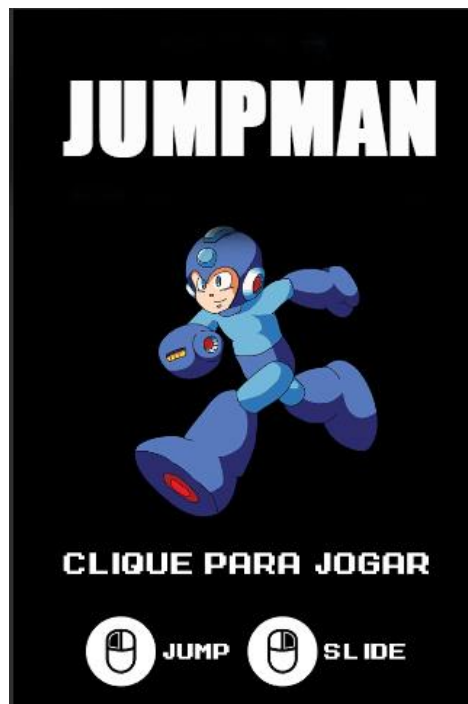


Curso Técnico de Gestão e Programação de Sistemas Informáticos

Relatório PAP



“JUMPMAN”

Pedro Fernandes Nº12

2017/2018

Índice

Introdução.....	3
Objectivos	4
Tecnologias Utilizadas.....	5
Estado da arte.....	7
Fases de desenvolvimento	8
Cronograma.....	9
Conclusão.....	10
Melhoramentos	10
Webgrafia.....	11
Anexos	12

Introdução

Este trabalho surgiu no âmbito da realização de um projeto final de curso, ou seja uma Prova de Aptidão Profissional (PAP), que está contemplada no desenho curricular do curso.

O projeto consiste na realização de um jogo com o nome “JUMPMAN”, será um jogo sobre um desenho animado que tem que passar obstáculos e fazer a maior pontuação possível.

Tive como principal objetivo para a realização deste projeto, a exploração de novos conhecimentos na área da programação e tentar evoluir a minha aprendizagem, tendo sempre como base os conhecimentos adquiridos ao longo dos três anos do curso. Para a concepção do jogo, irei utilizar a plataforma Unity. A escolha desta plataforma surgiu porque no ano passado, quando a turma realizou uma visita de estudo à “Lisboa Games Week”, existiu uma formação desta plataforma em que participei e apercebi-me das potencialidades desta plataforma.

Objectivos

A nível de objectivos, espero conseguir complementar as minhas aprendizagens com a exploração novos conhecimentos como a linguagem C++ ao nível da programação. Visto que a área da informática está sempre a evoluir, tornando-se necessário a formação pessoal em novos conteúdos de forma constante. Escolhi o desenvolvimento de um jogo, porque é um tema bastante atractivo e do qual eu gosto bastante, tornando-se bastante mais motivante a realização da PAP.

- Melhorar as minhas competências ao nível da programação;
- Trabalhar com softwares de edição gráfica;
- Desenvolver um jogo na plataforma Unity;
- Construir o mapa para o jogo;
- Construir os obstáculos.
- Desenvolver um site para a partilha do jogo

Tecnologias Utilizadas

As tecnologias utilizadas para o desenvolvimento deste jogo foram as seguintes:

1.1 Unity

Unity, também conhecido como Unity 3D, é um motor de jogo 3D criado pela Unity Technologies. Unity é similar ao Blender, Virtools ou Torque Game Engine, em relação a sua forma primária de autoria de jogos. O Unity disponibiliza de varias versões, Unity 1,2,3,4,5 e 2017 que é a mais recente e que foi a utilizada para a realização do jogo. As plataformas suportadas incluem BlackBerry 10, o Windows Phone 8, Windows, OS X, Linux (Ubuntu, principalmente), Android, iOS, Unity Web Player (incluindo Facebook, Adobe Flash, PlayStation 3, PlayStation 4, PlayStation Vita, Xbox 360, Xbox One, Wii U, e Wii.

1.2 C++

C++ é uma linguagem de programação multi-paradigma inclui linguagem imperativa, orientada a objetos) e de uso geral. Desde 1990 é uma das linguagens comerciais mais populares, sendo bastante usada também na escola por seu grande desempenho e base de utilizadores. Novas características foram adicionadas com o tempo, como funções virtuais, sobrecarga de operadores, herança múltipla, gabaritos e tratamento de exceções. Após a padronização ISO realizada em 1998 e a posterior revisão realizada em 2003, uma nova versão da especificação da linguagem foi lançada em dezembro de 2014, conhecida informalmente como C++17.

1.3 Microsoft Visual Studio

Microsoft Visual Studio é um ambiente integrado da Microsoft para desenvolvimento de software especialmente dedicado ao .NET Framework e às linguagens do Visual Basic (VB), C, C++, C# (C Sharp) e J# (J Sharp). Também é um produto de desenvolvimento na área web, usando a plataforma do ASP.NET, como websites, aplicativos web, serviços web e aplicativos móveis.

