

# Relatório do Trabalho Final

1)

Foi desenvolvido através do semestre diagramas para ajudar na implementação do trabalho final. Esses, além de complementar as informações necessárias, tornariam o projeto algo mais real com o que acontece nas grandes empresas.

A ideia inicial do trabalho era construir um jogo para Android no qual teria como objetivo entreter os usuários em momentos de filas, tédio entre outros. Com isso a ideia de um jogo infinito foi a que melhor se encaixou para tal. Para mais distração, foi usado de figuras cartunizadas com tema voltado ao humor. E isso se juntou ao fácil entendimento de como o jogo funciona para criar essa imersão.

A seguir, imagens dos menus e personagens:



2) Levantamento de requisitos:

José Eduardo Brandão

## Sistema de jogo 2D para Android

Olá, gostaria de um jogo arcade desenvolvido para Android, esse seria sem fim, possibilitando o usuário terminar o jogo apenas quando morrer. A temática desse jogo seria um cachorro caindo do céu, e nesse percurso ele encontraria obstáculos (brócolis) que o matariam e alguns bônus (hamburguers) que o deixariam feliz.

Requisitos funcionais:

- ☐ Jogo da categoria arcade;
- ☐ Jogo "endless-runner" (sem fim);
- ☐ O cachorro deve desviar de obstáculos (brócolis) que são gerados aleatoriamente;
- ☐ O cachorro pode passar por cima de bônus (hamburguers), esses aumentam a pontuação em 10;
- ☐ A pontuação é incrementada a cada segundo;
- ☐ Durante o jogo, trilha sonora característica com a temática;
- ☐ Menu com botões de Jogar e Fechar;
- ☐ Menu dentro do jogo com botões de Pause/Play ;
- ☐ Quando morrer, opções de botões: Play again, Menu, Quit;

Requisitos não funcionais: ☐ Desempenho;

☐ Facilidade de uso, menus com poucos botões e comandos e interface simples e precisa;

Restrições: ☐

Jogo para Android;

☐ Jogo em 2D;

☐ Motor gráfico utilizado Unity;

Diagrama de casos de uso:

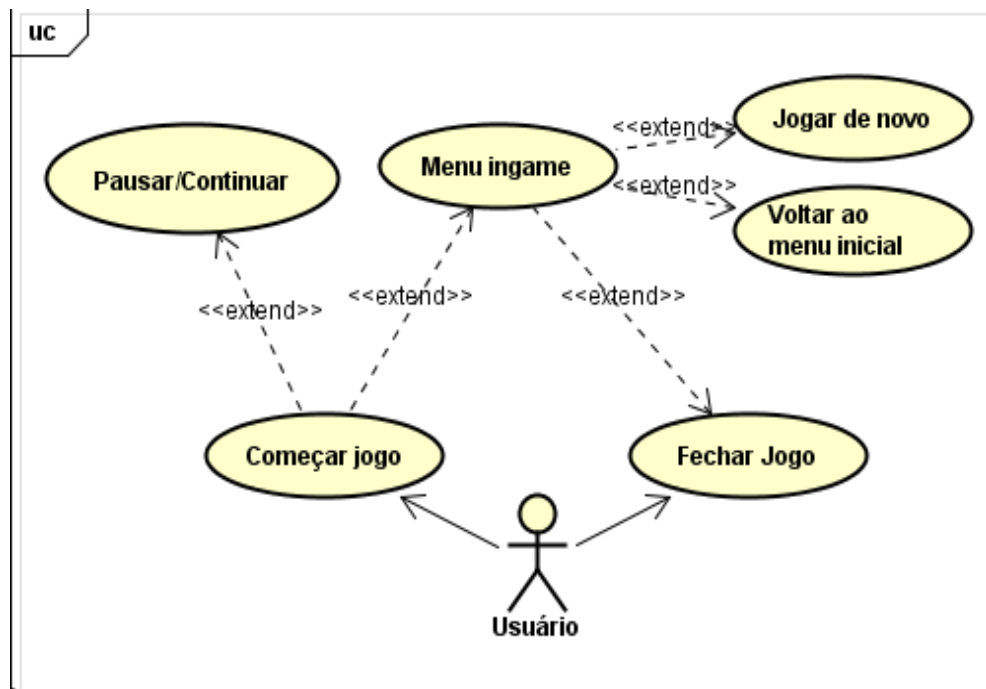
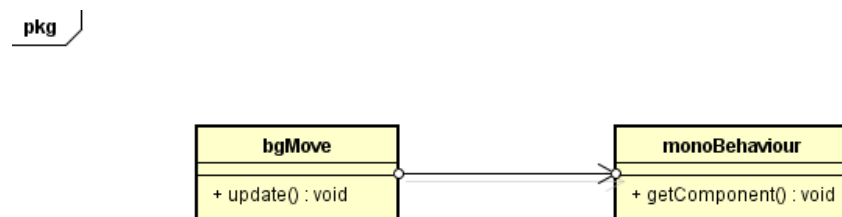
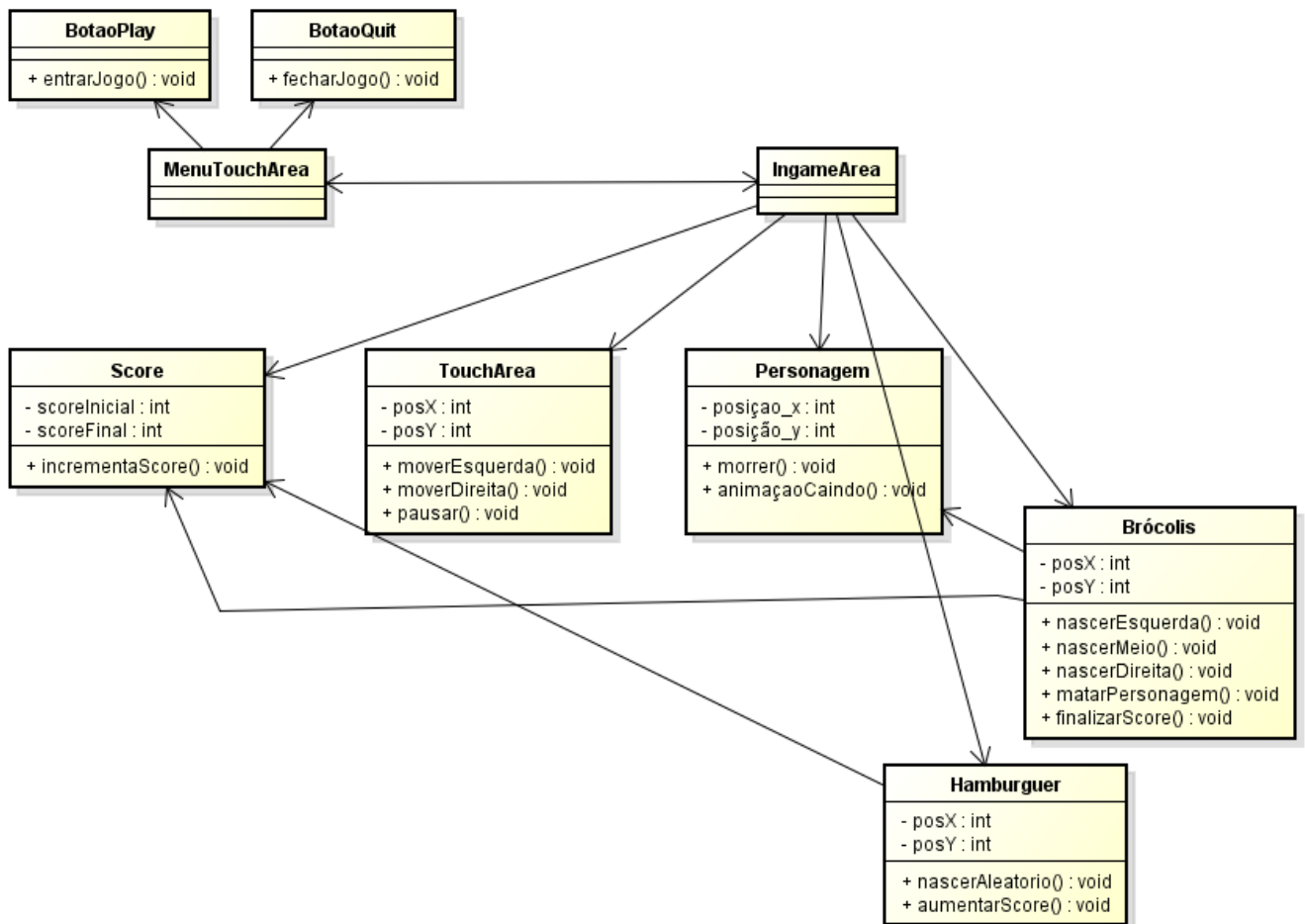
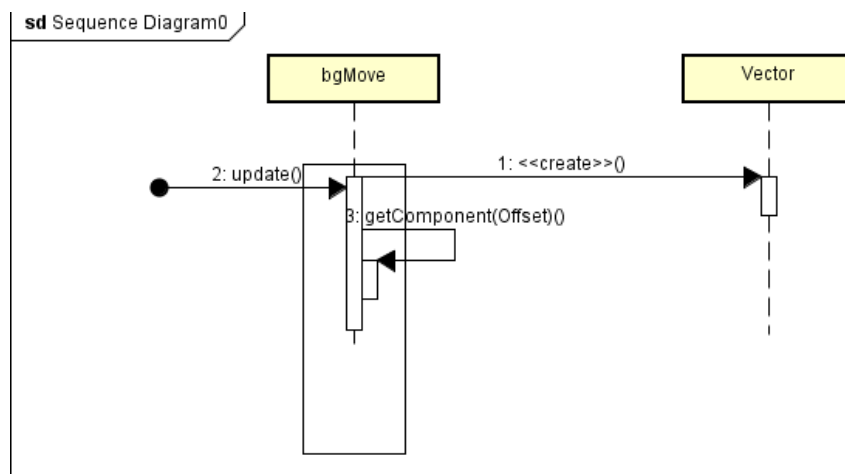


