

Projeto: Futebol de botão

Especificação de requisitos de Software

Versão 1.0

24/02/2021

Versão	Autores	Data	Ação
1.0	Augusto Silva de Oliveira Eduardo Bruggmann Pedro Gustavo Coelho da Costa	24/02/2021	Estabelecimento de requisitos
1.1	Augusto Silva de Oliveira Eduardo Bruggmann Pedro Gustavo Coelho da Costa	16/05/2021	Atualização nas regras do jogo Atualização de requisitos funcionais Adição de requisitos não funcionais

Conteúdo:

1. Introdução;
2. Visão geral;
3. Requisitos de software
4. Esboço da interface gráfica;

1. Introdução

1.1 Objetivo do desenvolvimento

Desenvolver um *software standalone* que simula uma partida de futebol de botão entre dois jogadores em um tabuleiro virtual, onde os jogadores se enfrentam em turnos alternados com a possibilidade de um vencedor ou empate.

1.2 Definições do jogo

- **Campo:** representação gráfica de um campo de futebol, composto por linhas e dois gols.
- **Peão(“Botão”):** entidade que representa os jogadores em campo e pode ser controlada por um *player*.
- **Goleiro:** entidade que representa o goleiro em campo e pode ser controlada por um *player*.
- **Bola:** entidade que recebe interação dos botões com objetivo de ser lançada até o gol.
- **Gol:** parte do tabuleiro onde os *players* devem chutar a bola para que seja registrado os pontos.
- **Player:** jogador que controla as ações (movimentos) do time escolhido e representa o usuário do sistema.
- **Time:** entidade que pode ser escolhida por um *player* na qual os botões estarão representados no tabuleiro.
- **Placar:** Elemento gráfico que representa a pontuação de cada um dos *players*.
- **Eventos:** ações disparadas para realizar uma jogada, que podem ser chute, defesa ou movimentação do goleiro.
 - **Movimentação do peão:** ação em que o *player* movimenta o botão designado como peão.
 - **Movimentação do goleiro:** ação em que o *player* movimenta o botão designado como goleiro.

Regras do jogo:

O jogo é constituído por dois participantes, que alternam entre turnos para poder realizar sua jogada, com um limite de turnos definido antes da partida. Cada turno do jogador o permite dar um toque na bola com o peão e ajustar a posição do goleiro, o movimento do goleiro deve ser realizado antes do peão caso contrário não será possível mover o goleiro. O toque na bola não é obrigatório, colisões entre peões não são penalizadas e não há saídas de campo, caso o peão ou a bola toque o limite do campo eles quicam e voltam. No final da partida, o jogador que tiver mais gols é declarado o vencedor, se o número de gols for igual acontece um empate. Não é preciso declarar chutes ao gol, sendo assim eles são liberados a qualquer momento, cabe ao jogador sempre deixar seu goleiro em posição vantajosa.

2. Visão geral

2.1 Arquitetura da aplicação

Aplicação orientada a objetos, *standalone*.

2.2 Premissas do desenvolvimento

- O *software* será desenvolvido na linguagem de programação Python
- O *software* deve apresentar uma interface gráfica bidimensional
- Tanto a interface gráfica quanto o *software em si* deve ser desenvolvida com o auxílio da biblioteca Pygame
- A física aplicada ao jogo deve ser feita com auxílio da biblioteca Pymunk
- A modelagem do *software* deve ser desenvolvida com base na UML 2

3. Requisitos da aplicação

3.1 Requisitos funcionais

1. **Escolher time:** deve permitir ao usuário (*player*) escolher o time com o qual deseja jogar.
2. **Selecionar turnos:** deve permitir ao usuário (*player*) selecionar a quantidade de turnos da partida.
3. **Iniciar partida:** após cada *player* selecionar seu time e o limite de turnos, o programa deve permitir a inicialização de uma nova partida. A partida só pode ser iniciada após a seleção dos times e a quantidade de turnos.
4. **Movimentar botões:** o *software* deve permitir a movimentação dos botões por parte do usuário (*player*) através de um movimento do tipo drag-and-drop. Cada usuário pode movimentar apenas um botão e seu goleiro durante seu turno de jogada, o goleiro deve ser movimentado antes de movimentar o botão. O movimento se dá de forma similar a um taco de sinuca, onde o jogador clica no peão desejado e arrasta o mouse formando a linha que indica o ângulo de batida e a força, ao soltar o botão do mouse o peão é lançado na direção.
5. **Marcar gol:** quando a bola atravessa a linha do gol o *software* deve atualizar o placar na interface gráfica. Quando um gol é marcado, os botões devem voltar para posição inicial e a bola deve ser colocada no meio do tabuleiro.
6. **Colidir:** o *software* deve permitir que os botões colidam uns com os outros. Cada colisão gera um movimento em um ou mais botões e na bola.
7. **Empatar:** caso os turnos da partida acabem e o placar de ambos os jogadores seja igual, o *software* deve indicar para os usuários que houve um empate.