1.4 WordPress

WordPress é um aplicativo de conteúdo para o *web*, escrito em PHP, baseado principalmente para a criação de sites e blogs via *web*. É uma das ferramentas mais famosas na criação de blogs disputando diretamente com o Google. No entanto, o WordPress é utilizado por quem quer um site mais profissional.

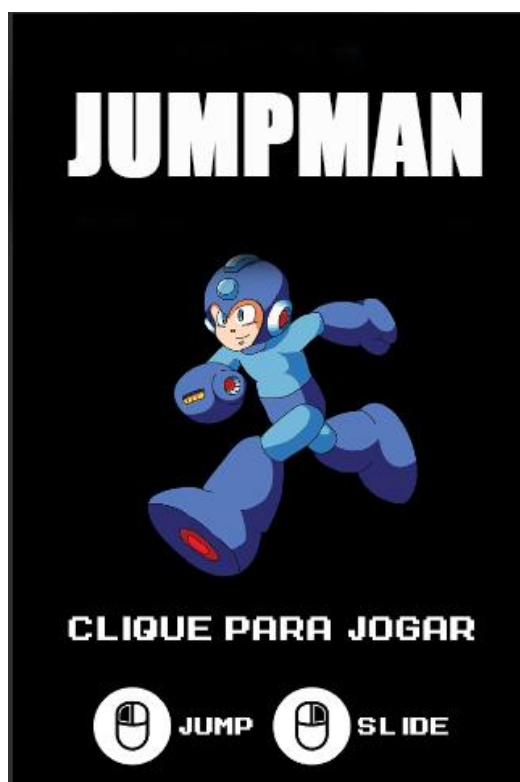
1.5 Photoshop

Adobe Photoshop é um software caracterizado como editor de imagens desenvolvido pela Adobe Systems. É considerado o líder no mercado dos editores de imagem profissionais, usado para edição profissional de imagens digitais e trabalhos de pré-impressão.

Estado da arte

Para a realização deste projeto, vou ter como inspiração jogos para o telemóvel como Temple Run e Flappy Bird, que podemos encontrar em qualquer dispositivo para iOS, Android e até mesmo Microsoft. Também como inspiração tivemos a ideia de usar super-heróis neste caso a personagem incluída e utilizada será o Megamen.

Também realizei um site próprio para o jogo onde terá informação acerca do jogo, no que consiste e objectivos, terá uma página brevemente para a instalação do exe do jogo. Também terá informação acerca dos contactos dos realizadores e uma página de FAQ para qualquer dúvida.



Fases de desenvolvimento

1. Pesquisa sobre a plataforma Unity
2. Aprendizagem da plataforma Unity
3. Pesquisa sobre a linguagem C++
4. Aprendizagem sobre a linguagem C++
5. Pré-produção ou planeamento do jogo
6. Criação do jogo
7. Pós-produção do jogo
8. Realização de um site para apresentar o jogo

Cronograma

Descrição das etapas	Novembro	Dezembro	Janeiro	Fevereiro	Março	Abril
Pesquisa sobre a plataforma Unity						
Aprendizagem da plataforma Unity						
Pré-produção ou planeamento do jogo						
Criação do jogo						
Pós-produção do jogo						
Realização de uma aplicação para apresentar o jogo						

Conclusão

Como conclusão, gostaria de referir que espero que este projeto melhore as minhas competências nas áreas acima referidas e simultaneamente me permita concluir com êxito o meu ciclo de formação.

Este trabalho permitiu-me aprender mais sobre o grafismo usado na plataforma Unity, neste caso em 2D mas também poderá ser utilizado em algum trabalho futuro em 3D.

Também espero que este tipo de trabalho me dê motivação para trabalhos futuros no mercado de trabalho, ao seguir este tipo de profissão ou mesmo qualquer outro tipo de profissão.

Melhoramentos

Tenciono melhorar este jogo adicionando uma parte competitiva criando uma base de dados que guarde as melhores pontuações dos jogadores, tornando o jogo mais apelativo para os jogadores, podendo até haver prémios para os melhores jogadores.

Pretendo melhorar o jogo graficamente para se tornar mais atrativo para os jogadores.

Pretendo tornar o jogo multiplataforma para abranger o maior número de pessoas, com diferentes faixas etárias.

