Instituto Federal de São Paulo (IFSP)

Curso Técnico em Desenvolvimento de Sistemas

**Integrantes:** Arthur Fukuyama de Andrade; Heitor Akira Isatugo; Leonardo Cabrera Corrêa; Rafael Fujita Masuko

**Fear’s Least Frame**

São Paulo

2025

Instituto Federal de São Paulo (IFSP)

Curso Técnico em Desenvolvimento de Sistemas

**Integrantes:** Arthur Fukuyama de Andrade; Heitor Akira Isatugo; Leonardo Cabrera Corrêa; Rafael Fujita Masuko

**Fear’s Least Frame**

Documentação de Projeto Interdisciplinar apresentado como requisito parcial para avaliação nas disciplinas de Lógica de Programação, Princípios de Desenvolvimento Web, Arquitetura de Computadores e Redes e Fundamentos do Desenvolvimento de Sistemas.

**Professores (Banca):** Ana Lucia Grici Zacarin Mamede; Claudete de Oliveira Alves; Claudia Miyuki Werhmuller; Antonio Ferreira Viana; Daniela dos Santos Santana; André Evandro Lourenço; João Antonio Temochko Andre; Paula Neves de Araújo; Paulo Henrique Netto de Alcantara

São Paulo

2025

# INTRODUÇÃO

O desafio proposto pelos professores foi criar um jogo do zero, aprendendo uma nova linguagem de programação e colocando em prática o que foi aprendido ao longo do curso. O objetivo deste documento é registrar como foi o processo de criação e desenvolvimento do projeto.

No começo, pensamos em fazer um jogo sobre Pokémon, mas depois decidimos criar algo original. Assim surgiu o Fear’s Last Frame, um jogo de ficção no estilo RPG, onde o protagonista Felipe, um menino autista, precisa derrotar monstros que aparecem em um parque misterioso. Cada vitória leva o personagem mais perto dos brinquedos e de um novo amigo, representando uma jornada de superação e coragem.

Escolhemos o gênero RPG porque ele permite mais liberdade para criar histórias, personagens e mecânicas diferentes. O público-alvo do jogo são principalmente crianças e jovens, já que é um estilo divertido e cheio de possibilidades.

Neste documento estão descritas as ideias que tivemos, as ferramentas que usamos, como nos organizamos e as etapas que seguimos até o jogo ficar pronto.

# PLANEJAMENTO DO PROJETO

Não analisamos jogos semelhantes, mas assistimos a tutoriais no YouTube dos canais Game Code Library e Two Tv Games e realizamos pesquisas para aprender a nova linguagem.  
O jogo é uma plataforma 2D em que o jogador derrota monstros e ganha níveis conforme progride.  
As principais ferramentas utilizadas foram:

* **Linguagem:** C# (pela semelhança com C)
* **Engine:** Unity (por ser otimizada e amplamente usada)
* **Outros programas:** Tiled Map Editor, Piskel, RPG Maker MZ e Pixilart (para sprites); VS Code (para o site); e IA (para correção de erros).

**Divisão da equipe:**

* Leonardo — documentação e mapa
* Arthur — parte gráfica e site
* Heitor — storyboards e site
* Rafael — código do jogo

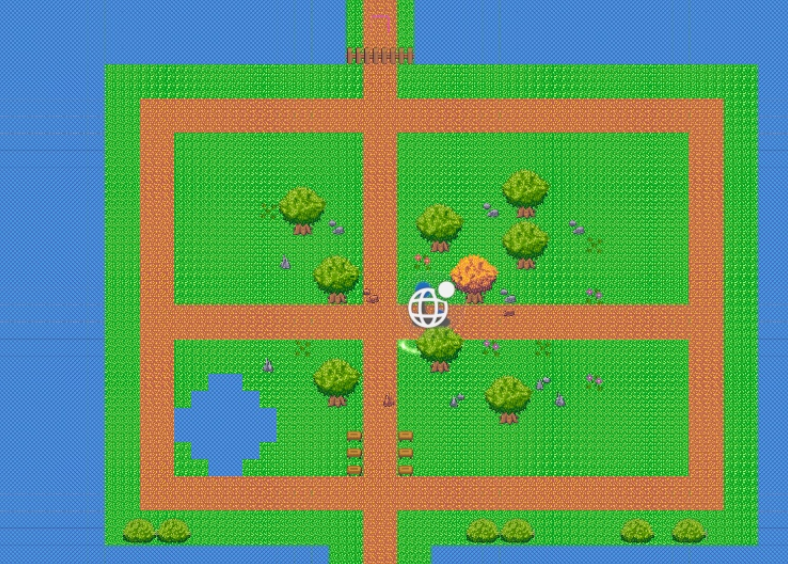
As principais decisões envolveram a divisão de tarefas conforme as habilidades de cada integrante e a escolha das ferramentas a serem usadas.

# DESENVOLVIMENTO

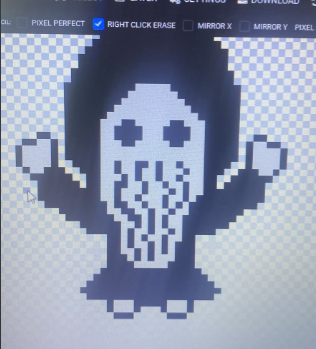
## Design e Arte

O estilo visual escolhido foi **pixel art**.  
**Personagens:**

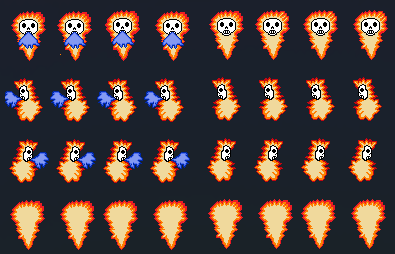
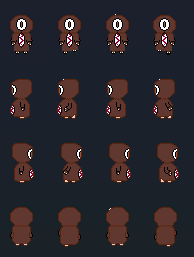
* Felipe (protagonista)
* Cumacanga (inimigo comum)
* Mapinguari (mini boss)
* Bradador (boss)
* Curupira (NPC)

Mapa

Inimigo 1 - Bradador Felipe (protagonista)

