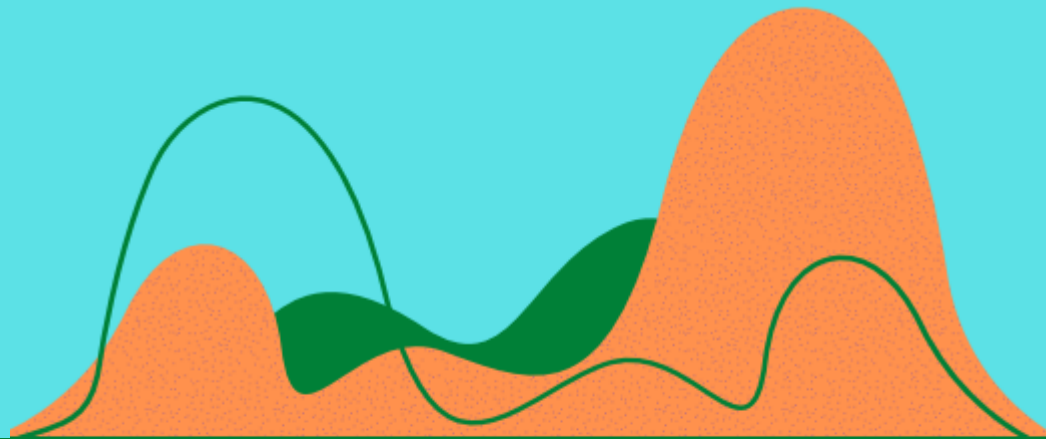


CoVictory-19



Letícia Namie Akamine

RA: 11201920105

Jogo Completo

+ O que foi implementado em relação a última entrega?

+ Como foi implementado?



O que foi implementado

- Correções dos problemas da última entrega
- Adição de complementos
- Mapa de fases
- Fase 2
- Fase 3

Correções dos problemas da última entrega

- Exibição no modo executável era diferente da janela game do Unity em todas as cenas
 - O problema foi resolvido apenas alterando os Canvas das cenas, deixando-os adaptável com o tamanho da tela



Correções dos problemas da última entrega

- A transição entre as telas de instrução e a fase 1 estavam separadas em dois scripts diferentes
 - O processo foi otimizado em um único script utilizado nas cenas: *Instructions*, *Instructions2*, *Level 1*, *Level 2* e *Level 3*
 - As telas de instrução já haviam sido criadas. Já as cenas de Level 1, 2 e 3 são referentes aos mapas e não às fases em si

- A transição entre as telas de instrução e game estavam separadas em dois scripts diferentes

- O processo foi otimizado em um único script utilizado nas cenas:

Instructions, Instructions2, Level 1, Level 2 e Level 3

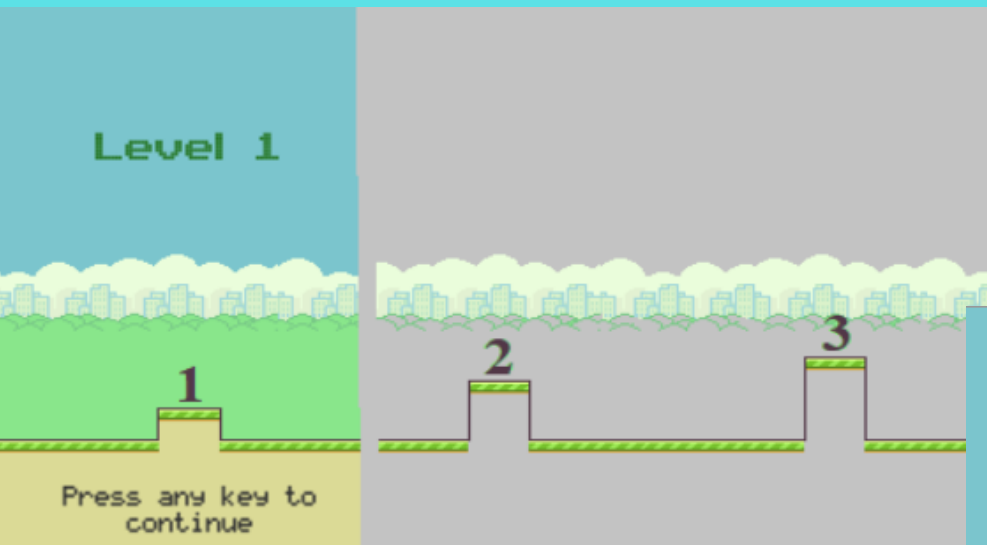
```
SceneManager.cs  x  MoveItem.cs  x  PlayerController.cs  x  S
public int scene;

void Start()
{
    Time.timeScale = 1;
}

void Update()
{
    if (Input.anyKey && scene == 0)
    {
        SceneManager.LoadScene("Instructions2");
    }
    else if (Input.anyKey && scene == 1)
    {
        SceneManager.LoadScene("Level 1");
    }
    else if (Input.anyKey && scene == 2)
    {
        SceneManager.LoadScene("Game");
    }
    else if (Input.anyKey && scene == 3)
    {
        SceneManager.LoadScene("Game 2");
    }
    else if (Input.anyKey && scene == 4)
    {
        SceneManager.LoadScene("Game 3");
    }
}
```

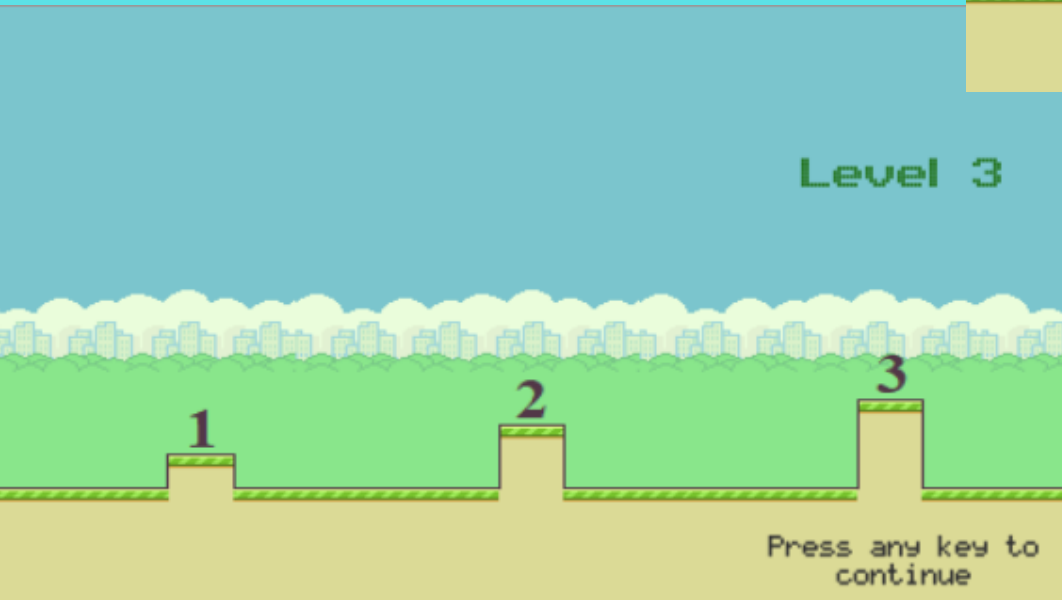
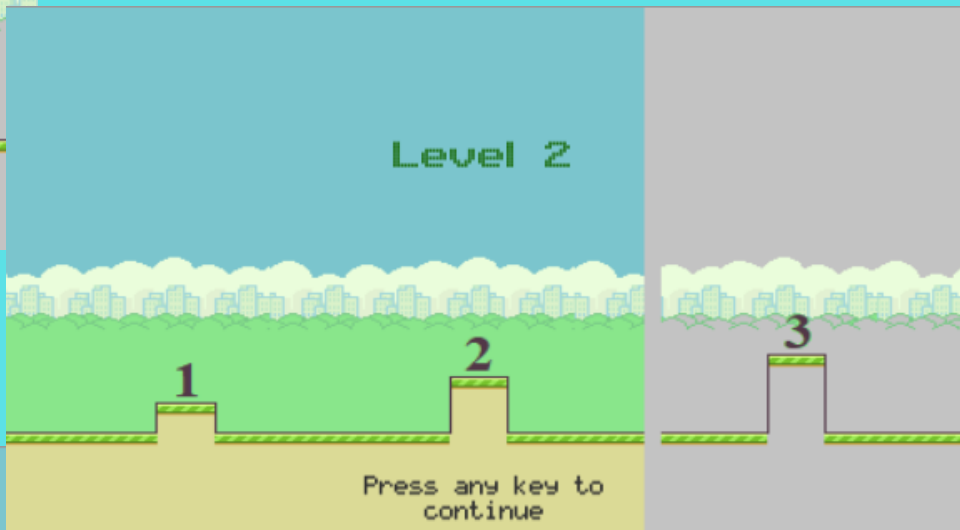
Cada cena possui um valor inteiro de *scene*

A mudança ocorre dependendo da cena em que estamos



Cenas *Level 1*, *Level 2* e *Level 3*

Tornam-se coloridas de acordo com a fase



Todas essas cenas possuem o script *Transition*

Correções dos problemas da última entrega

- Outro problema corrigido foi o de falta de precisão na colisão dos objetos
 - Os objetos passaram a fazer parte da cena: todos foram colocados como filhos do objeto vazio Gameltems.

Cada objeto possui o script *Moveltem*, que movimenta-os de modo análogo ao chão.

O objeto Gameltems possui o script *game*, que ativa aleatoriamente os objetos filhos

game.cs

```
void Update()
{
    time = time + Time.deltaTime;
    vaccineTime += Time.deltaTime;

    if(gameLevel == 1 && vaccineTime >= 30)
    {
        Vaccine.gameObject.SetActive(true);

        vaccineTime = 0;

        time = -1;
    }
    else if(gameLevel == 2 && vaccineTime >= 60)
    {
        Vaccine.gameObject.SetActive(true);

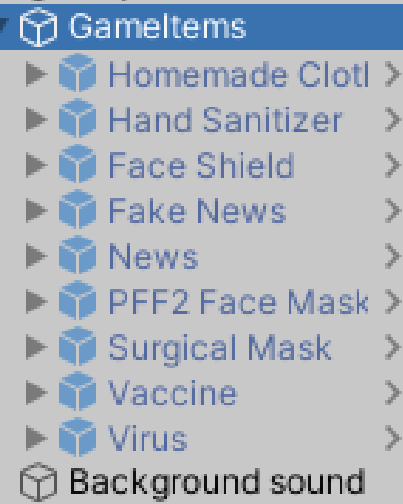
        vaccineTime = 0;

        time = -1;
    }
    else if(time >= 2)
    {
        RandomItem();

        time = 0;
    }
}
```

A vacina demora
60s para ser
ativada na
segunda fase do
jogo

0 script sorteia itens a cada 2
segundos e os ativa



Objeto vazio
GameItems que
ativa os objetos
filhos

```
void RandomItem()
{
    bool x = false;

    do{

        if(gameLevel == 1)
        {
            randomNumber = Random.Range(0,8);
        }
        else if(gameLevel == 2)
        {
            randomNumber = Random.Range(0,9);
        }
        else
        {
            randomNumber = Random.Range(0,10);
        }

        if(randomNumber == 0)
        {
            if(!HandSanitizer.activeSelf)
            {
                HandSanitizer.gameObject.SetActive(true);
                x = true;
            }
        }

        else if (randomNumber == 1)
        {
```

