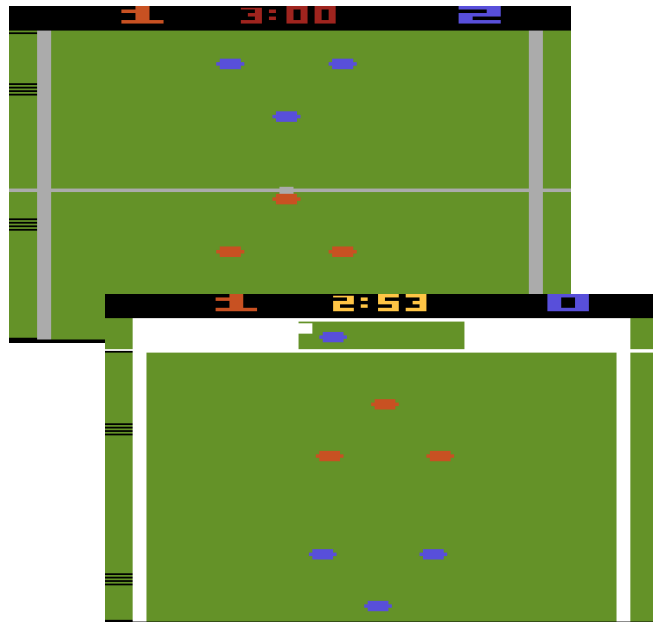


Pelé Soccer



Um pouco de história: o videogame Pelé's Soccer foi publicado em 1980 para rodar no Atari 2600. Foi um dos primeiros, senão o primeiro, videogame simulador de futebol. A visão do jogador era aérea, e cada time tinha um goleiro e mais três jogadores: um atacante e dois defensores. Os 3 jogadores de linha se moviam sempre em formação. Ao fazer contato com a bola, o jogador a conduz por um espaço curto (2x o tamanho do jogador) e depois chuta na direção em que o jogador está se movimentando, dando um chute curto. Caso o jogador pressionasse o botão do joystick, era dado um chute longo, também na direção do movimento. Os goleiros eram automaticamente controlados pela máquina, seguindo a bola com um pequeno atraso. O jogo tinha 54 variações entre single-player e dual-player, onde mudavam a velocidade, a duração e o tamanho dos gols (goleiras em jargão gaúcho).

Encontrem detalhes e experimentem o jogo nos links abaixo e em outros sites na internet:

https://atariage.com/software_page.php?SoftwareID=1195

<https://www.youtube.com/watch?v=LSEASVJIE4>

http://www.retrogames.cz/play_334-Atari2600.php?language=EN

Jogo ASCII C Soccer

É uma implementação em C do Pelé's Soccer usando modo texto, apenas com caracteres da tabela ASCII estendida. O objetivo do jogo é marcar mais gols que o adversário no tempo do jogo. Para conduzir seu time, o jogador humano usará o teclado. O jogo deve exibir o placar e o maior placar já alcançado (recorde).

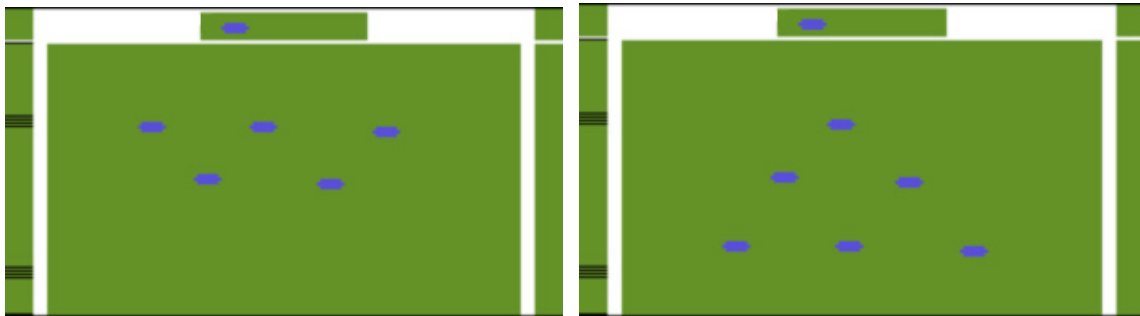
O recorde deve ser salvo em um arquivo binário "record.bin" para ser utilizado em próximas partidas, mesmo que o jogo seja fechado.