8. **Fim da Partida:** no fim da partida o jogo dá duas opções, voltar para o menu principal, onde é possível iniciar uma nova partida ou sair do jogo através do fechamento da janela.

3.2 Requisitos não funcionais

1. **Especificação do projeto:** o código do *software* deve ser desenvolvido em Python com Pygame e Pymunk e a especificação do projeto baseada em UML 2.
2. **Interface gráfica:** deve ser utilizada a biblioteca Pygame do Python para criação dos elementos gráficos.
3. **IDE de desenvolvimento:** Deverá ser utilizado o Pycharm IDE para o desenvolvimento do projeto.
4. **Física:** A física do jogo deverá ser feita com o auxílio da biblioteca Pymunk.

4. Esboço da interface gráfica

-Menu Principal:



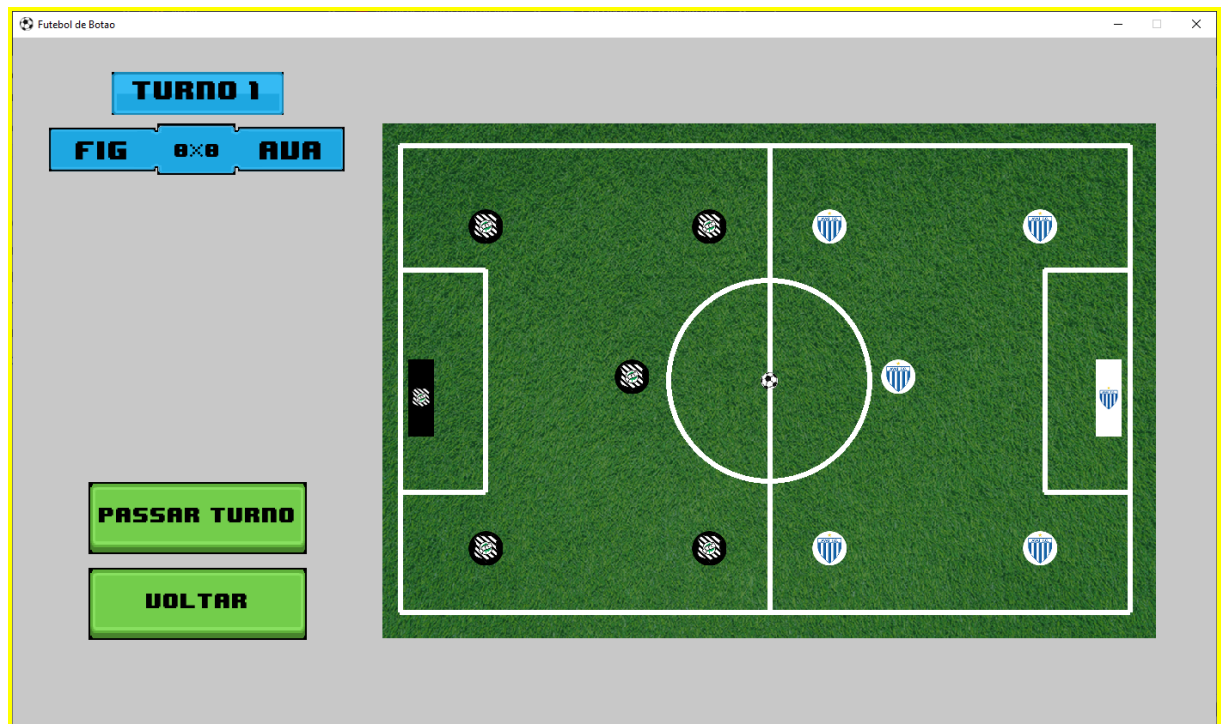
-Tela de Escolha de Times:



-Tela de Escolha de Turnos:



-Tela do Jogo:



-Tela de Créditos:

