



Engenharia em Desenvolvimento de Jogos Digitais

Técnicas de Desenvolvimento de Jogos

Docente: João Almeida



Trabalho realizado por:



Bruno Ferreira, 13227

João Martins, 13228

Índice

Introdução	3
Características do jogo	3
Tipo de jogo	3
Descrição	3
Fases de construção	3
Público alvo	3
Mecânicas	4
Código e implementação do mesmo	5
Código	5
Funções mais relevantes	6
Arte	8
Desenvolvimento do projeto	9
Ideia inicial e desenvolvimento da mesma	9
Dificuldades em realizar a ideia inicial	9
Ideia secundária	9
Ideia final	9
Distribuição de tarefas	10
Cronograma	10
Vantagens	10
Dificuldades no geral	11
Desenvolvimento futuro	11
Conclusão	11
Bibliografia	11

→Introdução:

Este trabalho foi realizado no âmbito da introdução ao *monogame* e à criação de um jogo através do mesmo.

Ao longo deste trabalho, utilizamos o livro "Learn 2D Game Development with C#" como auxiliar, tutoriais e outros exemplos de jogos, o que nos proporcionou uma melhor orientação.

Características do jogo:

Tipo de jogo:

Consiste num jogo 2D, para 2 jogadores, com mecânicas simples e intuitivas.

Descrição:

O jogo desenvolve-se num espaço desportivo (campo de futebol).

Os jogadores têm como objetivo não permitir que a bola chegue ao chão e alcançar o máximo de pontos possíveis. Os pontos são conseguidos através do contacto de cada personagem com a bola.

Fases de construção:

Inicialmente, a ideia era construir um jogo do género de "*HeadSoccer*", em que um jogador compete contra o outro e ambos têm o objetivo de marcar golos. Porém, tentamos ser mais originais e criar um jogo diferente, o que fez com que surgisse este jogo de cooperação que acaba por ser viciante.

Público alvo:

Qualquer pessoa que seja competitiva e tenha como objetivo divertir-se.

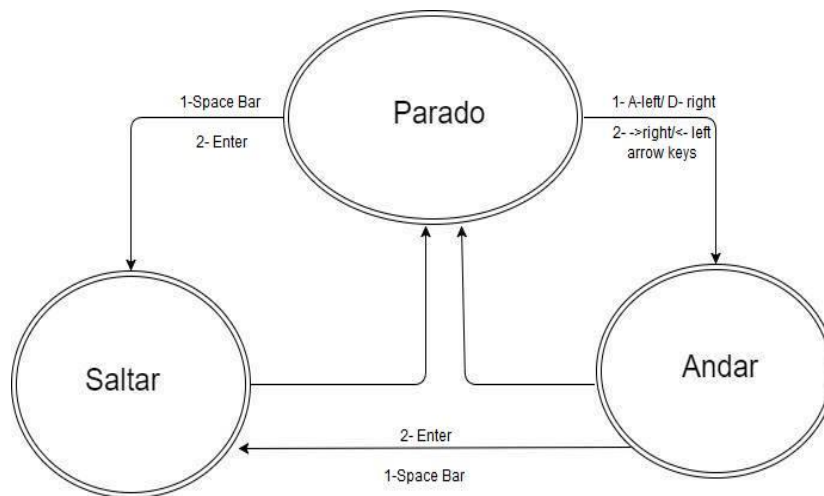
Mecânicas:

Fig 1. Máquina de estados do jogo

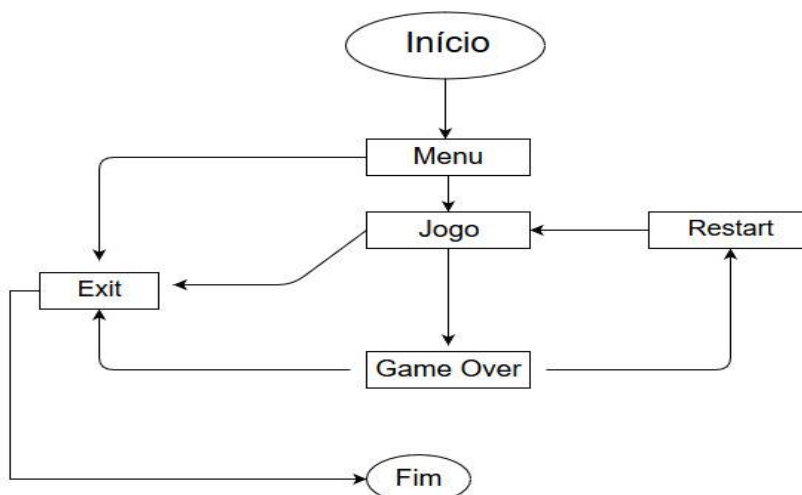


Fig 2. Diagrama de blocos

O jogo é relativamente simples. O jogador precisa apenas de três informações para conseguir jogá-lo.

Seta para o lado direito → Mover-se para a direita.

Seta para o lado esquerdo → Mover-se para a esquerda.

Espaço ("Space") → Saltar.

Ao mesmo tempo, o jogo não funciona por níveis, mas sim por grau de dificuldade, isto porque com o passar do tempo em que os jogadores não permitem que a bola toque no chão, esta começa a ganhar mais velocidade, o que torna a tarefa dos dois bem mais complicada.

Assim, a cooperação entre os jogadores é o que pode permitir um maior número de pontos, o que faz com que o jogo se torne competitivo e provoque um maior entretenimento.

→Código e implementação do mesmo:

Utilizamos 2 tutoriais como auxiliares na realização deste projeto. Um deles ensinou-nos a criar um menu e a mover os diversos objetos. O outro permitiu a implementação das diferentes mecânicas, sendo a mais desconhecida por nós a da "gravidade" (saltos e bola).

Código:

O nosso código é composto por 3 classes:

GUIElement: Tem como função auxiliar na criação dos menus do jogo e na ilustração dos botões.

Personagem: Está responsável pela criação e caracterização das personagens. É utilizada para os dois jogadores, já que estes têm as mesmas características, sendo estas a posição e a velocidade.

Game1: Classe "mãe", é onde se pode manipular todas as funções inseridas nas classes anteriores. Com isto, permite desenhar não só os menus (Inicial e de "gameover", como também todos os elementos que são visíveis durante o jogo (*background*, personagens e bola).

Ao mesmo tempo, permite que seja alterado o tamanho do ecrã, limitar o espaço em que o jogador e as bolas se podem movimentar e tratar os assuntos mais complexos como a colisão entre elementos e a gravidade dos mesmos.

Funções mais relevantes:

```
menus[0].Find(x => x.ElementName == "menu").MoveElement(0, -25);  
menus[0].Find(x => x.ElementName == "play").MoveElement(0, 25);  
menus[1].Find(x => x.ElementName == "gameover").MoveElement(0, -100);  
menus[1].Find(x => x.ElementName == "restart").MoveElement(0, 100);
```

Esta função permite que sejam alteradas as posições dos diferentes sprites.

```
case GameState.inGame:  
    scorefinal = score;  
    jogador1.Update(gameTime);  
    jogador2.Update(gameTime);  
  
    if(MediaPlayer.State != MediaState.Playing)  
        MediaPlayer.Play(musica);
```

Engloba variáveis que têm como função guardar o score para apresentar como score final, fazer *update* de cada jogador ao longo do tempo e reproduzir um clip de som (enquanto se está a jogar).

```
/*-----Controlos do JOGADOR 1:-----*/  
if (Keyboard.GetState().IsKeyDown(Keys.Right)) jogador1.velocity.X = 10f;  
else if (Keyboard.GetState().IsKeyDown(Keys.Left)) jogador1.velocity.X = -10f;  
else jogador1.velocity.X = 0;  
  
//Função que permite ao jogador 1 saltar:  
if (Keyboard.GetState().IsKeyDown(Keys.Enter) && hasjumped == false)  
{  
    jogador1.position.Y -= 10f;  
    jogador1.velocity.Y = -5f;  
    hasjumped = true;  
}
```

Funções que tratam dos controlos dos jogadores (andar e saltar).

```

case GameState.GameOver:
    foreach (GUIElement button in menus[1])
    {
        MediaPlayer.Stop();
        score = 0;
        ballXSpeed = 2; ballYSpeed = 2;
        ball1XSpeed = 2; ball1YSpeed = 2;
        ballPos = new Vector2(GraphicsDevice.Viewport.Width / 2 - ballText.Width / 2, 0);
        ball1Pos = new Vector2(GraphicsDevice.Viewport.Width / 2 - ball1Text.Width / 2, 0);

        jogador1 = new TopBall.personagem(Content.Load<Texture2D>("HeadPlayer2"), new Vector2(700, 500));
        jogador2 = new TopBall.personagem(Content.Load<Texture2D>("HeadPlayer1"), new Vector2(0, 500));
        segundabola = false;
        button.Update();
    }
    break;

```

Importante pelo facto de fazer com que as variáveis, depois de os jogadores perderem, voltem a ter os valores iniciais, o que permite o recomeço do jogo.

```

case GameState.inGame:
    spriteBatch.Draw(backgroundText,new Vector2 (0,0),Color.White);
    spriteBatch.Draw(ballText, ballPos, null);
    jogador1.Draw(spriteBatch);
    jogador2.Draw(spriteBatch);
    spriteBatch.DrawString(font, "Score: " + score, new Vector2(Graphics
    if (segundabola == true)
    {
        spriteBatch.Draw(ball1Text, ball1Pos, null);
    }
    break;

```

Inserida na função *Draw*, nesta função estão inseridos os diversos elementos que queremos que sejam visíveis enquanto se estiver a jogar (*background*, *personagens*, *bolas* e *score*).

```

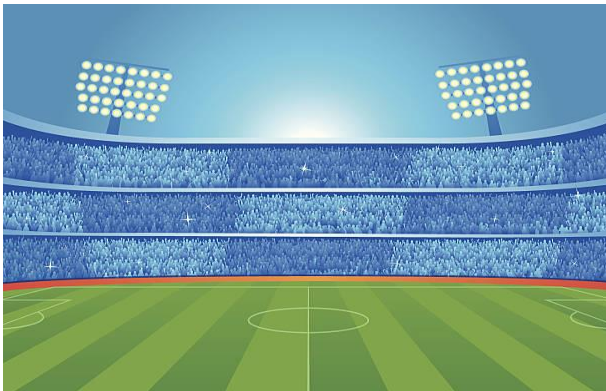
case GameState.GameOver:
    foreach (GUIElement button in menus[1])
    {
        spriteBatch.DrawString(font, "Final Score : "
        button.Draw(spriteBatch);
    }
    break;

    spriteBatch.End();
    base.Draw(gameTime);

```

Caso os jogadores percam, através desta função podemos visualizar um "Final Score" e o botão de *restart*.

→Arte:



Como se pode perceber, em termos de *design*, o jogo é simples, apenas três objetos foram trabalhados por nós, sendo estes os *headplayers* (utilizados por cada um dos jogadores) e o logotipo – desenhados através do *Illustrator CC 2015* e *Logomaker*, respetivamente. Os restantes elementos foram retirados da *internet*.

Para além destes sprites elementos, inserimos uma “música” retirada também da *internet* que consiste no “barulho” do público durante uma partida de futebol.

→Desenvolvimento do projeto:

Ideia inicial e desenvolvimento da mesma:

A nossa ideia inicial era criar um jogo 2D de badminton (1 contra 1). Começamos por pensar em quais seriam as principais mecânicas de jogo - o jogador iria poder saltar, andar e executar os diversos tipos de batimentos técnicos deste desporto.

Dificuldades em realizar a ideia inicial:

Logo após a idealização do jogo, a maior dificuldade surgiu logo no início, quando começamos a refletir sobre como iríamos programar o código para as físicas do volante num *clear*, *smash*, *lob* e *amorti* (diferentes batimentos no *badminton*).