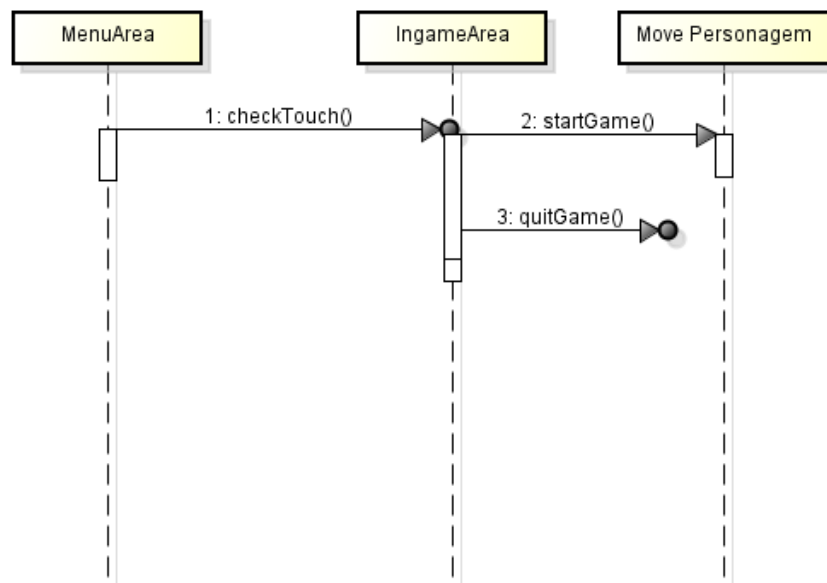
Diagrama de classes:





## Diagrama de sequência:





### 3) Códigos-fonte/Scripts

Movimento do plano de fundo:

```

1 using UnityEngine;
2 using System.Collections;
3
4 public class bgMove : MonoBehaviour {
5
6     public float speed;
7     Vector2 offset;
8
9
10    // Use this for initialization
11    void Start () {
12
13    }
14
15    // Update is called once per frame
16    void Update () {
17
18        offset = new Vector2( 0 , -Time.time * speed);
19        GetComponent<Renderer>().material.mainTextureOffset = offset;
20    }
21 }
22 }
23

```

Destruir dos itens:

```

1 using UnityEngine;
2 using System.Collections;
3
4 public class Destroyer : MonoBehaviour {
5
6     // Use this for initialization
7     void Start () {
8
9     }
10
11     // Update is called once per frame
12     void Update () {
13
14     }
15
16     void OnCollisionEnter2D (Collision2D col)
17     {
18         if (col.gameObject.tag == "BadItem")
19         {
20             Destroy(col.gameObject);
21         }
22         if (col.gameObject.tag == "GoodItem")
23         {
24             Destroy(col.gameObject);
25         }
26     }
27 }
28

```

Destruidor do personagem:

```
1 using UnityEngine;
2 using System.Collections;
3
4
5 public class burgSumir : MonoBehaviour {
6     public UImanager UIburg;
7
8
9
10    // Use this for initialization
11    void Start () {
12
13    }
14
15    // Update is called once per frame
16    void Update () {
17
18    }
19
20    void OnCollisionEnter2D (Collision2D col)
21    {
22        if (col.gameObject.tag == "PrincipChar")
23        {
24            Destroy(gameObject);
25
26        }
27    }
28
29 }
30 }
31
```