Inimigo 2 – Cumacanga Inimigo 3 - Mapinguari

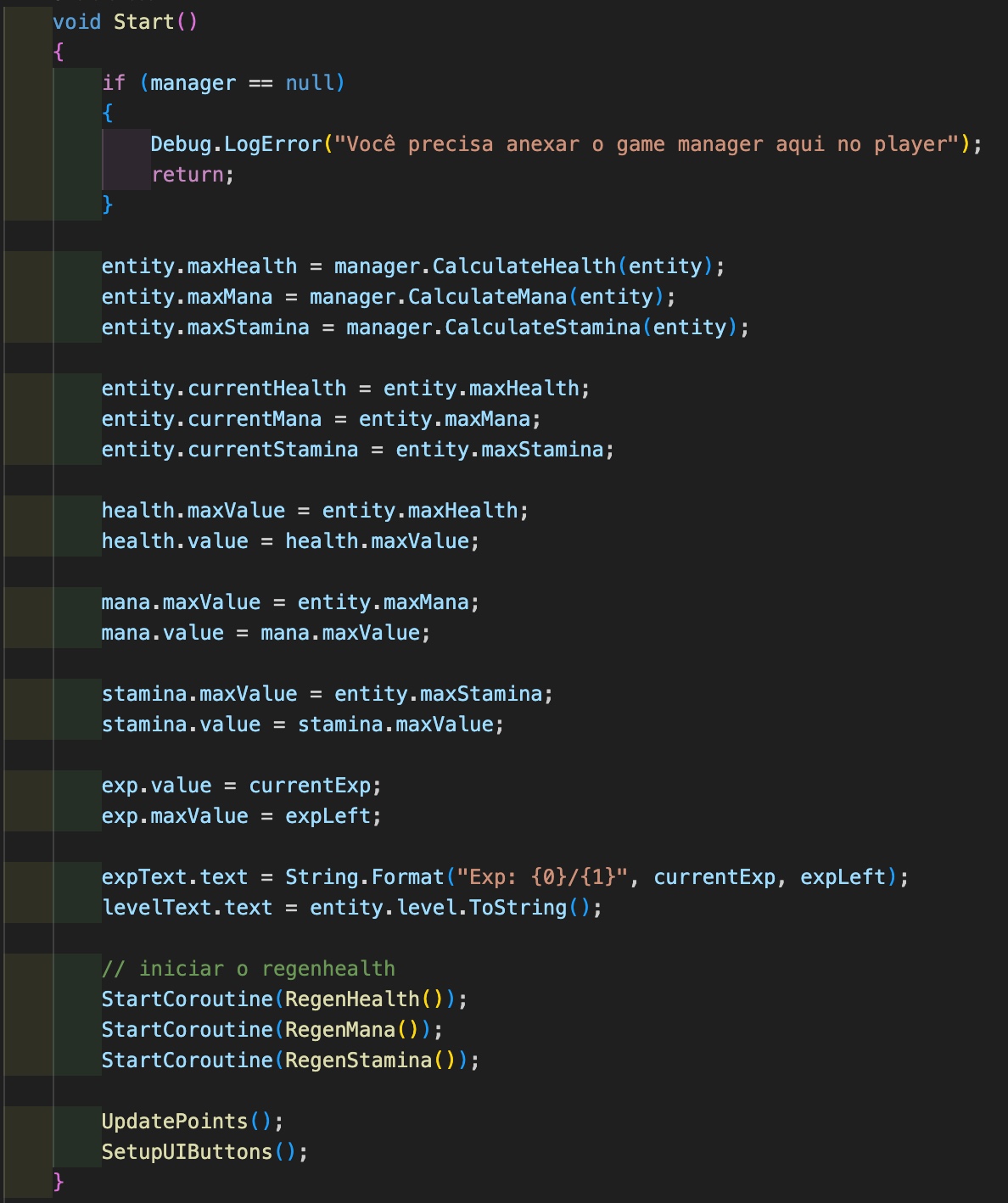
Iminigo – Bradador Curupira

## Arquitetura e Código

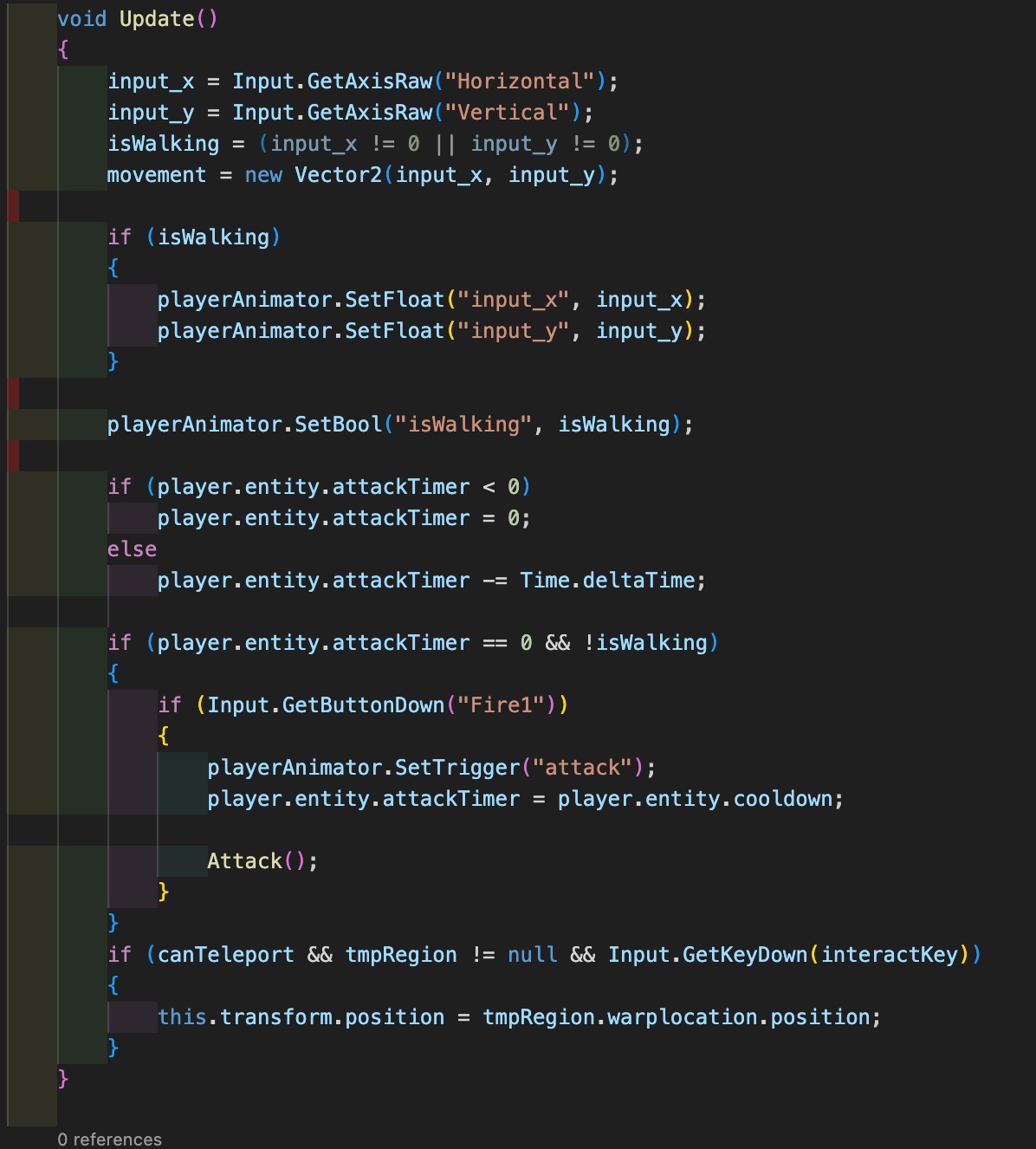
O jogo utiliza mecânicas básicas de movimentação, ataque e progressão de personagem. O objetivo principal é derrotar os inimigos e chegar ao parquinho no final do jogo. Toda a lógica foi construída em C# dentro da Unity, que gerencia a física, colisões e o comportamento dos personagens.

A **lógica principal** do jogo se baseia em sistemas que controlam vida, mana, estamina, experiência e level do jogador. Foram utilizadas **estruturas condicionais (if/else)** para verificar situações como a quantidade de vida do personagem, se ele pode regenerar status ou realizar ataques. Além disso, foram aplicados **laços de repetição (loops)** para manter o movimento contínuo do player e para atualizar as funções de regeneração de forma automática durante o jogo.

Um dos elementos mais importantes do código é a função de **Level Up**, responsável por evoluir o personagem conforme ele acumula experiência.



Essa parte é onde atribui os status calculados ao player e chama as funções de regeneração de vida, mana e estamina, logo depois chama as funções de alteração de status, com os botões adicionados para realizar essa alteração.