Webgrafia

<https://pt.wikipedia.org/wiki/Unity>

<https://unity3d.com/pt>

<https://pt.wikipedia.org/wiki/C%2B%2B>

<https://pt.wordpress.org/>

<https://pt.wikipedia.org/wiki/WordPress>

https://pt.wikipedia.org/wiki/Microsoft_Visual_Studio

<https://www.visualstudio.com/pt-br/?rr=https%3A%2F%2Fwww.google.pt%2F>

https://pt.wikipedia.org/wiki/Adobe_Photoshop

<https://www.adobe.com/pt/products/photoshop/free-trial-download.html>

Anexos

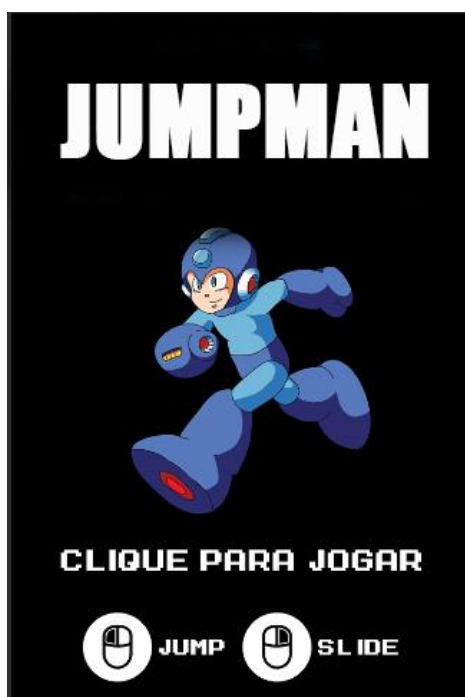


Fig. 1- Este é o ecrã inicial do jogo.

