santos, Thiago de Sousa Messias

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, messiassthi@gmail.com



1. Resumo

Angry Robots é um jogo desenvolvido em linguagem C, utilizando a biblioteca gráfica allegro 5, fundamentado em visão computacional "OpenCV". Um de seus principais objetivos é induzir a atividade física, fazendo com que o usuário se movimente para conseguir derrotar um 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)).

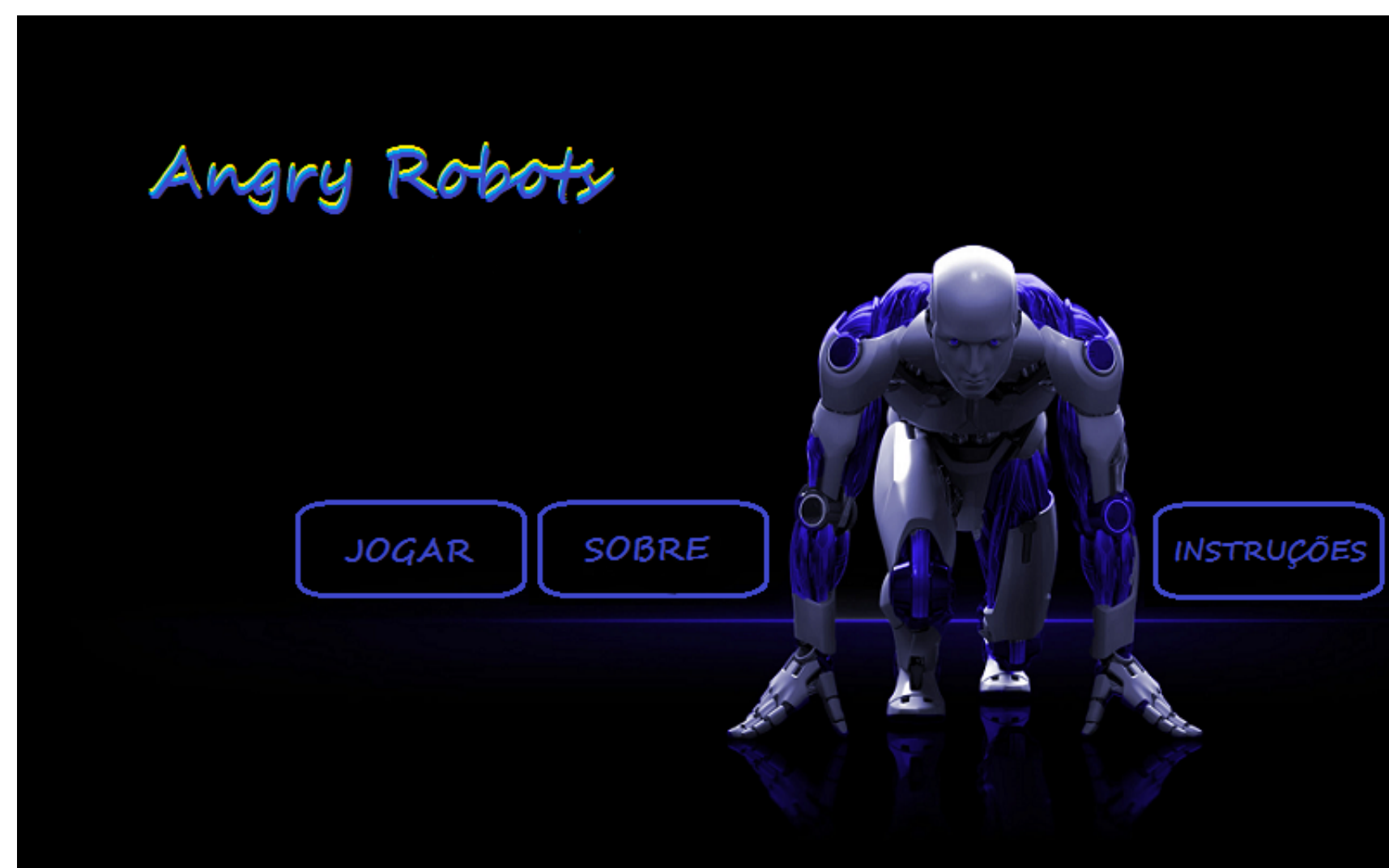


Figura 1: Versão beta do jogo, com design teste.