O jogo deve permitir pelo menos 4 níveis, apresentando 2 variações de velocidade e 2 de tamanho do gol. Antes de iniciar, o jogo permite o usuário escolher a duração, sendo 1 minuto o valor default. Mais níveis serão considerados um bônus.

O campo de jogo terá tamanho fixo de 64 colunas por 100 linhas, análogo às dimensões mínimas em metros de um campo oficial da FIFA para jogos internacionais. As distâncias serão indicadas em metros neste enunciado; cada metro corresponde a 1 caractere em qualquer direção.

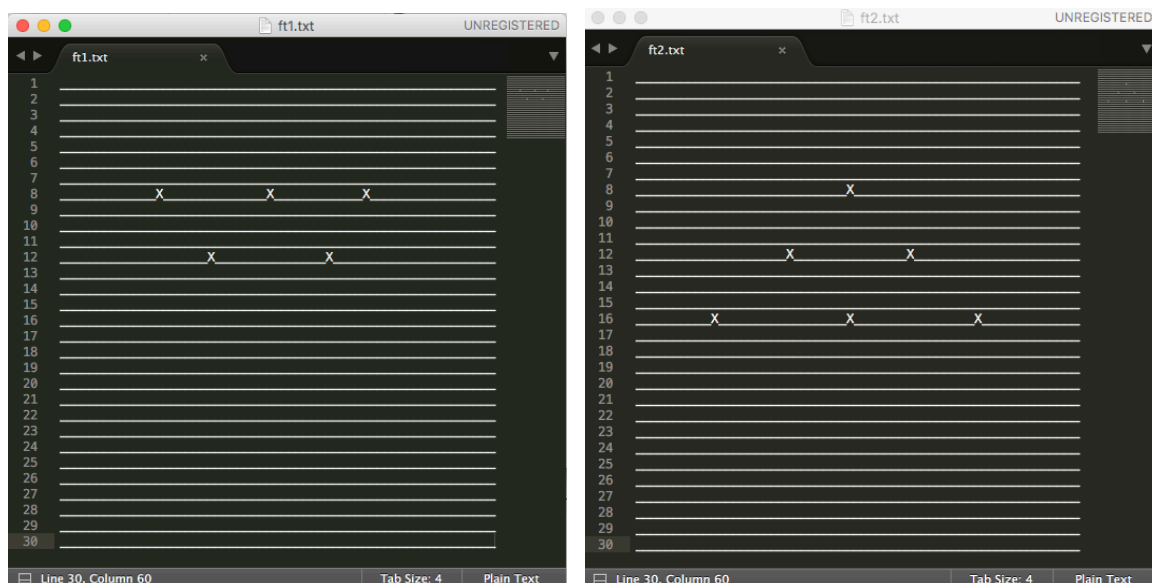
Deve ser exibido o tempo todo o placar do jogo e a contagem do tempo restante.

Formações táticas



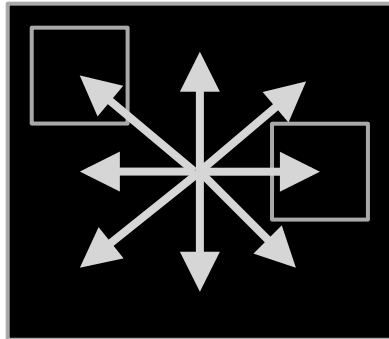
Exemplos de formação tática

Além da formação original, com 2 defensores e 1 atacante, o jogo deve permitir carregar outras 9 formações, que estarão armazenadas em arquivos texto conforme o formato abaixo. Os arquivos de formação táticas serão nomeados ft0.txt, ft1.txt..., ft9.txt, sendo que ft0.txt é a formação original. Cada arquivo contém 30 linhas com 60 caracteres 'X' ou '_'. representando os espaços com jogadores e sem. Caracteres 'X' representam a posição de um jogador, mas ao desenhá-lo na tela outros caracteres e até um conjunto de caracteres podem ser usados.