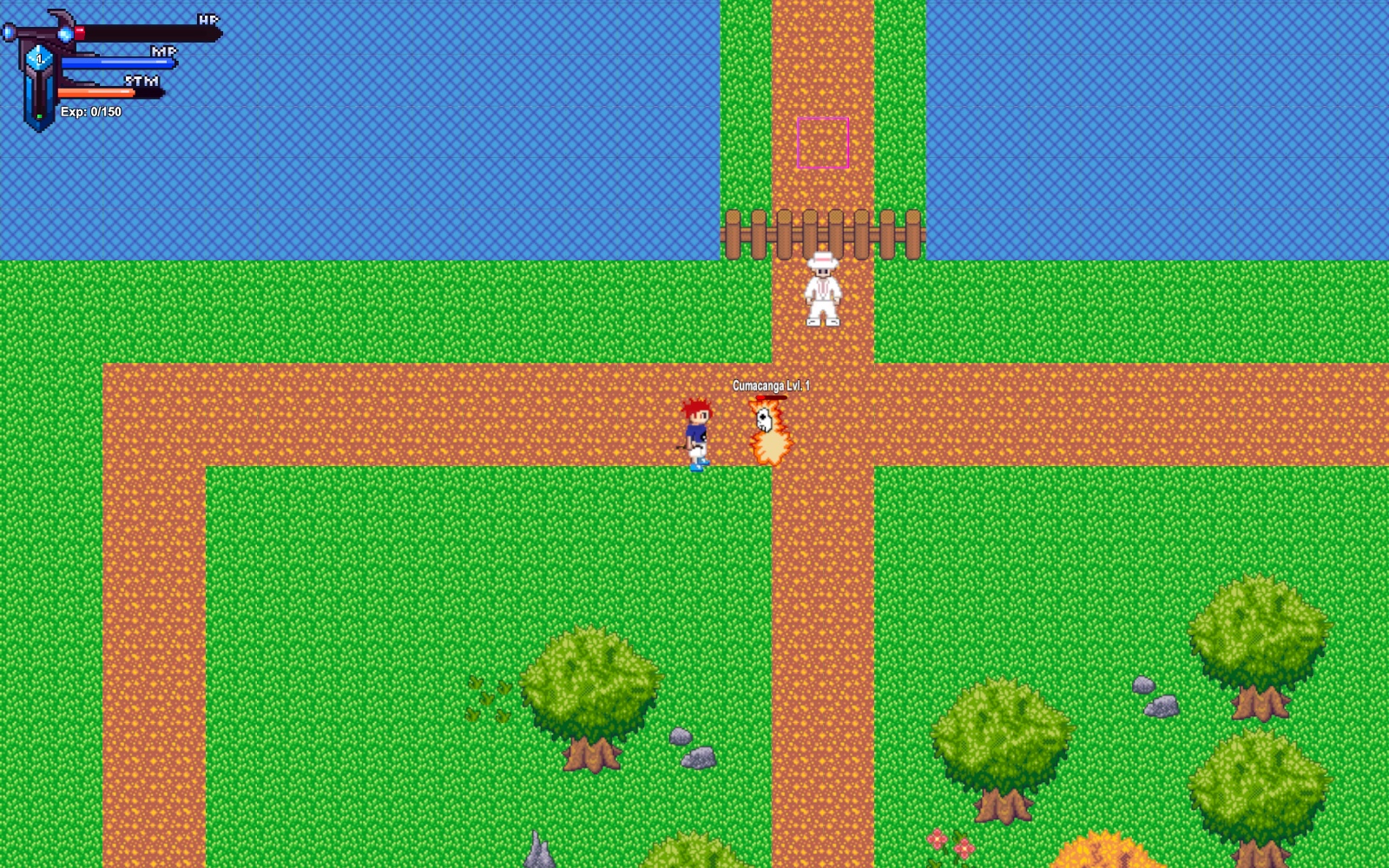
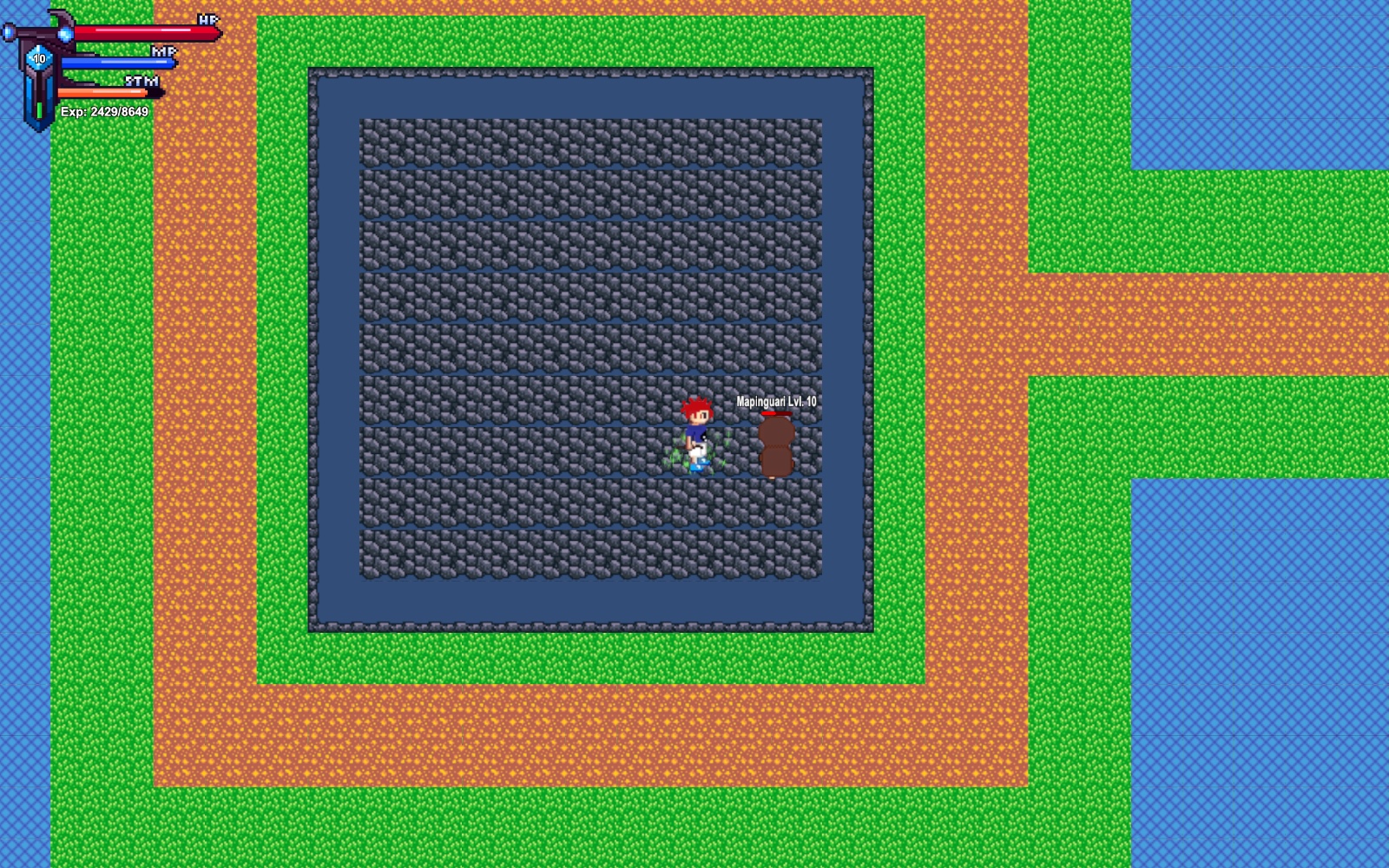


Nessa parte está integrada a movimentação do personagem, mostrando também o ataque, chamando o botão esquerdo do mouse para realiza-lo e também declarando o botão, e para interagir com o teleporte.

## O Site de Divulgação

# PROTÓTIPO E TESTES

Durante os testes, encontramos muitos erros, especialmente em combates — os inimigos atravessavam paredes e andavam sobre a água.  
Foi necessário usar IA para detectar e corrigir alguns bugs.

Link da gameplay: https://youtu.be/7-utVs1nyA0

Link do Tutorial: https://youtu.be/ZQ09vB8Kp8I?si=3-o\_krXSbFqNMi36

# CONCLUSÃO

Fazer o jogo foi uma experiência muito boa e cheia de aprendizados. Durante o processo, conseguimos melhorar nossa **organização** e o **trabalho em equipe**, já que cada um ficou responsável por uma parte diferente e todas precisavam se encaixar para o jogo funcionar.

Mesmo que o resultado final tenha ficado diferente da ideia inicial, que seria um jogo sobre Pokémon, achamos que essa nova versão ficou bem mais criativa e original. Criar o personagem Felipe e pensar em toda a história por trás dele foi algo que deixou o projeto mais interessante.

As **maiores dificuldades** foram aprender a usar as ferramentas certas para fazer o jogo, principalmente a Unity, que no começo parecia bem complicada. Também tivemos que lidar com vários **erros e bugs**, e às vezes usamos IA para nos ajudar a entender o que estava dando errado e como corrigir.