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. Angry Robots, o mesmo induz que o jogador se movimente de forma rápida e lógica para que consiga chegar ao objetivo final, ganhar a partida e derrotar o robô. Quando tratado à imagem é utilizado algoritmos e fórmulas matemáticas, inclusive e principalmente a física para que seja calculado a frequência e tonalidade de cores de cada pixel's, iluminação e luminosidade do local, tanto para sabermos em qual escala de cor ele se encontra, quanto para melhorar a qualidade visual.



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

3. Revisão de Literatura

Angry Robots é um jogo limitado, pois dentro dos requisitos fora permitidos somente as bibliotecas, allegro 5 gráfica e

a multiplataforma OpenCV utilizada para desenvolvimento de aplicativos na área de visão computacional. Existem diversos jogos em que o Angry Robots foi baseado, como no Cube Slam, um jogo que os usuários se enfrentam em uma partida visual de air hockey, onde o jogador luta contra um urso, no caso do Angry Robots o adversário é o robô rápido e inteligente. Também existiam projetos no blog "Laboratório Garagem", entretanto todos fugiam do ambiente proposto, a maioria deles utilizavam Arduino, OpenCV e Python, mesmo com os obstáculos foi de grande utilidade pois algumas ideias surgiram através de post's referentes à robôs. Gran Slam Tennis 2, foi utilizado em questão dos movimentos do jogo, e o enredo em si, do MechWarrior. Os algoritmos desenvolvidos são autorais sem base em outros jogos, visto que mesmo com fundamento em tantos jogos, Angry Robots é um diferencial de jogos.

4. Desenvolvimentos

Centróide é nome dado ao ponto interior que define seu centro geométrico. Caso a forma geométrica represente uma seção homogênea de um corpo, então o centróide coincide com o centro de massa. Nos casos em que não só o corpo é homogêneo e está submetido a um campo gravítico constante, então esse ponto coincide com o centro de gravidade. A centróide é implementada no código para captar o meio da tela, facilitando a detecção do resto da tela para variação é identificação da cor de preferência.

```
Centroid.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 #endif
103
104
```

Figura 3: Exemplo: Código Centróide.

Conversão para HSV calcula a intensidade da tonalidade, saturação e brilho da imagem, isto possibilitou um aprimoramento no rastreamento do jogo, pois a iluminação, e os objetos ao redor poderão ser ignorados, não atrapalhando a jogabilidade. A tonalidade permite distinguir as cores puras de 0 a 360 graus, a saturação verifica a intensidade da pureza da tonalidade, o brilho verifica a iluminação da imagem.

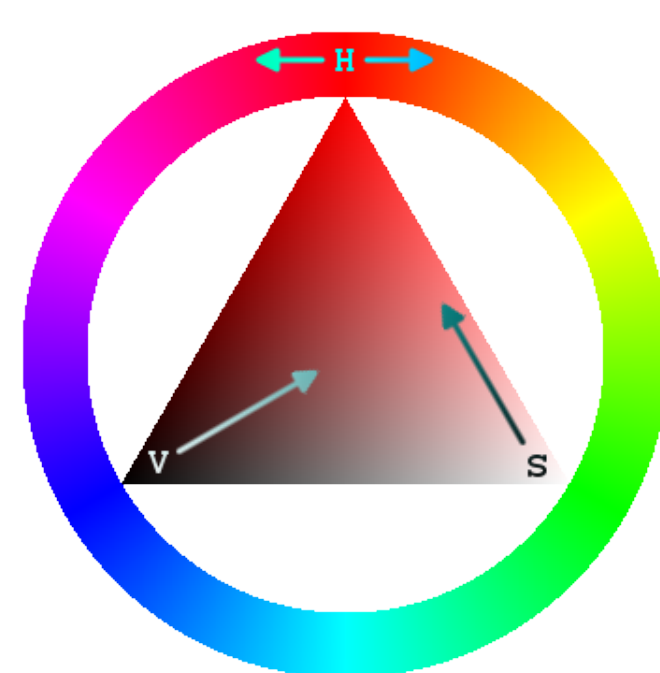


Figura 4: Exemplo: triângulo de explicação HSV

Ponto de maior reflexão de pixels

Nível de saturação

Histograma

Em estatística, um histograma, também conhecido como distribuição de frequências ou diagrama das frequências, é uma representação gráfica na qual um conjunto de dados é agrupado em classes uniformes, representado por um retângulo cuja base horizontal são as classes e seu intervalo e a altura vertical representa a frequência com que os valores desta classe estão presente no conjunto de dados.

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 5: 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.](http://www.rafaeltoledo.net/tutoriais-allegro-5/Allegro%205)
- [4] <http://opencv.org/>
OpenCv
- [5] http://www.rapidtables.com/web/color/RGB_Color.htm
RGB