Ideia secundária:

Acabamos por mudar de ideia porque não íamos ser capazes de criar as físicas necessárias para a nossa ideia original sem a ajuda de algo ou de alguém experiente. Assim, pensamos num novo jogo, também de 1 vs 1. No entanto, desta vez com o tema de futebol, onde os 2 jogadores iriam, através de cabeceamentos, tentar marcar golo ao seu adversário (estilo "*HeadSoccer*"). Por fim, quem marcasse mais golos seria o vencedor.

Ideia final:

Concluímos que era uma ideia um pouco "banal" e decidimos ser diferentes (devido também à grande variedade de jogos similares à ideia anterior). Portanto, escolhemos continuar no mesmo género mas desta vez fazer com que fosse um jogo cooperativo, onde os 2 jogadores não podem deixar que a bola toque no chão. Quantos mais passes conseguirem efetuar entre eles, melhor será o *score* final.

→Distribuição de tarefas:

Inicialmente, através da visualização de tutoriais, dividimos as tarefas de modo a um tratar do menu e do *design* e o outro das mecânicas do jogo. Porém, com o passar do tempo, acabamos por trabalhar de forma mais uniforme (realizando ambos a mesma tarefa), isto porque foram aparecendo problemas que precisávamos de pensar juntos para os conseguir resolver.

→Cronograma:

A gestão do tempo foi talvez o nosso maior “inimigo” ao longo do trabalho que realizamos, isto porque não tínhamos a noção do quanto era complicado implementar um código para se realizar um “bom” jogo. Fomos tentando gerir o tempo, de forma a conseguirmos adiantar as coisas o mais rápido possível, mas existem sempre obstáculos pelo caminho que não permitem o sucesso desejado, que foi o que acabou por acontecer.

Isto porque idealizávamos algo melhor trabalhado, com design melhorado, outras opções para o utilizador... No entanto, continua a ser um objetivo, mesmo depois de terminado este tempo de entrega, realizar o resto das tarefas que queríamos ter feito a tempo.

→Vantagens:

Devido aos diversos problemas que surgiram ao longo da realização do projeto, fomos aprendendo o porquê dos mesmos acontecerem, o que nos ofereceu uma maior habilidade a realizar tarefas em *monogame*.

Ao mesmo tempo, permitiu-nos desenvolver as qualidades cooperativas devido à constante entre-ajuda ao longo da realização das tarefas.

→Dificuldades no geral:

De facto, as dificuldades foram algo muito presente ao longo do trabalho, isto porque o nosso conhecimento em C# é relativamente pequeno e, perante erros, temos alguma dificuldade em detetar o que pode estar na causa dos mesmos.

Evidentemente, perdemos imenso tempo no que supostamente deveria ser mais fácil de executar. Tornamos o que era fácil em algo bastante complicado, o que não permitiu que o trabalho se tornasse no projeto que desejávamos.

Em suma, o problema que esteve mais presente ao longo do trabalho foi em relação às colisões, isto porque estas continham alguns "*bugs*" que não soubemos como resolver.

→Desenvolvimento futuro:

Efetivamente, não conseguimos concretizar o que idealizamos inicialmente, tendo ficado por implementar várias coisas, tais como outro tipo de mecânicas e diferentes opções para o utilizador. Mas como já foi referido, não é algo que nos perturbe a nível mental, muito pelo contrário, depois de tantas dificuldades ultrapassadas, acreditamos que podemos chegar bem mais longe e finalizar o jogo que desejamos.

→Conclusão:

Foi um projeto muito importante para a nossa aprendizagem, não só como estudantes mas também como seres humanos, devido ao facto de nos fazer perceber que temos de dar mesmo o devido valor às coisas. Mesmo aquilo que nos parece mais simples, para ser feito precisou de um trabalho que está "escondido" aos nossos olhos.

→Bibliografia:

"Learn 2D Game Development with C";

<https://www.youtube.com/watch?v=ReS1VIUrTnw&t=3135s;>

<https://www.youtube.com/watch?v=qtmN5sgl5Ps> .