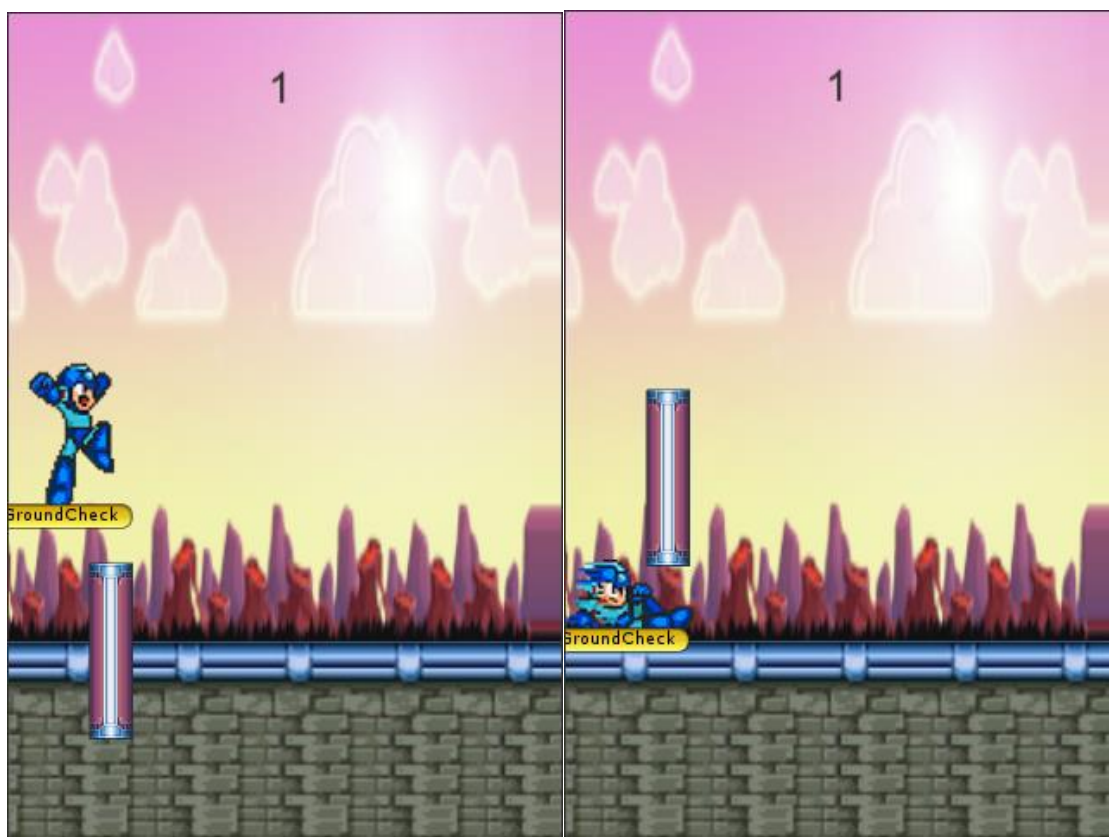


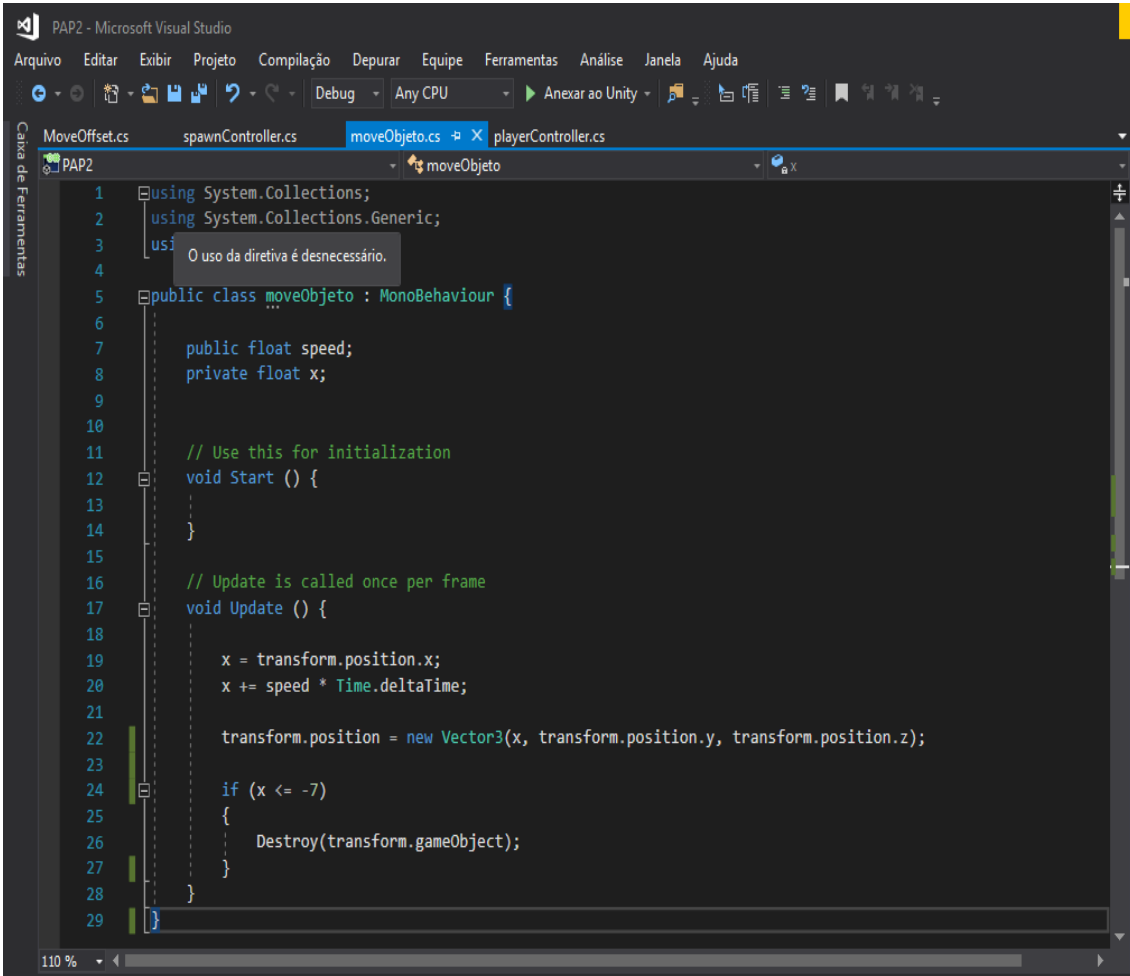
Fig. 2 – Estas 2 imagens mostram o decorrer do jogo onde a personagem se move e também salta e desliza.



