

Implementação do Jogo “Color Ball 2”

Fazer um programa que implementa o jogo Color Ball 2. O objetivo do jogo é capturar todas as bolinhas da mesma cor da bolinha atual do jogador. Na figura 1, a bolinha atual do jogador é de cor amarela e aparece de forma destacada das demais.

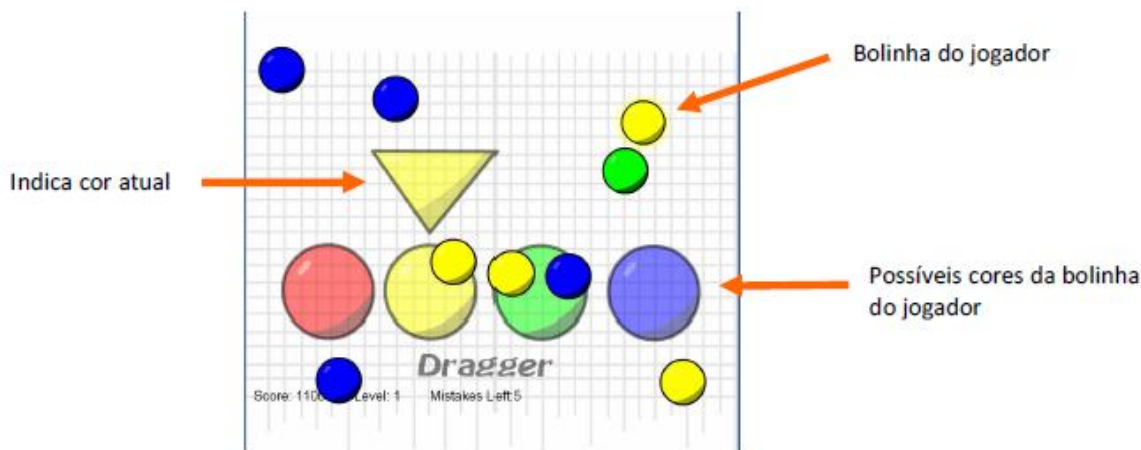


Figura 1: Área de jogo

1) Tela inicial

O jogo deve possuir uma tela inicial, que deve ser apresentada ao jogador a cada execução do programa. Além do nome do jogo, devem constar as opções: jogar, exibir *hank* e *help-online*.

2) Tela final

O jogo encerra com a apresentação da pontuação do jogador e sua colocação no *rank* do jogo. Deve haver uma opção onde o jogador pode decidir entre jogar novamente ou sair.

3) Bolinha do jogador

Deve ser desenhada de forma destacada em relação às demais bolinhas (contorno, sombreado, ...). Deve ser controlada exclusivamente pelo *mouse*. Seus movimentos devem estar limitados à área de jogo.

A bolinha do jogador pode assumir as seguintes cores: vermelho, amarelo, azul e verde. Cada cor de bola apresenta uma característica diferente, conforme segue:

- Amarela (Dragger): tem o funcionamento normal.
- Vermelha (Hyper Ball): a velocidade de todas as bolinhas que surgem na área de jogo deve ser aumentada.
- Azul (Inviso Ball): a bolinha do jogador assume uma coloração azul quase invisível.
- Verde (Expander): a bolinha do jogador aumenta seu tamanho, dificultando bastante o jogo.

4) Bolinhas que surgem

Devem ter uma determinada velocidade, tamanho, cor, os quais devem ser gerados de forma aleatória a cada nível. Sua direção é horizontal e seu sentido é da esquerda para a direita. A cada colisão com a bolinha do jogador, cada bolinha deve aumentar um pouquinho de tamanho e então desaparecer. Deve haver uma indicação de que a bolinha foi capturada correta (✓) ou incorretamente (X). As bolinhas que saem da tela à direita, caso sejam da mesma cor do jogador, devem ser indicadas como não capturadas (X). O programa deve suportar ao menos dez bolinhas simultâneas na tela.