O sorteio é similar ao da outra entrega, mas na Fase 2 há um item a mais (Virus) e na Fase 3 há dois items a mais (Virus e Fake News)

Como os objetos são apenas ativados (e não instanciados), o método checa se já está ativo, se sim, fará um outro sorteio

```
void Update()
```

```
{
```

```
    time = time + Time.deltaTime;
```

```
    if (GameManager.Instance.GameOver)
        return;
```

```
    Vector3 curPos = transform.position;
```

```
    if(gameLevel == 3)
    {
        SpeedScale += 0.007f*time;
    }
```

```
    curPos.x -= GameManager.Instance.ScrollSpeed * SpeedScale * Time.deltaTime;
```

```
    transform.position = curPos;
```

```
    if(curPos.x < -1380)
    {
        ResetItem(gameObject);
    }
```

```
    if(clicked)
    {
        if(Input.GetButtonDown("Fire1") && gameObject.CompareTag("FN or Virus"))
        {
            PlayerController pc = GameObject.FindObjectOfType<PlayerController>();
            pc.ChangeXp(15, 2);

            ResetItem(gameObject);

            clicked = false;
        }
    }
```

```
void OnMouseEnter()
```

MovelItem.cs

Na fase 3 a velocidade aumenta com o tempo

A movimentação é similar ao do outro código e, se passar da cena, o objeto é desativado e muda-se sua posição

Detecta se o mouse passa pelo vírus ou Fake News e se há clique

```
void OnMouseEnter()
{
    clicked = true;
}
```

Correções dos problemas da última entrega

- Assim, quanto ao problema da colisão dos objetos:
 - Com os objetos fazendo parte da cena, foi possível utilizar o *OnTriggerEnter2D*
 - No *PlayerController* fez-se todo esse processo de detecção de colisão e suas consequências (aumento de pontuação, vitória e derrota do jogo e afins)

PlayerController.cs

```
private void OnTriggerEnter2D(Collider2D collision)
{
    if (GameManager.Instance.GameOver)
        return;

    else
    {
        if(collision.CompareTag("Score"))
        {
            ChangeXp(10, 0);

            Vector3 curPos = collision.gameObject.transform.position;

            curPos.x = 620;

            collision.gameObject.transform.position = new Vector3(curPos.x, curPos.y, curPos.z);

            collision.gameObject.SetActive(false);
        }
        else if(collision.CompareTag("Vaccine"))
        {
            ChangeXp(30, 1);

            BackgroundSound.SetActive(false);

            winAS.Play();

            Time.timeScale = 0;
            youWinText.gameObject.SetActive(true);
            instructionsText.gameObject.SetActive(true);

            if(gameLevel != 3)
            {
                winInstructions.gameObject.SetActive(true);
            }

            win = true;
        }
        else
        {

```

Quando ocorre a colisão, compara-se a Tag. Se for "Score", o xp será aumentado. Se for vacina, mostra-se "You win". Se for outro, "Game Over"

```
if(!GameManager.Instance.GameOver && win)
{
    if (Input.GetKey(KeyCode.Alpha1))
    {
        SceneManager.LoadScene("Level 1");

        win = false;
    }
    else if(Input.GetKey(KeyCode.Alpha2))
    {
        SceneManager.LoadScene("Level 2");

        win = false;
    }
    else if(gameLevel != 1 && Input.GetKey(KeyCode.Alpha3))
    {
        SceneManager.LoadScene("Level 3");

        win = false;
    }
    else if(Input.GetButtonDown("Jump"))
    {
        SceneManager.LoadScene("Menu");

        win = false;
    }
}
else if(!GameManager.Instance.GameOver && lose)
{
    if (Input.GetKey(KeyCode.Alpha1))
    {
        SceneManager.LoadScene("Level 1");

        lose = false;
    }
    else if(gameLevel != 1 && Input.GetKey(KeyCode.Alpha2))
    {
        SceneManager.LoadScene("Level 2");

        lose = false;
    }
}
```