Fig. 3 – Esta imagem mostra quando perdemos, mostra também a pontuação e o recorde e também se clicarmos poderemos voltar ao jogo e tentar de novo.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class spawnController : MonoBehaviour {
6
7     public GameObject barreiraPrefab; //objeto a ser spawnando
8     public float ratesSpawn; //intervalo de spawn
9     public float currentTime;
10    private int posicao;
11    private float y;
12    public float posA;
13    public float posB;
14
15    // Use this for initialization
16    void Start () {
17        currentTime = 0;
18    }
19
20
21    // Update is called once per frame
22    void Update () {
23        currentTime += Time.deltaTime;
24        if (currentTime >= ratesSpawn)
25        {
26            currentTime = 0;
27            posicao = Random.Range(1, 100);
28            if (posicao > 50)
29            {
30                y = posA;
31            }
32            else
33            {
34                y = posB;
35            }
36            GameObject tempPrefab = Instantiate(barreiraPrefab) as GameObject;
37            tempPrefab.transform.position = new Vector3(transform.position.x, y);
38        }
39    }
40 }
```

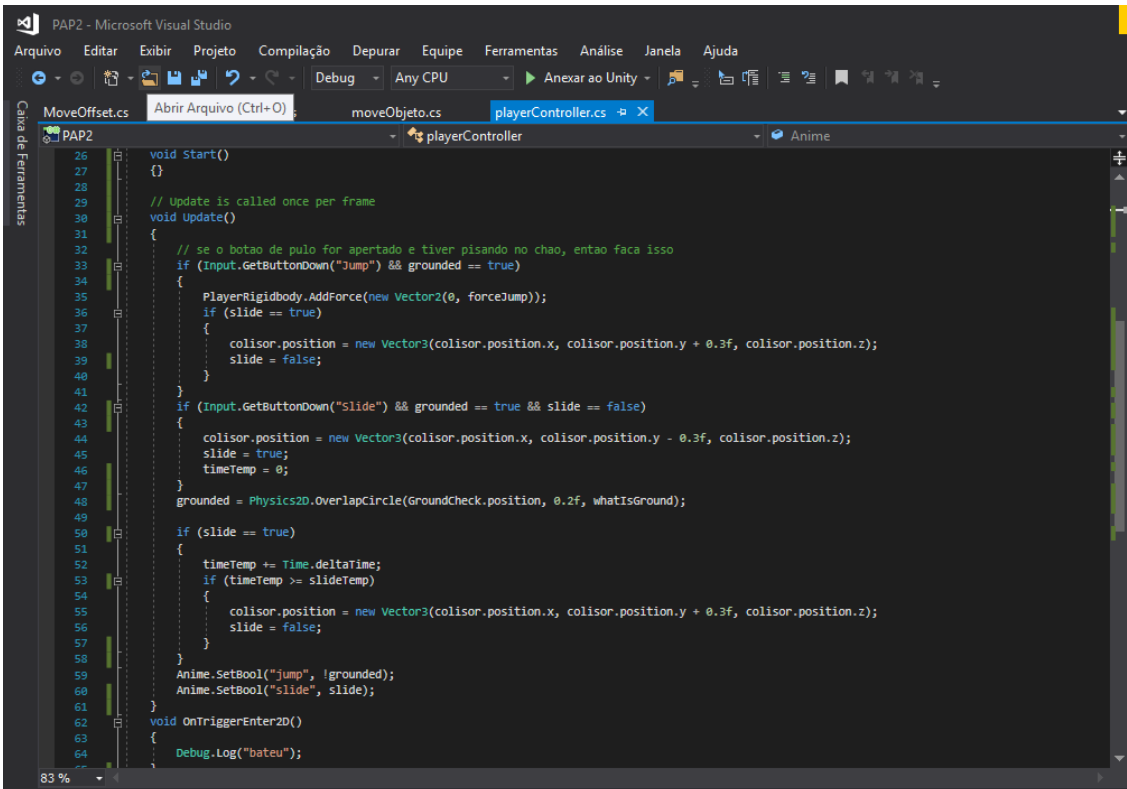
Fig. 4 – Este print mostrar o código utilizado para o spawn automático dos obstáculos.



The screenshot shows the Microsoft Visual Studio IDE with the file 'moveObjeto.cs' open. The code defines a class 'moveObjeto' that inherits from 'MonoBehaviour'. It includes a 'Start' method for initialization and an 'Update' method for frame-by-frame movement. The movement logic involves updating the 'x' coordinate of the object's position based on its speed and the time delta. A boundary check is implemented where the object is destroyed if its 'x' coordinate reaches or goes below -7.

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using
4
5 public class moveObjeto : MonoBehaviour {
6
7     public float speed;
8     private float x;
9
10
11     // Use this for initialization
12     void Start () {
13
14     }
15
16     // Update is called once per frame
17     void Update () {
18
19         x = transform.position.x;
20         x += speed * Time.deltaTime;
21
22         transform.position = new Vector3(x, transform.position.y, transform.position.z);
23
24         if (x <= -7)
25         {
26             Destroy(transform.gameObject);
27         }
28     }
29 }
```

Fig. 5 – Este print mostra o código utilizado para a movimentação dos obstáculos.



```
26 void Start()
27 {
28 }
29 // Update is called once per frame
30 void Update()
31 {
32     // se o botao de pulo for apertado e tiver pisando no chao, entao faca isso
33     if (Input.GetButtonDown("Jump") && grounded == true)
34     {
35         PlayerRigidbody.AddForce(new Vector2(0, forceJump));
36         if (slide == true)
37         {
38             colisor.position = new Vector3(colisor.position.x, colisor.position.y + 0.3f, colisor.position.z);
39             slide = false;
40         }
41     }
42     if (Input.GetButtonDown("Slide") && grounded == true && slide == false)
43     {
44         colisor.position = new Vector3(colisor.position.x, colisor.position.y - 0.3f, colisor.position.z);
45         slide = true;
46         timeTemp = 0;
47     }
48     grounded = Physics2D.OverlapCircle(GroundCheck.position, 0.2f, whatIsGround);
49
50     if (slide == true)
51     {
52         timeTemp += Time.deltaTime;
53         if (timeTemp >= slideTemp)
54         {
55             colisor.position = new Vector3(colisor.position.x, colisor.position.y + 0.3f, colisor.position.z);
56             slide = false;
57         }
58     }
59     Anime.SetBool("jump", !grounded);
60     Anime.SetBool("slide", slide);
61 }
62 void OnTriggerEnter2D()
63 {
64     Debug.Log("bateu");
65 }
```

Fig. 6 – Este print mostra o código utilizado para que o personagem reaja aos comandos do utilizador neste caso para saltar ou deslizar.

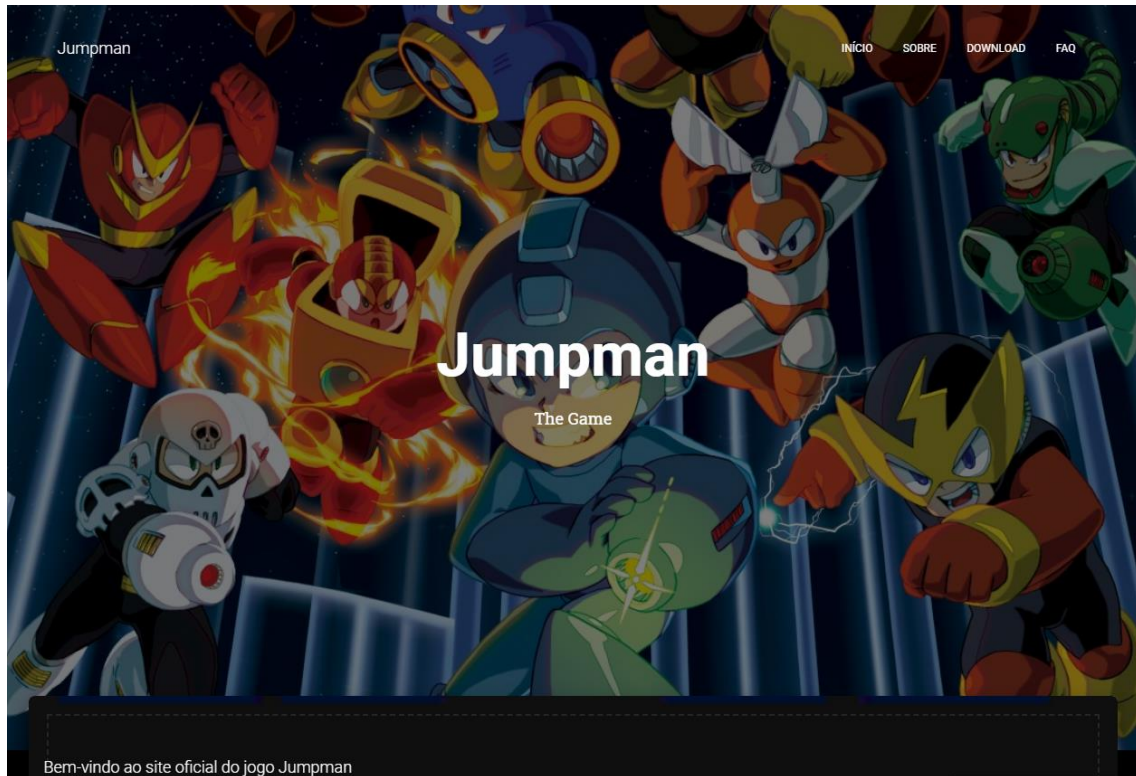


Fig. 7 – Este print mostra a pagina inicial do site da partilha do jogo em WordPress.

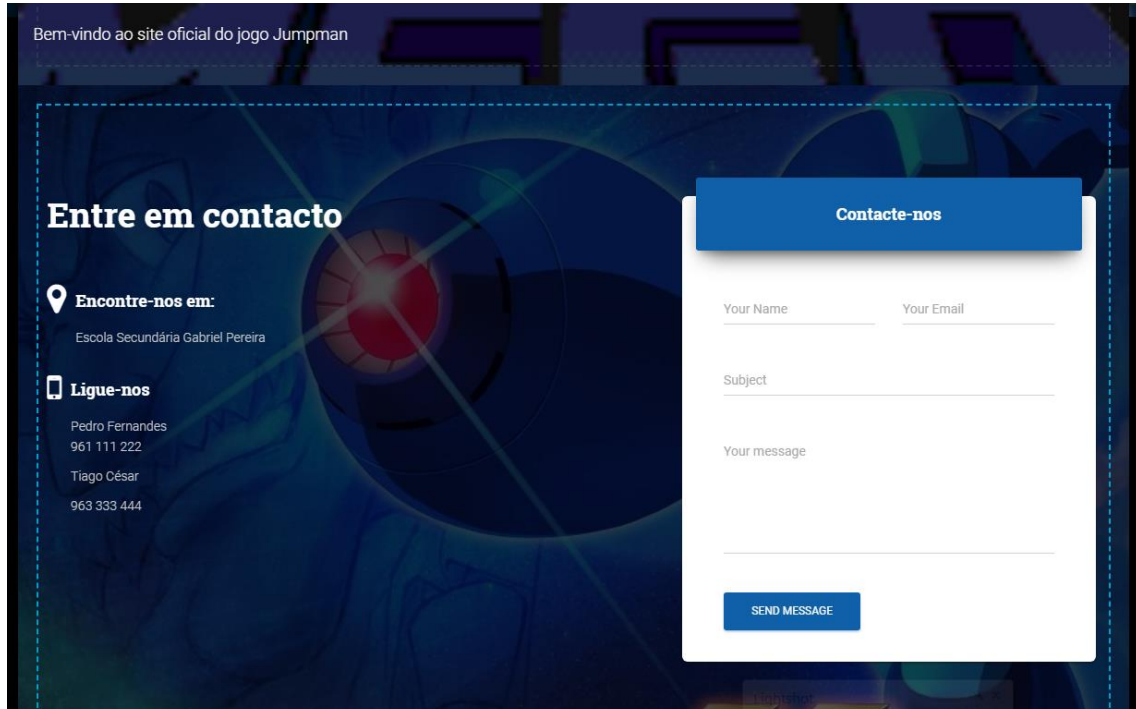


Fig. 8 – Este print mostra a parte dos contactos onde também poderá deixar um mail com qualquer dúvida, também situado na página inicial.