5) Cor atual

Na área de jogo devem estar exibidas as possíveis cores da bolinha do jogador, assim como a indicação da sua cor atual, conforme mostram as figuras 1 e 2. Momentos antes da alteração de cor, o jogador deve ser avisado (conforme exibe a figura 2). Desta forma, a cada alteração de cor, esta informação deve ser atualizada. As alterações de cor podem ocorrer dentro de um mesmo nível.

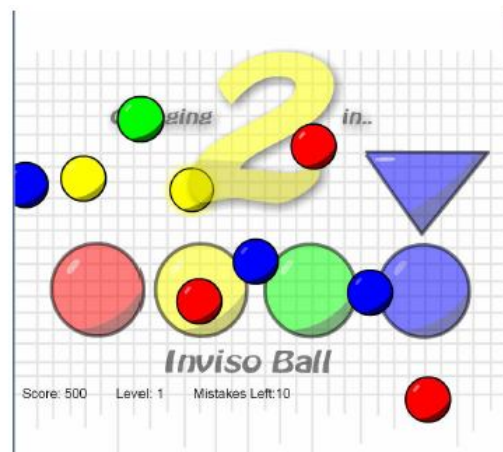


Figura 2: Exibe a contagem regressiva para a alteração de cor

6) Mudança de nível

A mudança de nível pode estar vinculada tanto a uma determinada pontuação, como a uma determinada quantidade de alterações de cor efetuadas.

7) Pontuação

Na área de jogo, deve ser apresentado o quadro de pontuação, onde devem estar o nível atual, a pontuação e o número de vidas restantes do jogador.

O jogo deve ser composto por **níveis**, onde a cada novo nível, a dificuldade do jogo aumenta (velocidade, tamanho, cor ou posição das bolinhas que surgem devem variar a cada nível).

Para fins de **pontuação**, a cada bolinha capturada, o jogador soma 100 pontos.

Ao iniciar o jogo, o jogador possui 10 (dez) **vidas**. Caso o jogador não seja bem sucedido e a sua bolinha entre em contato com uma bolinha diferente da esperada, o número de vidas restantes do jogador é decrementado. Quando este número atinge zero, o jogo termina. (nas figuras 1 e 2 essa informação aparece como *Mistakes Left*). O número de vidas do jogador deve ser exibido na tela do jogo.

8) Record

Deve ser implementado o armazenamento e recuperação de *records* anteriores, bem como a posição do jogador no *rank* de jogadores. Estas informações podem ser exibidas através do menu inicial e ainda ao final de cada jogo.

9) Help-online

O jogo deve possuir uma opção de *help-online*, o qual deve explicar o jogo, assim como suas opções.

Questões de ordem:

- O trabalho pode ser realizado individualmente ou em dupla.
- O jogo deve conter um arquivo *readme* com instruções para instalação e execução do jogo.
- O trabalho deve ser apresentado à professora, por todos os integrantes do grupo, em data e horário definidos e divulgados pela professora.
- Datas importantes:
 - **29/10/2015:** enviar e-mail com a definição da dupla ou informação de que vai realizar o trabalho individualmente (assunto Lab1-grupo);
 - **10/11/2015:** entrega da versão atual do trabalho, através de e-mail (pitthan@inf.ufsm.br), com assunto Lab1-previa-aluno1-aluno2, anexando o(s) arquivo(s) fonte do trabalho em andamento;
 - **26/11/2015:** prazo final para entrega - enviar e-mail, com assunto Lab1-tfinal-aluno1-aluno2, anexando o(s) arquivo(s) fonte(s) comentado(s) (.h, .c ou .cpp, **não o executável !!!**) e o arquivo *readme*;
 - **30/11 a 04/12/2015:** apresentações.
- O não cumprimento à especificação, bem como aos prazos estipulados, invalida a avaliação do trabalho.
- É fortemente recomendado que o aluno experimente jogar o jogo original, o qual se encontra em diversos sites, como por exemplo:
<http://www.aulavaga.com.br/jogo/color-ball-2.html>