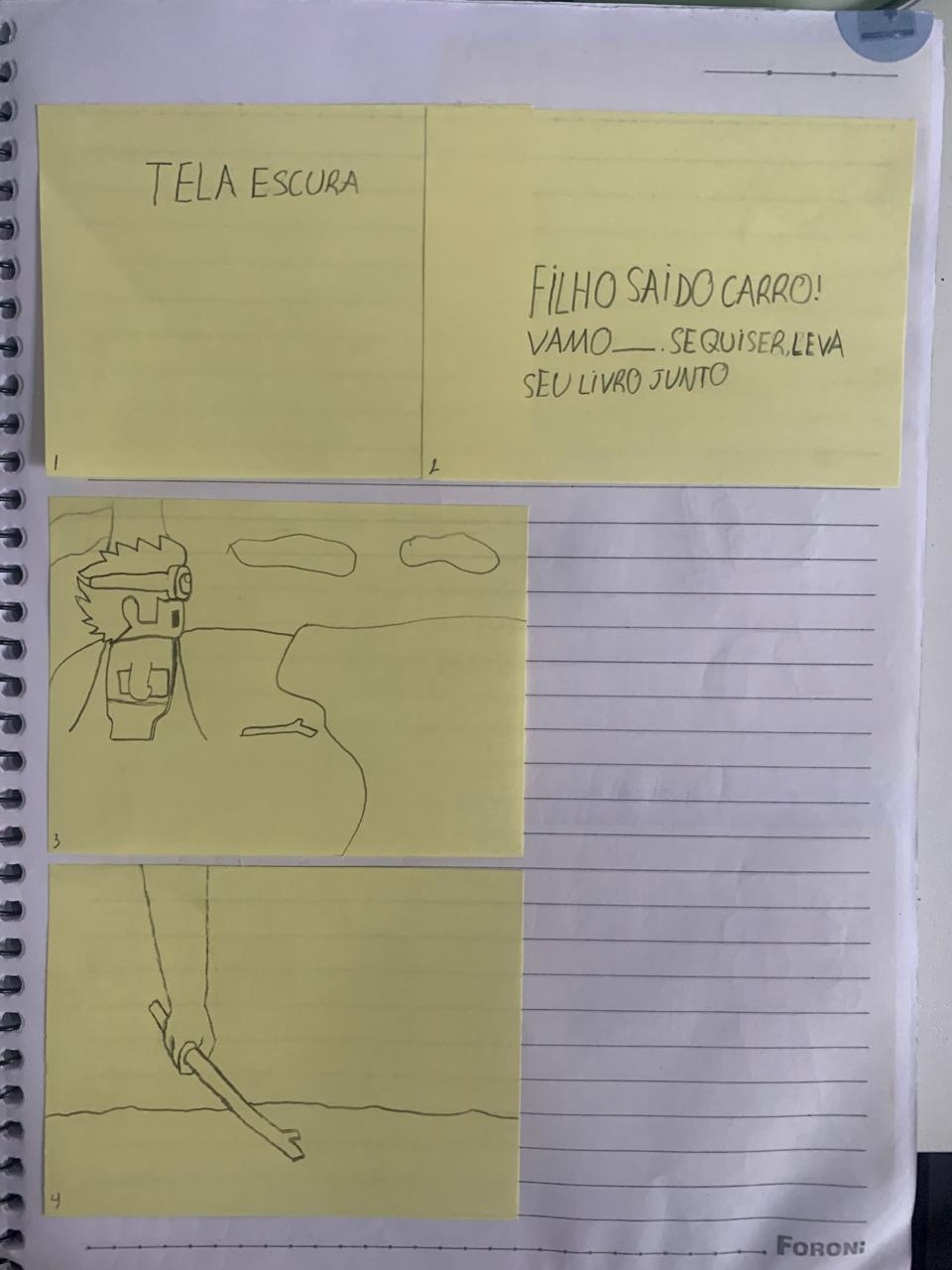
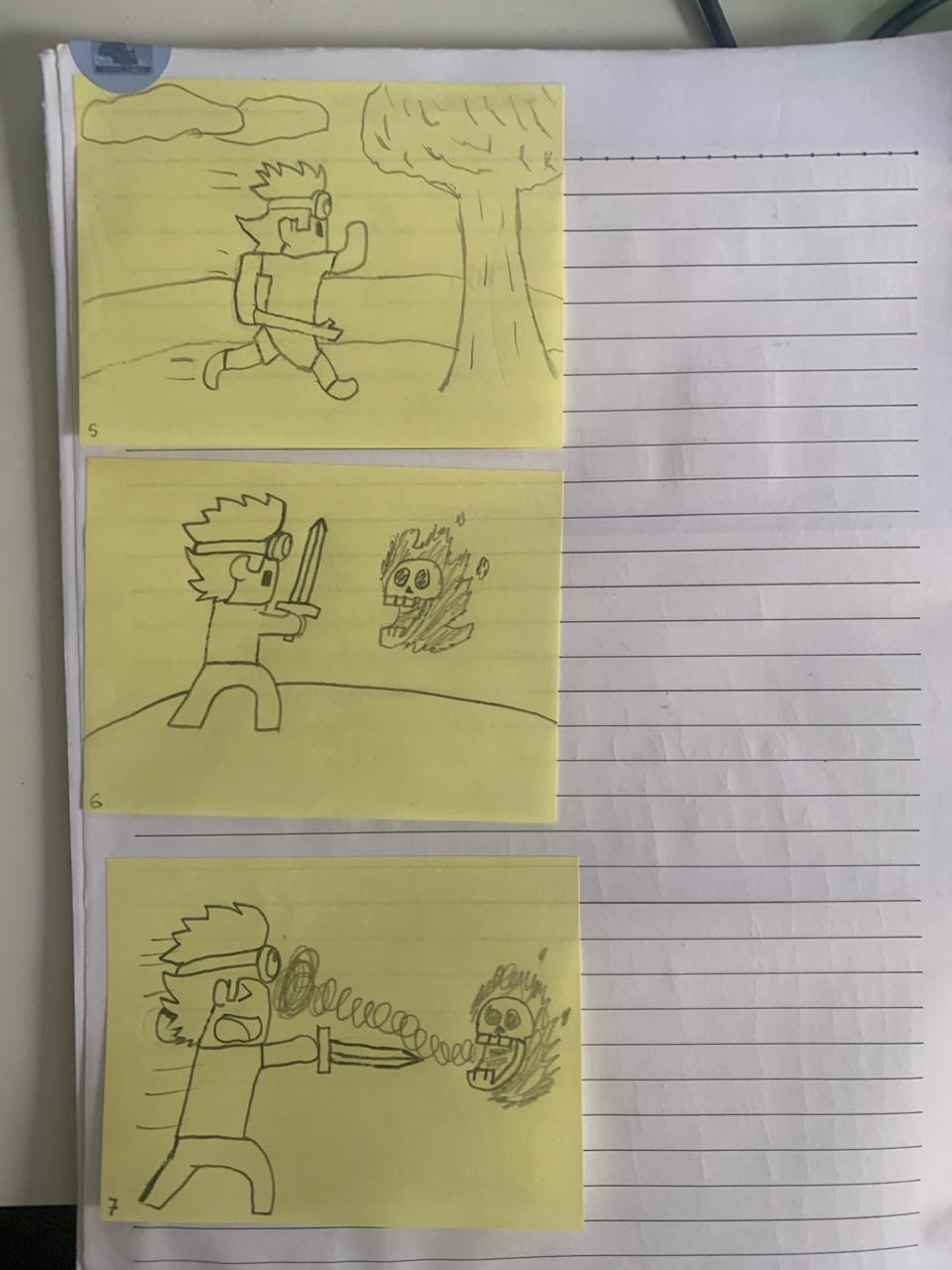
A parte **gráfica** foi uma das mais legais de fazer, porque deu pra usar bastante criatividade nos desenhos e nas ideias. Já a programação foi mais trabalhosa, mas ver o jogo funcionando depois de tanto esforço foi muito gratificante.

Se tivéssemos **mais tempo**, gostaríamos de colocar mais personagens, novos inimigos e um mapa maior, além de criar mais mecânicas para deixar o jogo mais divertido. No fim, ficamos bem satisfeitos com o que conseguimos fazer e com tudo o que aprendemos nesse projeto.