De acordo com a vitória ou derrota, faz-se a detecção dos inputs para se prosseguir com as telas e o jogo

Ainda neste script, temos:

```
public void ChangeXp(int points, int i)
{
    if(i == 1)
    {
        vaccineAS.Play();
    }
    else if(i == 2)
    {
        xpAS.Play();
    }
    else
    {
        scoreAS.Play();
    }

    xp += points;

    xpText.text = "xp: " + xp;
}
```

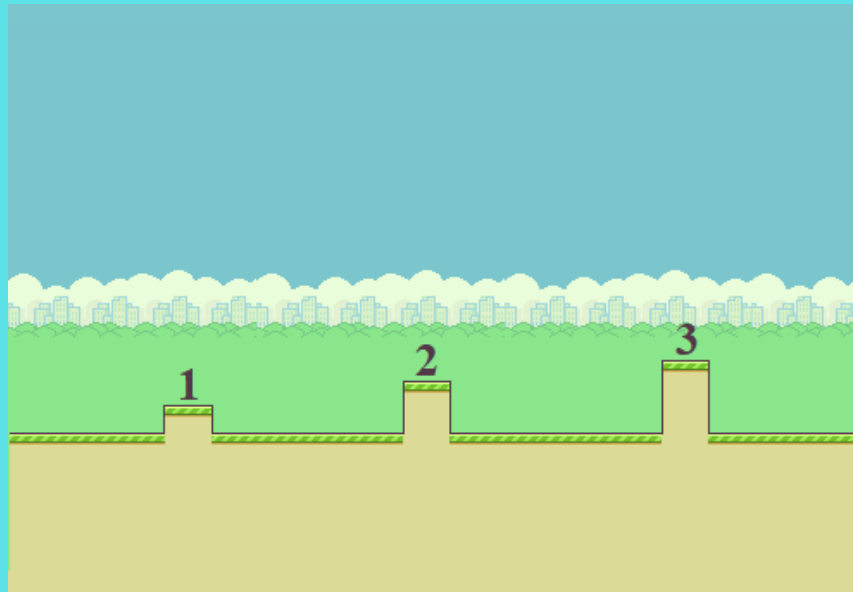
Método que aumenta o xp e coloca o som de acordo com a colisão

Adição de complementos

- Além de corrigir os problemas, também houve a adição de efeitos sonoros: de fundo, e de acordo com a colisão e movimentação do *Player*
- Como mencionado anteriormente, também se tornou possível que os vírus e as fake news fossem “exterminados” com um clique
- Há ainda a contagem de xp, a qual não ocorria na última entrega

Mapa das fases

- O mapa das fases foi utilizado para compor as cenas Level 1, Level 2 e Level 3, assim como mencionado anteriormente



Fase 2

- A fase 2 foi feita da mesma forma da Fase 1, exceto pelas seguintes diferenças:
 - A vacina é ativada somente com 60 segundos de jogo, e não 30, como na fase 1
 - Houve a adição de um objeto a mais: um segundo vírus. Assim, o sorteio aleatório contava com um número a mais a ser sorteado, correspondendo a esse objeto
 - Pequenas alterações no Canvas, exibindo, por exemplo, a possibilidade de ir para a fase 3 em caso de vitória



Fase 3

- A fase 3 foi feita de maneira análoga à fase 2, exceto pelas seguintes diferenças:
 - Não há vacina, pois a personagem já tomou as duas doses
 - Houve a adição de um objeto a mais: uma segunda fake news. Assim, o sorteio aleatório contava com dois números a mais em comparação a fase 1, correspondendo a esses objetos
 - Pequenas alterações no Canvas, exibindo, por exemplo, a possibilidade de jogar as 3 fases, pois, afinal, a personagem já passou pelas 3 fases