Controlador do personagem:

```

1 using UnityEngine;
2 using System.Collections;
3
4 public class dogControl : MonoBehaviour {
5     public float dogSpeed;
6     Vector3 position;
7     public UIManager ui;
8     public AudioManager am;
9     public MenuManager mm;
10    bool currentPlatformAndroid = false;
11    public GameObject principChar;
12
13    bool _isMoveLeft = false;
14    bool _isMoveRight = false;
15
16    // Use this for initialization
17    void Awake(){
18
19        #if UNITY_ANDROID
20            currentPlatformAndroid = true;
21        #else
22            currentPlatformAndroid = false;
23        #endif
24
25    }
26
27
28
29
30    void Start () {
31
32
33
34        position = transform.position;
35        am.windSound.Play ();
36
37        if (currentPlatformAndroid == true) {
38            Debug.Log ("Android");
39
40        } else {
41            Debug.Log ("Windows");
42        }
43
44    }
45
46    // Update is called once per frame
47    void Update () {
48        //TouchMove ();
49
50        position.x = Mathf.Clamp(position.x, -2.6f, 3.06f);
51
52
53        if (_isMoveLeft == true) {
54            transform.Translate (-dogSpeed * Time.deltaTime, 0, 0);
55
56        } else {
57            transform.Translate (0, 0, 0);
58        }
59
60        if (_isMoveRight == true) {
61            transform.Translate (dogSpeed * Time.deltaTime, 0, 0);
62        } else {
63            transform.Translate (0, 0, 0);
64        }
65
66        /*position.x += Input.GetAxis("Horizontal") * dogSpeed * Time.deltaTime;
67        position.x = Mathf.Clamp(position.x, -2.6f, 3.06f);
68        transform.position = position;*/
69
70    }
71

```



```

72 void OnCollisionEnter2D (Collision2D col)
73 {
74     if (col.gameObject.tag == "BadItem")
75     {
76         Destroy(principChar);
77         ui.gameOverAct();
78         am.windSound.Stop ();
79     }
80
81     if (col.gameObject.tag == "GoodItem")
82     {
83         Destroy(col.gameObject);
84         ui.scoreCheck();
85     }
86
87
88 }
89
90
91 public void MoveLeft() {
92     _isMoveLeft = true;
93 }
94
95 public void MoveRight() {
96     _isMoveRight = true;
97 }
98
99 public void SetVelocityZero() {
.00     _isMoveLeft = false;
.01     _isMoveRight = false;
.02 }
.03
.04
.05
.06 }
.07

```

## Movimentação do brócolis:

```

1 using UnityEngine;
2 using System.Collections;
3
4 public class EnemMov : MonoBehaviour {
5     public float speed = 5f;
6
7     // Use this for initialization
8     void Start () {
9
10    }
11
12    // Update is called once per frame
13    void Update () {
14        transform.Translate(new Vector3(0, 1, 0) * speed * Time.deltaTime);
15    }
16 }
17

```

## Criador randômico dos brócolis:

```

1 using UnityEngine;
2 using System.Collections;
3
4 public class legumeSpawnerLeft : MonoBehaviour {
5
6     public GameObject legume;
7     public float delayTimer = 0.5f;
8     float timer;
9
10    // Use this for initialization
11
12
13    void Start () {
14
15
16    }
17
18    // Update is called once per frame
19    void Update () {
20
21        timer -= Time.deltaTime;
22
23        if (timer <= 0)
24        {
25            Vector3 legPos = new Vector3(Random.Range(-2.6f, -0.72f), Random.Range(-6f, -9f), transform.position.z);
26
27            Instantiate(legume, legPos, transform.rotation);
28            timer = delayTimer;
29        }
30    }
31 }
32

```

```

1 using UnityEngine;
2 using System.Collections;
3
4 public class legumeSpawner : MonoBehaviour {
5
6     public GameObject legume;
7     public float delayTimer = 0.5f;
8     float timer;
9
10    // Use this for initialization
11
12
13    void Start () {
14
15
16    }
17
18    // Update is called once per frame
19    void Update () {
20
21        timer -= Time.deltaTime;
22
23        if (timer <= 0)
24        {
25            Vector3 legPos = new Vector3(Random.Range(-2.6f, 1.22f), Random.Range(-6f, -9f), transform.position.z);
26
27            Instantiate(legume, legPos, transform.rotation);
28            timer = delayTimer;
29        }
30    }
31 }
32