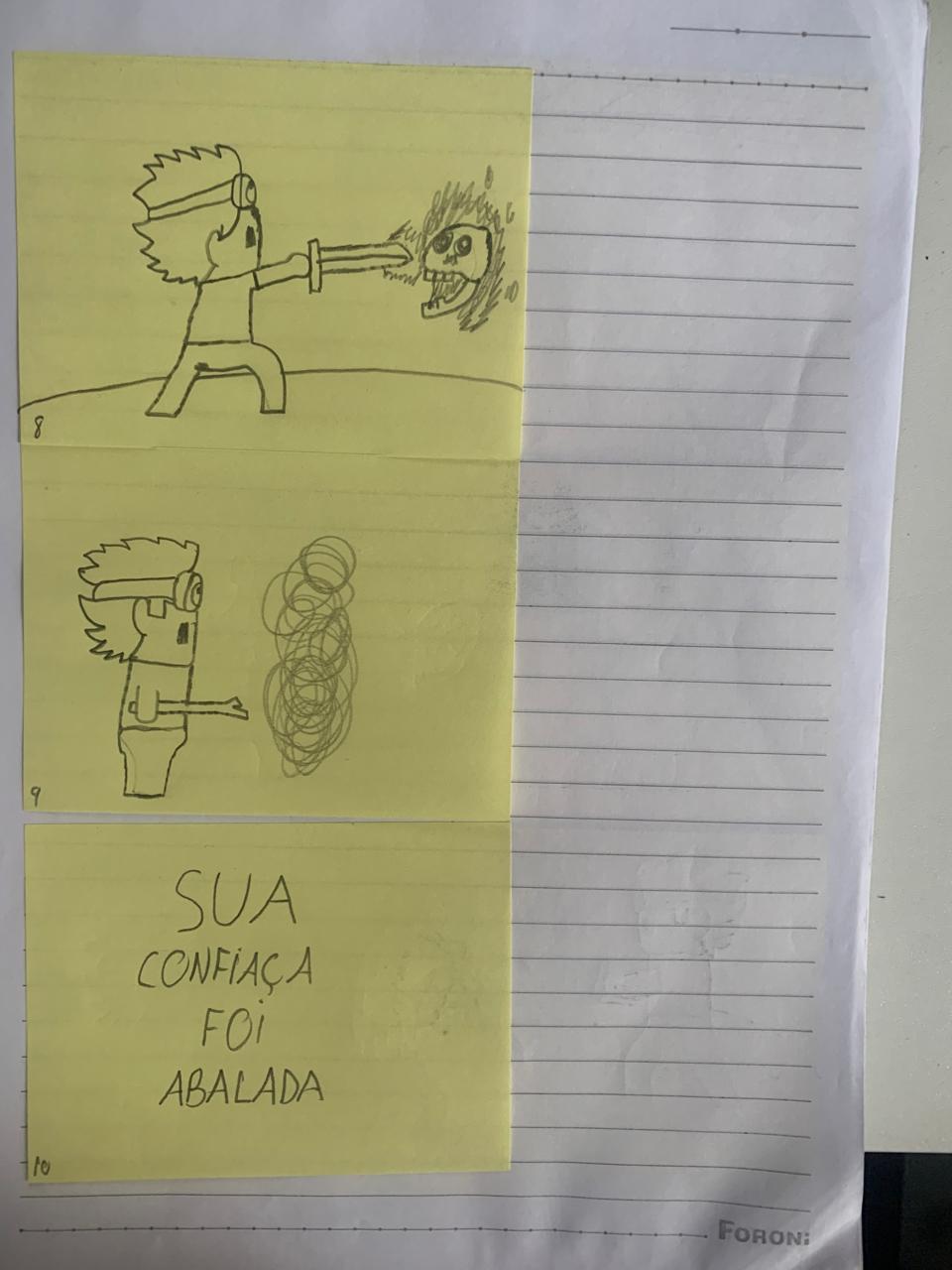
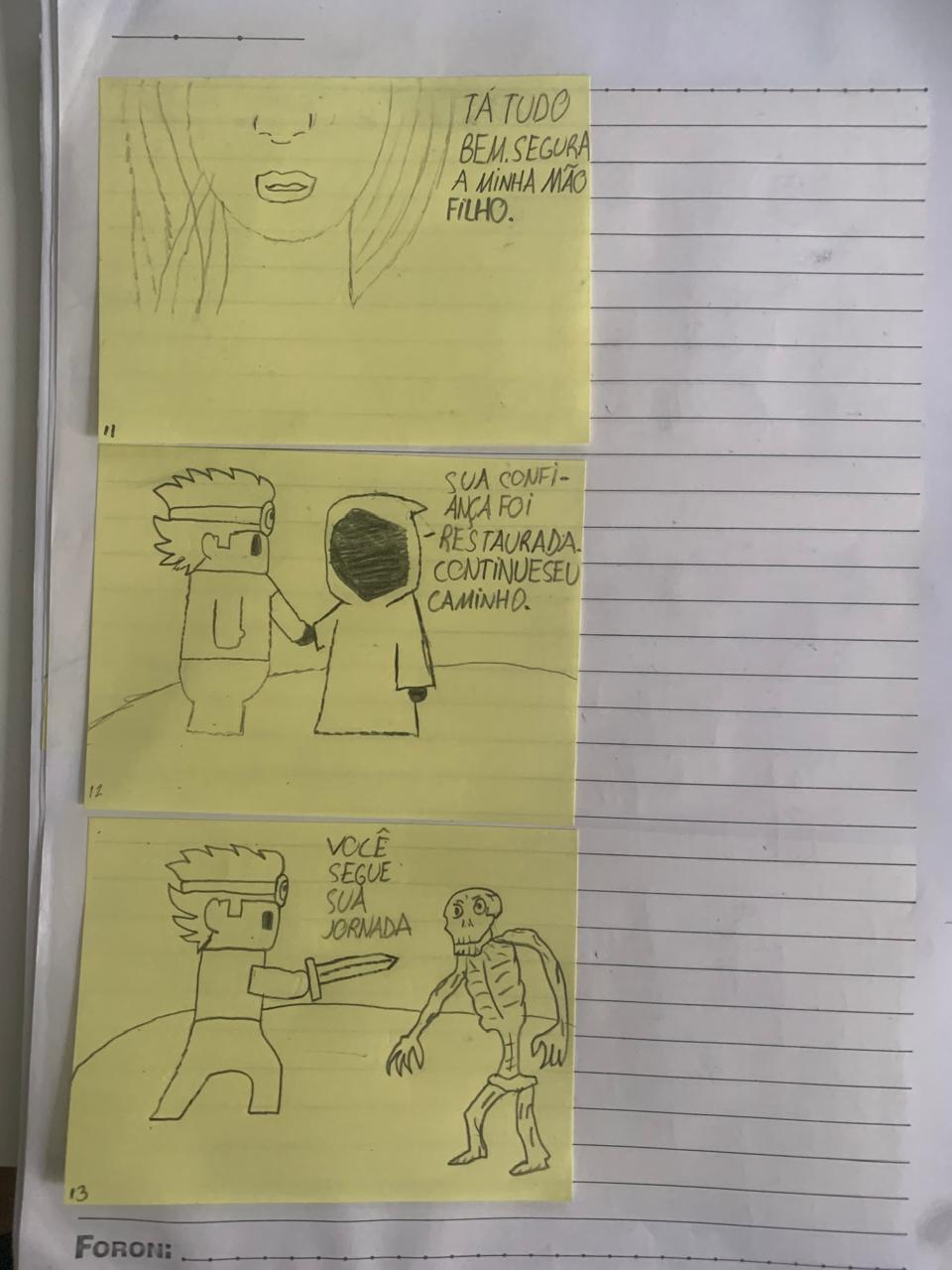
# REFERÊNCIAS

* **DOCUMENTAÇÃO oficial da Unity.**
* **FERRAMENTAS utilizadas:** Unity, Tiled Map Editor, Piskel, RPG Maker MZ, Pixilart, Visual Studio Code e Inteligência Artificial (para identificar e corrigir erros).
* **GAME CODE LIBRARY.** Tutoriais de desenvolvimento de jogos na Unity. YouTube, 2024.  
  Disponível em: <https://www.youtube.com/watch?v=4-bu0YhbVE0&list=PLm43vXIKl0F-1uxmGW84NmlJh6dob296f>.  
  Acesso em: 7 nov. 2025.
* **SITES de pesquisa** sobre desenvolvimento de jogos 2D.
* **TUTORIAIS do YouTube** sobre Unity e C#.
* **TWO TV GAMES.** Dicas e tutoriais de criação de jogos 2D na Unity. YouTube, 2024.  
  Disponível em: <https://www.youtube.com>.  
  Acesso em: 7 nov. 2025.
* **UNITY TECHNOLOGIES.** Criação e desenvolvimento de jogos com Unity. Unity, 2025.  
  Disponível em: <https://unity.com/pt/games>.  
  Acesso em: 7 nov. 2025.

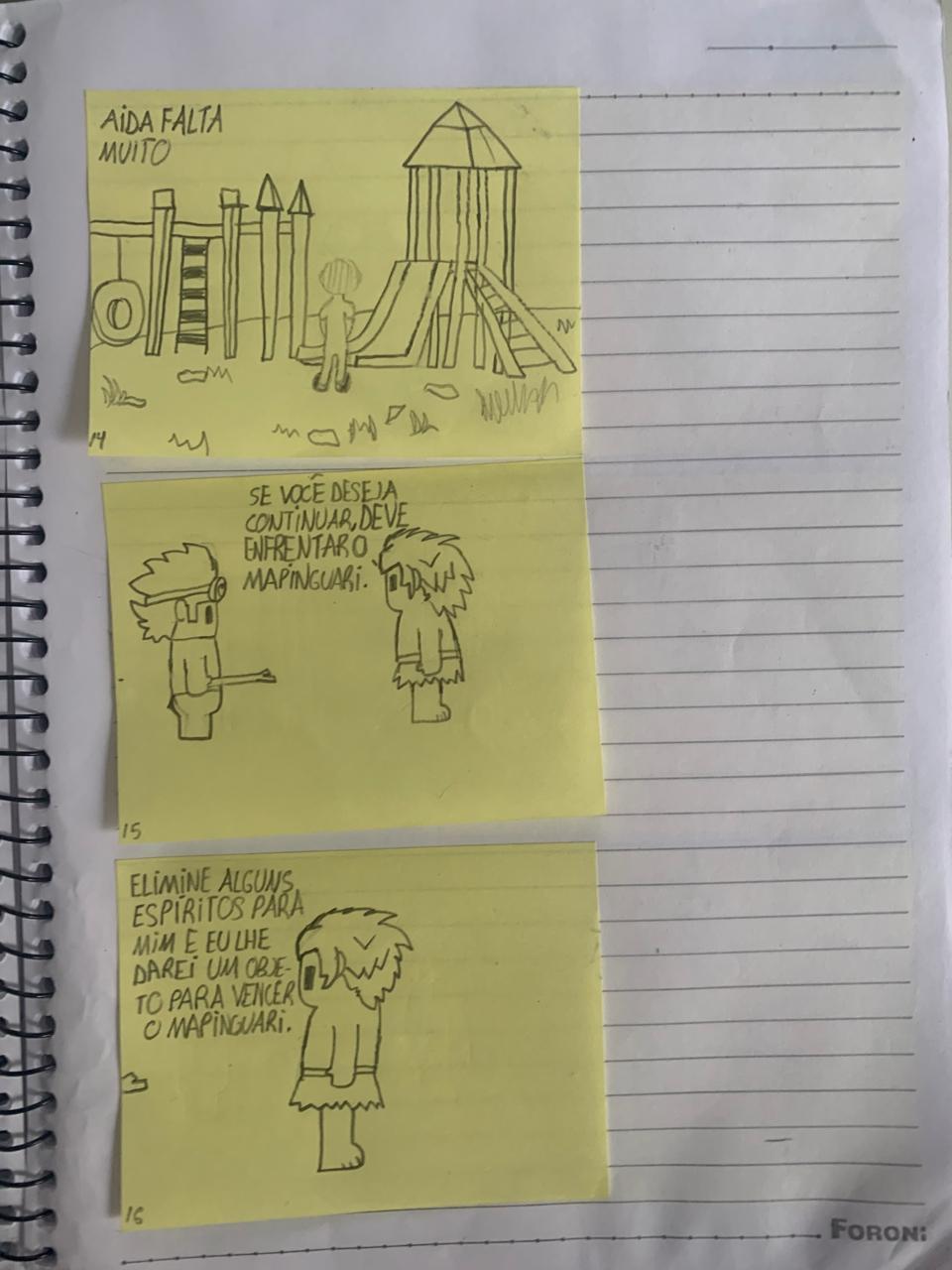
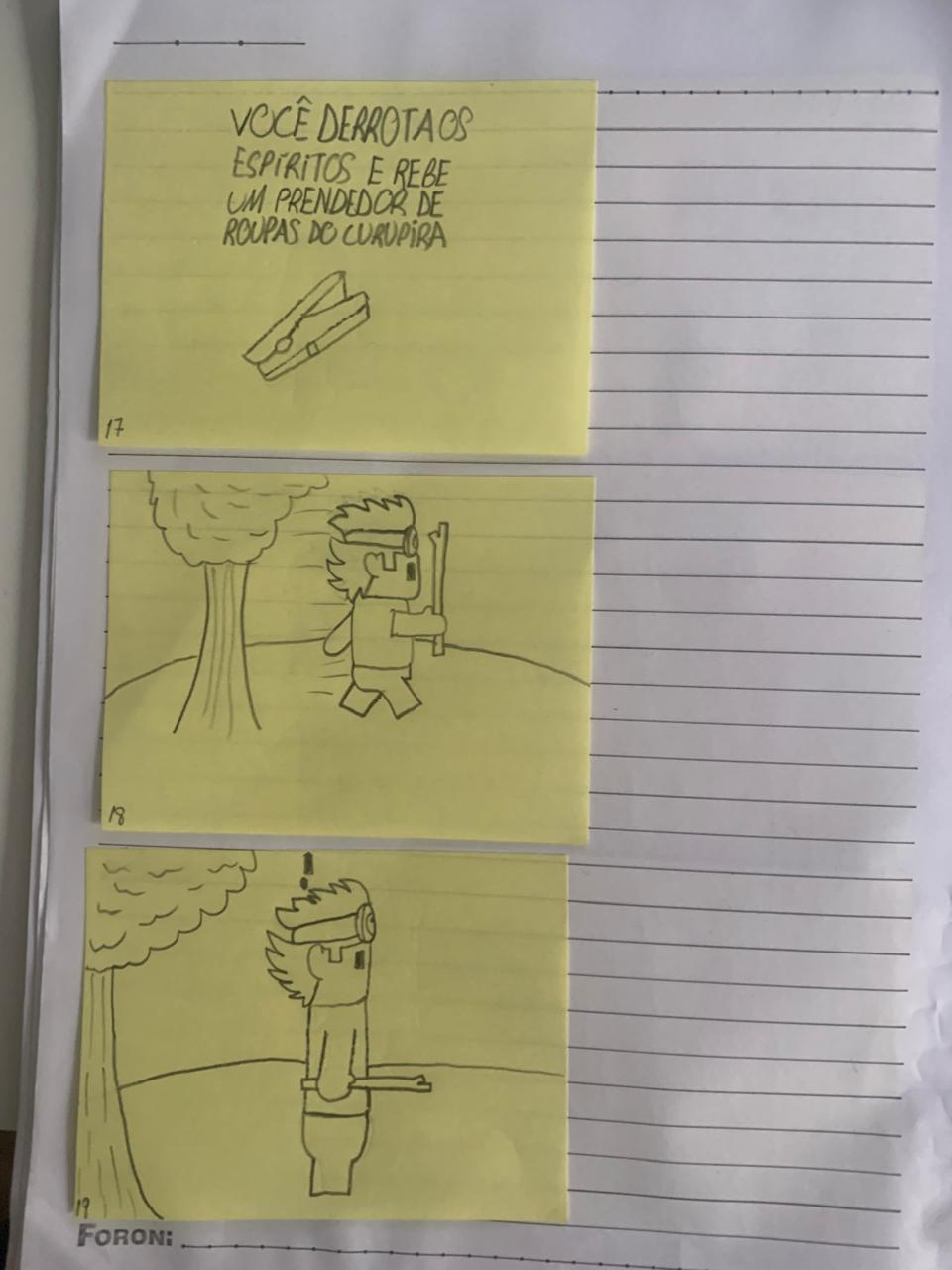
**APÊNDICE A – STORYBOARDS DO JOGO**