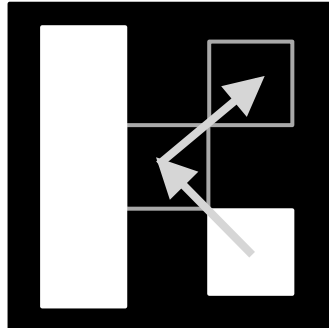


Exemplos de formato do arquivo de formação tática

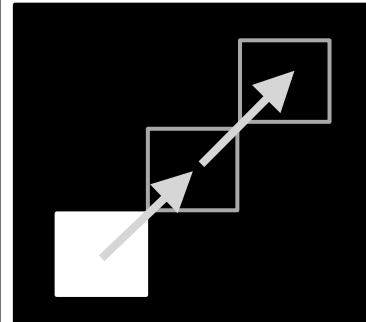
Movimentos



(9 direções possíveis)



(Bola rebatendo)



(Movimento em diagonal)

Movimentos possíveis para a bola e os jogadores

A bola inicia parada no centro do campo e os times devem estar posicionados de modo que nenhum jogador fique a menos de 9 metros da bola. Os times se deslocam em linhas verticais, horizontais ou diagonais. A bola, quando colocada em movimento, também se move desta forma. A bola rebate nas linhas laterais e de fundo do campo, não gerando, portanto a necessidade de reposição por arremesso lateral ou tiro de meta.

Controle Manual (jogador) e automático (bots)

A priori, é mais fácil implementar multi-player, tendo algumas teclas para controlar o time vermelho e outras para o time azul. Isso é o mínimo necessário para esta implementação. No entanto, a título de bônus, o adversário do jogador pode ser controlado pelo computador (bot), tendo-se que, neste caso, implementar uma estratégia de controle automático para o bot. O bot não pode ser invencível, pois isso tiraria a diversão do jogo. Isso pode ser feito de maneira simples, limitando sua velocidade a um patamar inferior à velocidade do jogador.

Requisitos (lista preliminar)

- O trabalho deverá ser feito em duplas. Informar os componentes da dupla até o dia prática 12, 23/06, pela tarefa criada especificamente no moodle.
- Entrega do trabalho: até o dia 19/07 ao meio dia, para ambas as turmas, a dupla deverá submeter via Moodle um arquivo zip cujo nome deve conter o(s) nome(s) do(s) aluno(s). O arquivo zip deve conter:
 - Uma descrição do trabalho realizado contendo a especificação completa das estruturas usadas e uma explicação de como usar o programa
 - Os programas-fonte devidamente organizados e documentados (arquivos.c e arquivos.h, comentários e indentação)
 - Executável do programa

- Trabalho será obrigatoriamente apresentado no horário da aula prática no dia 21/07 para todas as turmas. Ambos membros da dupla deverão saber responder perguntas sobre qualquer trecho do código.
- No dia da apresentação serão realizados diferentes testes de condições de jogo para avaliar o programa.
- Os seguintes itens serão considerados na avaliação do trabalho: estruturação do código, performance de execução, documentação geral do código (comentários, identificação), “jogabilidade” do jogo e atendimento aos requisitos definidos.
- **Importante:** trabalhos copiados não serão considerados. Saibam que há ferramentas que possibilitam a detecção automática de plágio.

Pontos Extras

- Adversário inteligente (prever o caminho da bola, etc.);
- Mais formações táticas e variações de dificuldade;
- Usar os caracteres da tabela ASCII estendida (<http://www.asciitable.com/>);
- Colorir o game;
- Comemorações de gol;
- Novas Ideias.

Dicas:

- Imprimir toda a tela de uma vez é mais rápido que imprimir caracter por caracter.
- Para não imprimir a tela uma após a outra, deve-se alterar a posição do cursor para o início da janela antes de imprimir. A biblioteca windows.h tem uma função para isto.
- Na windows.h também há uma função para identificar o estado das teclas.
- Para limitar o número de atualizações da tela por segundo, use a função Sleep.
- Este trabalho irá exigir bastante tempo e dedicação, então faça um bom planejamento antes de começar programar.