```

```

1 using UnityEngine;
2 using System.Collections;
3
4 public class legumeSpawnerMid : MonoBehaviour {
5
6     public GameObject legume;
7     public float delayTimer = 0.5f;
8     float timer;
9
10    // Use this for initialization
11
12
13    void Start()
14    {
15
16
17    }
18
19    // Update is called once per frame
20    void Update()
21    {
22
23        timer -= Time.deltaTime;
24
25        if (timer <= 0)
26        {
27            Vector3 legPos = new Vector3(Random.Range(-1.0f, 1.0f), Random.Range(-6f, -9f), transform.position.z);
28
29            Instantiate(legume, legPos, transform.rotation);
30            timer = delayTimer;
31        }
32    }
33 }

```

```

1 using UnityEngine;
2 using System.Collections;
3
4 public class legumeSpawnerRight : MonoBehaviour {
5
6     public GameObject legume;
7     public float delayTimer = 0.5f;
8     float timer;
9
10    // Use this for initialization
11
12
13    void Start()
14    {
15    }
16
17    // Update is called once per frame
18    void Update()
19    {
20
21        timer -= Time.deltaTime;
22
23        if (timer <= 0)
24        {
25            Vector3 legPos = new Vector3(Random.Range(-1.22f, 3.06f), Random.Range(-6f, -9f), transform.position.z);
26
27            Instantiate(legume, legPos, transform.rotation);
28            timer = delayTimer;
29        }
30    }
31 }
32

```

Controlador do menu:

```

1 using UnityEngine;
2 using System.Collections;
3 using UnityEngine.UI;
4 using UnityEngine.SceneManagement;
5
6 public class MenuManager : MonoBehaviour {
7
8     public AudioManager amg;
9
10    // Use this for initialization
11    void Start () {
12
13    }
14
15    // Update is called once per frame
16    void Update () {
17
18    }
19
20    public void Plaay(){
21        SceneManager.LoadScene ("level1");
22    }
23
24    public void Quuit(){
25        Application.Quit();
26    }
27
28
29 }
30

```

Controlador da interface do usuário:

```
1 using UnityEngine;
2 using System.Collections;
3 using UnityEngine.UI;
4 using UnityEngine.SceneManagement;
5 public class UImanager : MonoBehaviour {
6
7     public Button[] buttons;
8     public Text scoreText;
9     public int score;
10    bool gameover;
11    bool scoreUp;
12
13    // Use this for initialization
14    void Start () {
15        scoreUp = false;
16        score = 0;
17
18        gameover = false;
19        InvokeRepeating("scoreUpdate", 0.6f, 1f);
20        InvokeRepeating("upScore", 0.6f, 0.1f);
21    }
22
23    // Update is called once per frame
24    void Update () {
25        scoreText.text = " " + score;
26    }
27
28
29    void scoreUpdate()
30    {
31        if (!gameover)
32        {
33            score += 1;
34        }
35    }
36
```

```

37     public void gameOverAct()
38     {
39
40         gameover = true;
41         foreach (Button button in buttons) {
42             button.gameObject.SetActive (true);
43         }
44     }
45     public void scoreCheck()
46     {
47         if (!scoreUp)
48         {
49             score = score + 10;
50             scoreUp = false;
51         }
52     }
53
54
55     public void Play(){
56         SceneManager.LoadScene ("level1");
57     }
58
59     public void Quit(){
60         Application.Quit();
61     }
62
63     public void Menu(){
64         SceneManager.LoadScene ("Menu");
65     }
66
67
68     public void Pause()
69     {
70         if (Time.timeScale == 1)
71         {
72             Time.timeScale = 0;
73         }
74         else if (Time.timeScale == 0)
75         {
76             Time.timeScale = 1;
77         }
78     }
79
80
81 }
82

```

Os scripts foram construídos de acordo com a necessidade do Unity, esses usam de bibliotecas do motor gráfico e possuem ajuda das documentações do API dele próprio.

#### 4) Considerações finais:

O uso dos diagramas facilitou para compreender e executar todos os passos desse projeto, conseguiu-se entender todos os problemas que foram criados de forma inteligente e foi possível solucioná-los rapidamente. Além disso trouxe uma seriedade a mais para o trabalho final, tornando esse algo mais profissional e com aparência da realidade do mercado de trabalho.