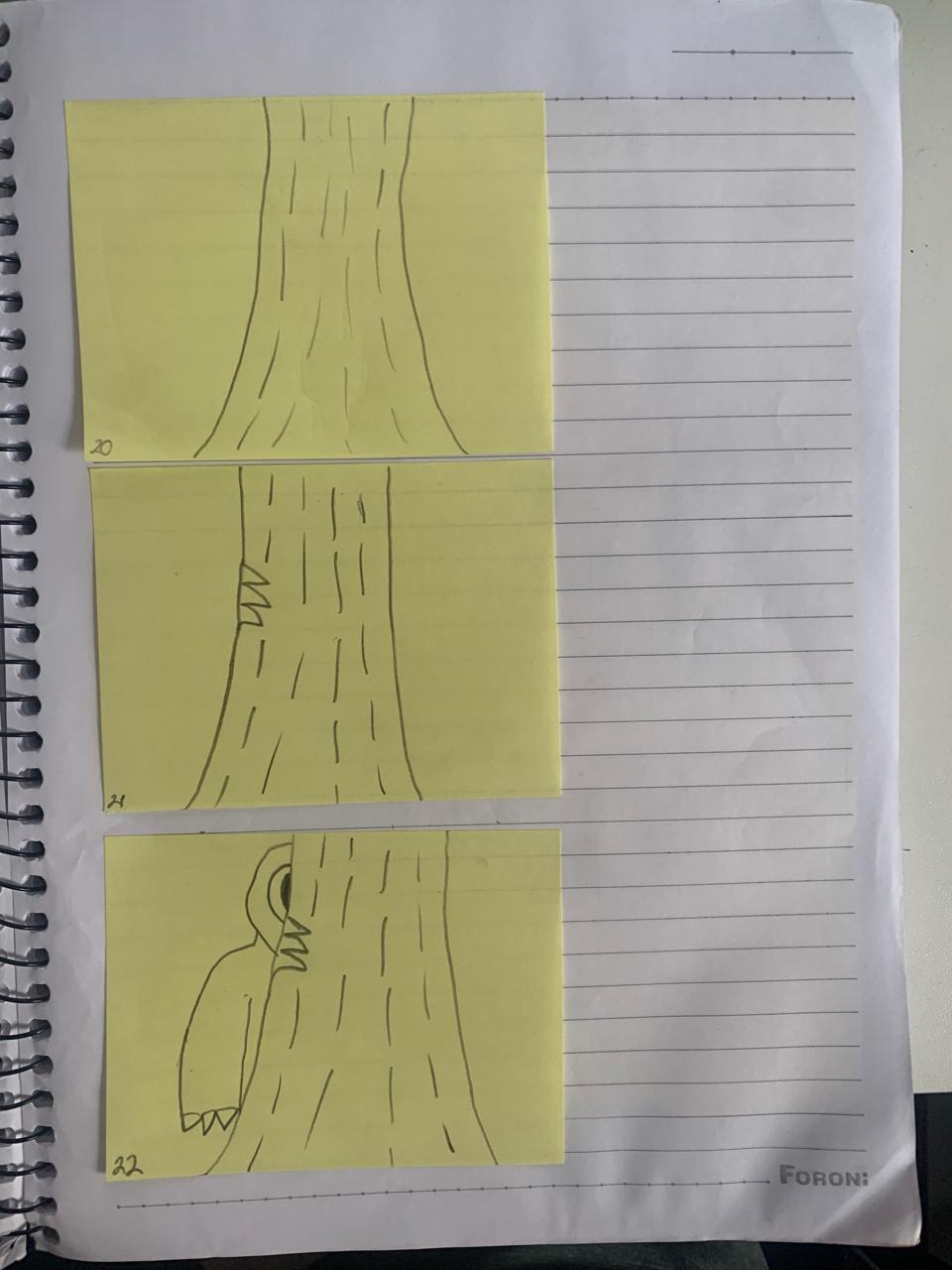
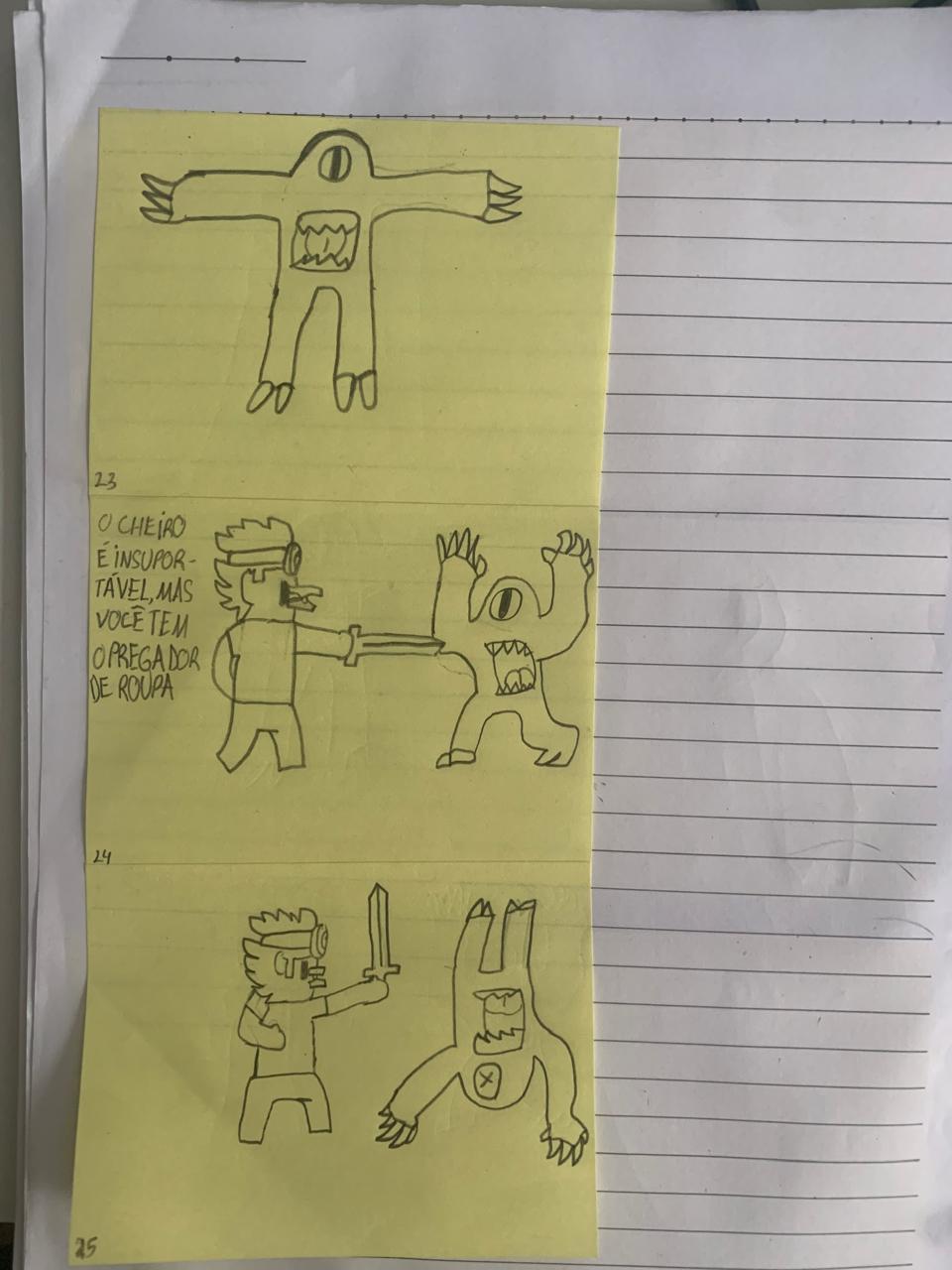
Cena 1 Cena 2