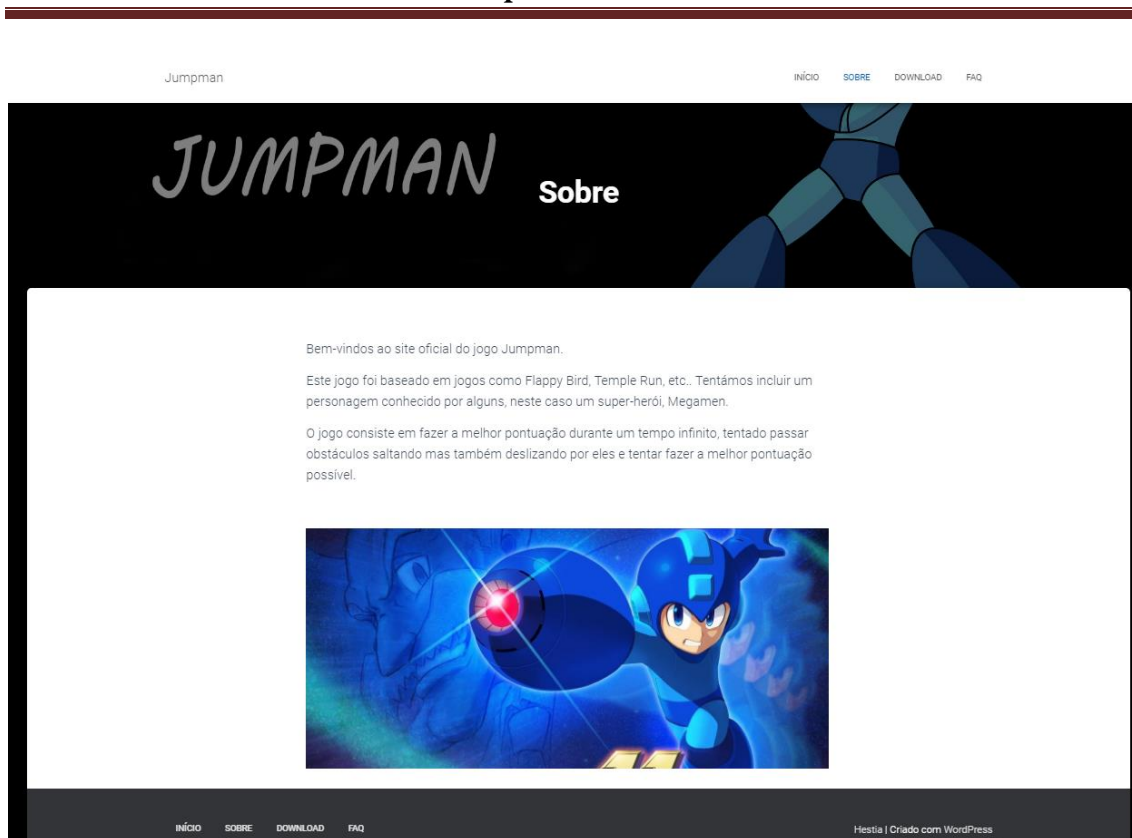


Fig. 9 – Este print mostra a página Sobre, onde podemos ter informação sobre o jogo.

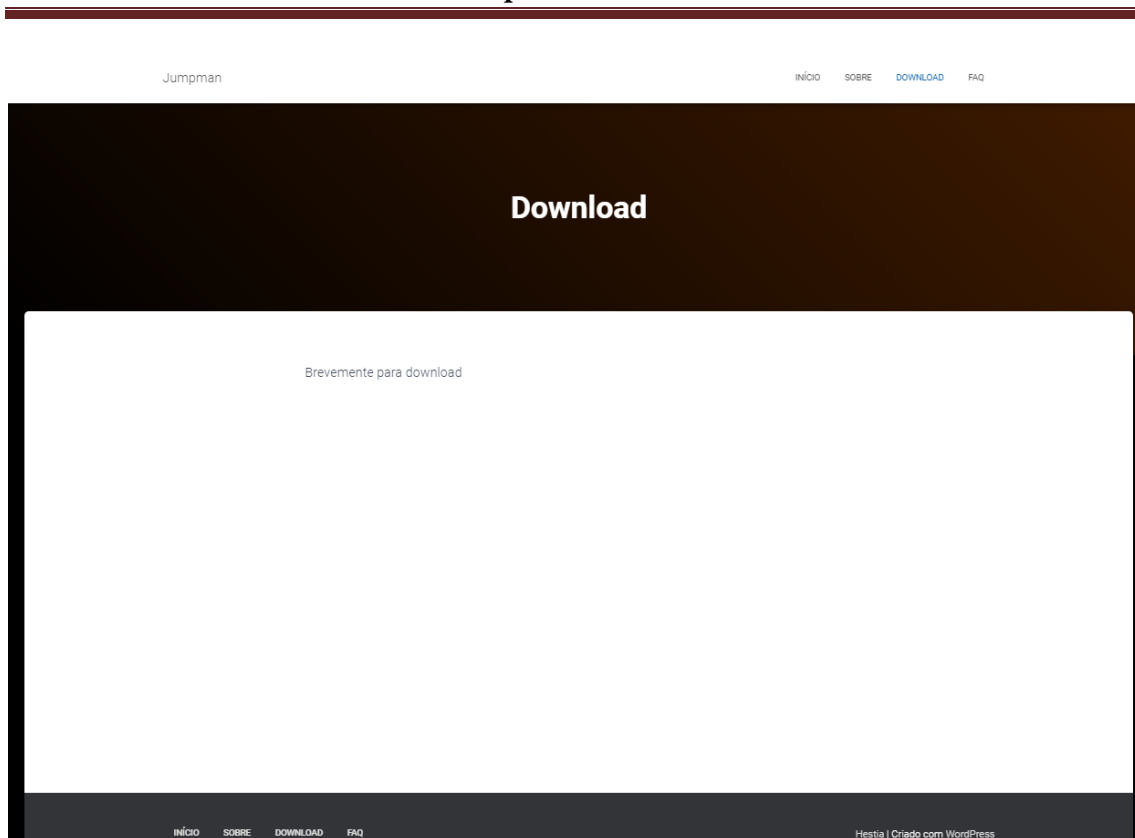


Fig. 10 – Este print mostra a página Download onde brevemente terá a possibilidade de efectuar o download do jogo.

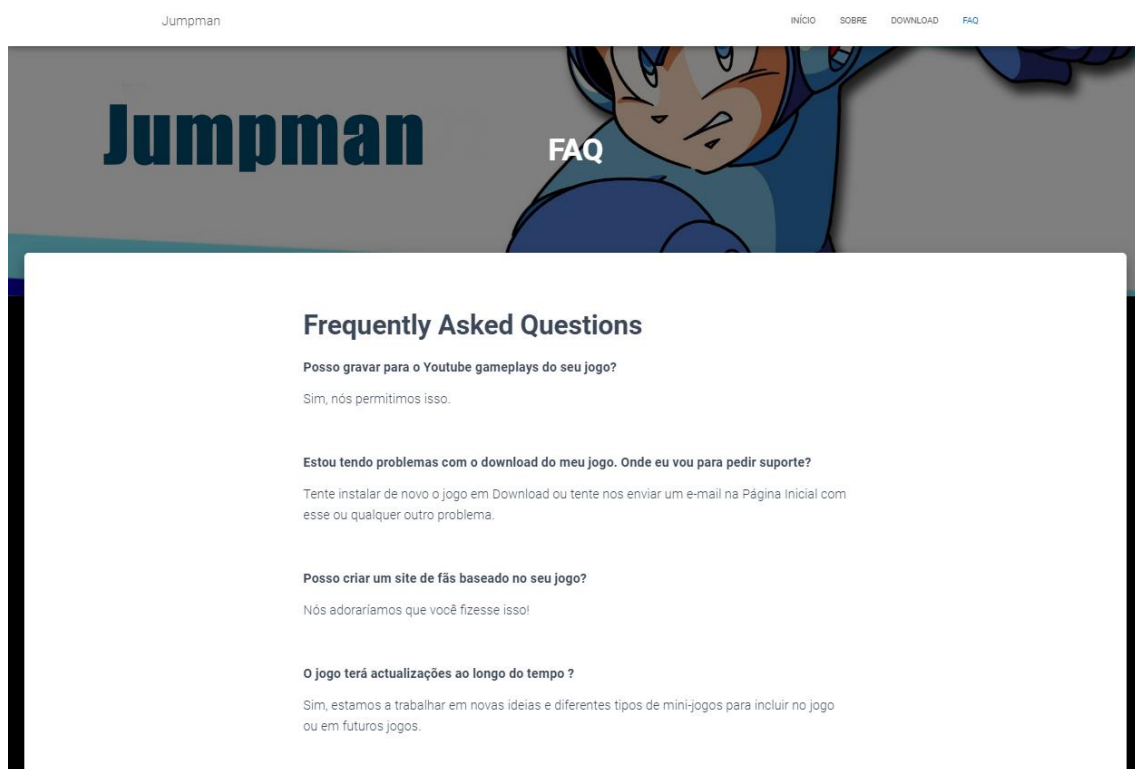


Fig. 10 – Este print mostra a página FAQ, será uma pagina onde terá perguntas frequentes sobre duvidas regulares.