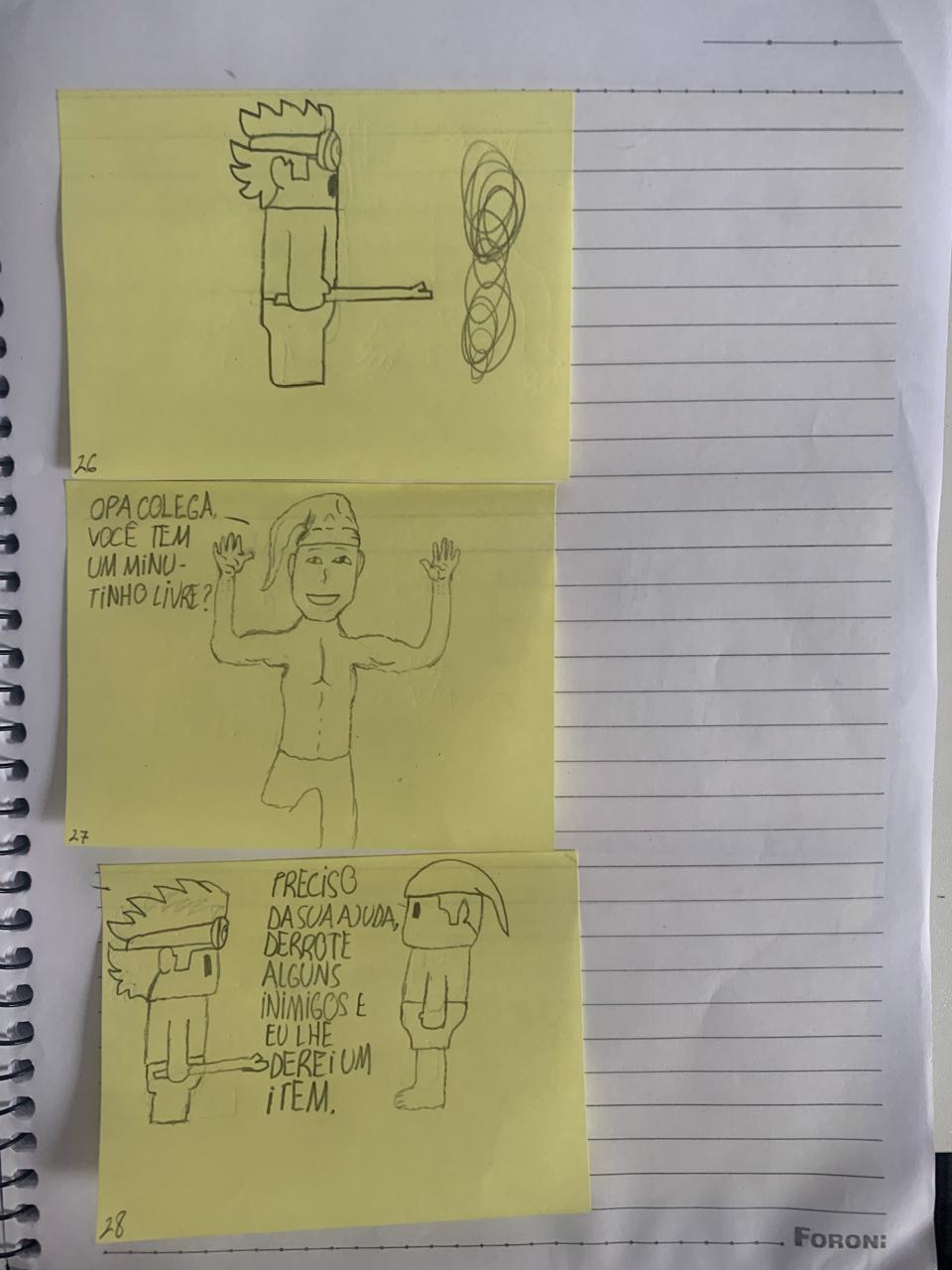
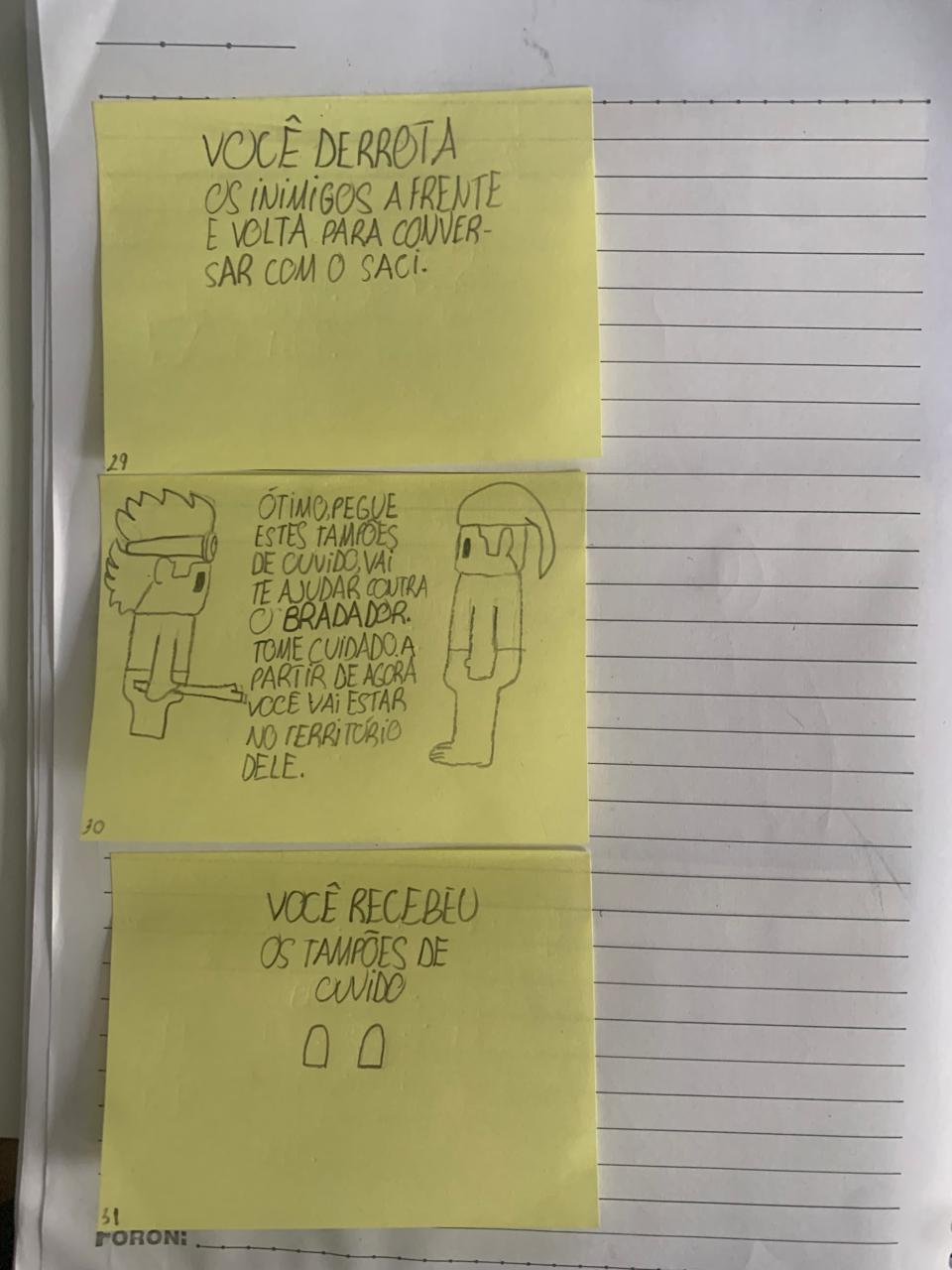
Cena 3 Cena 4

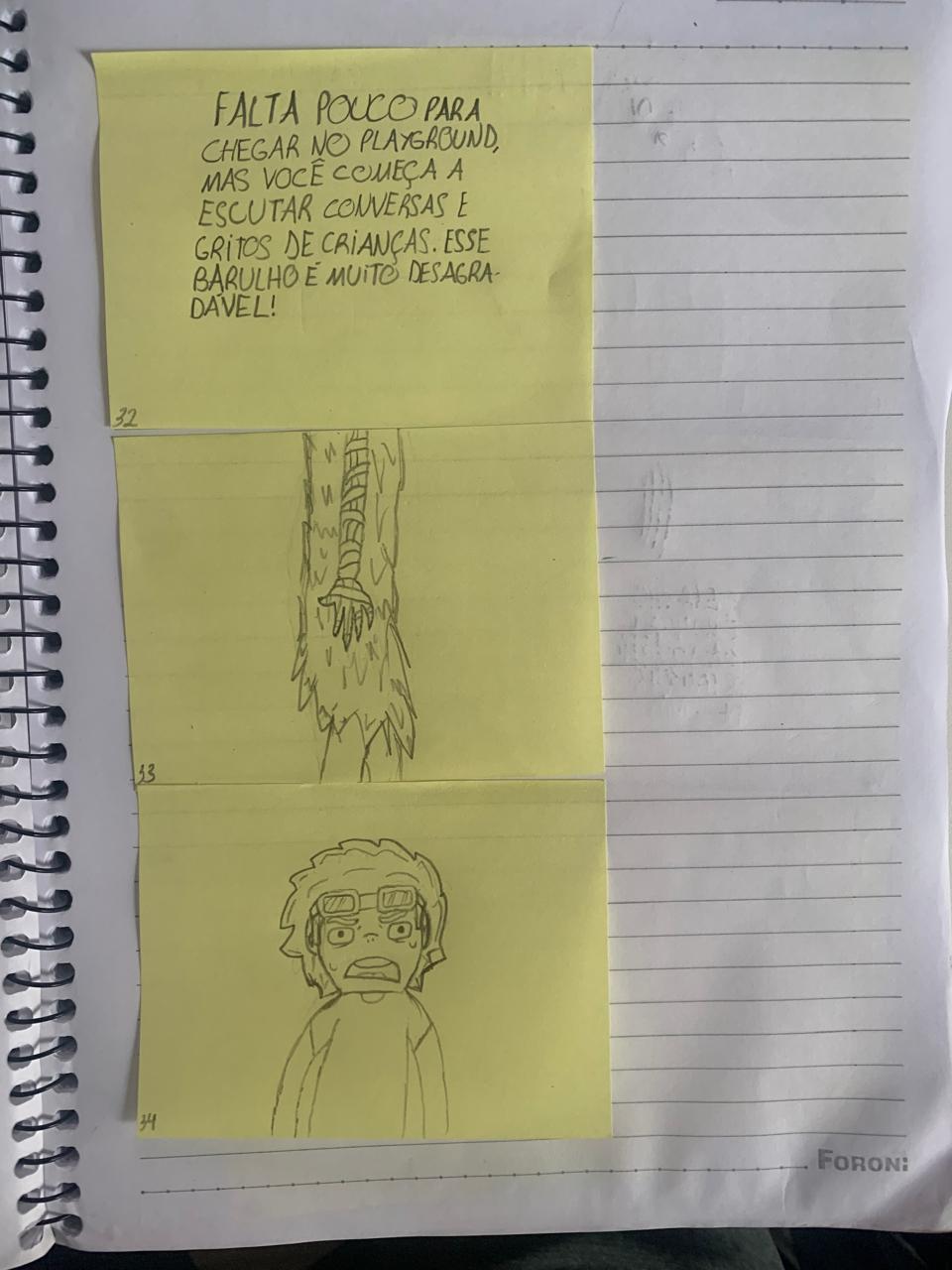
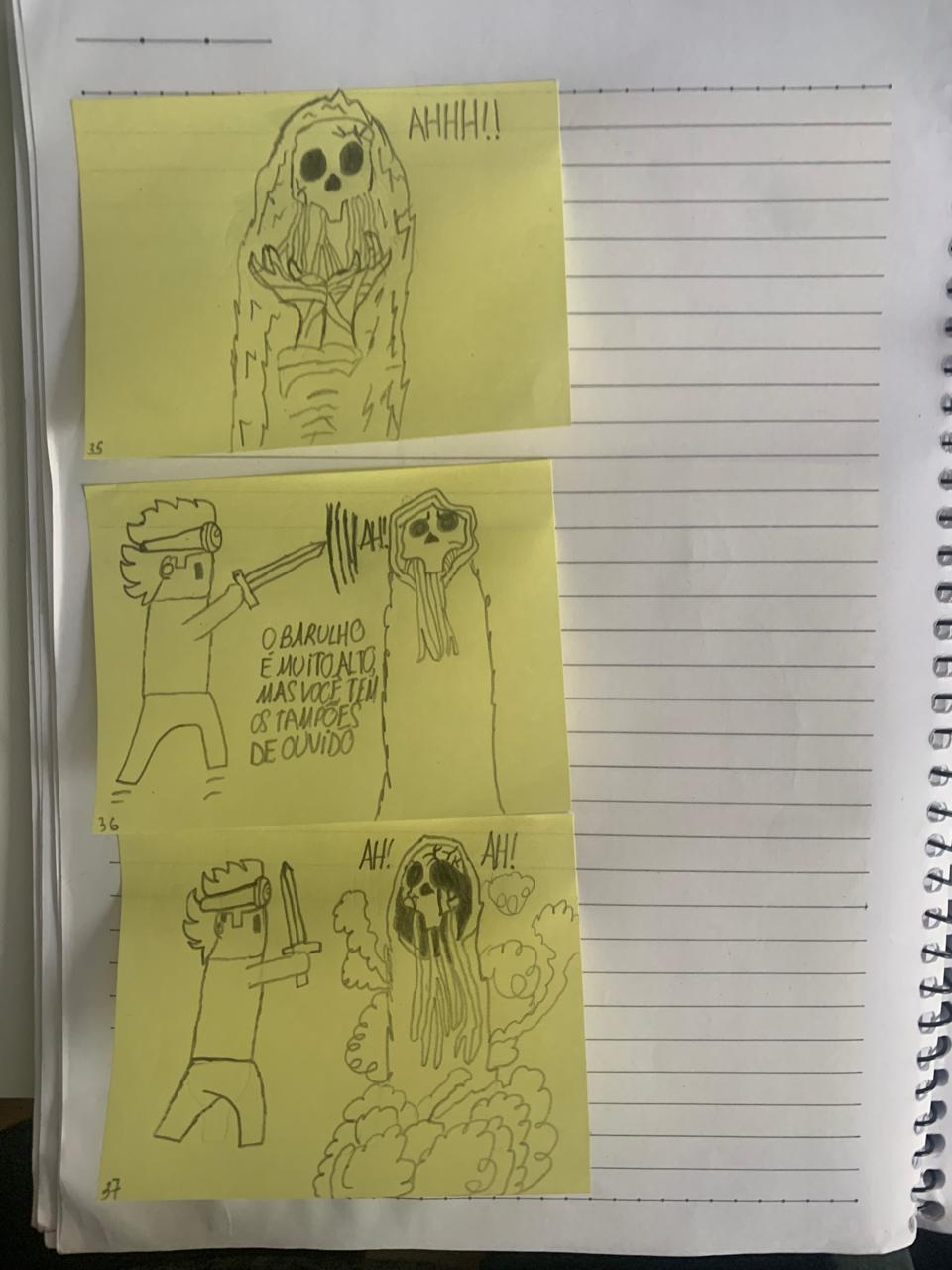
Cena 5 Cena 6

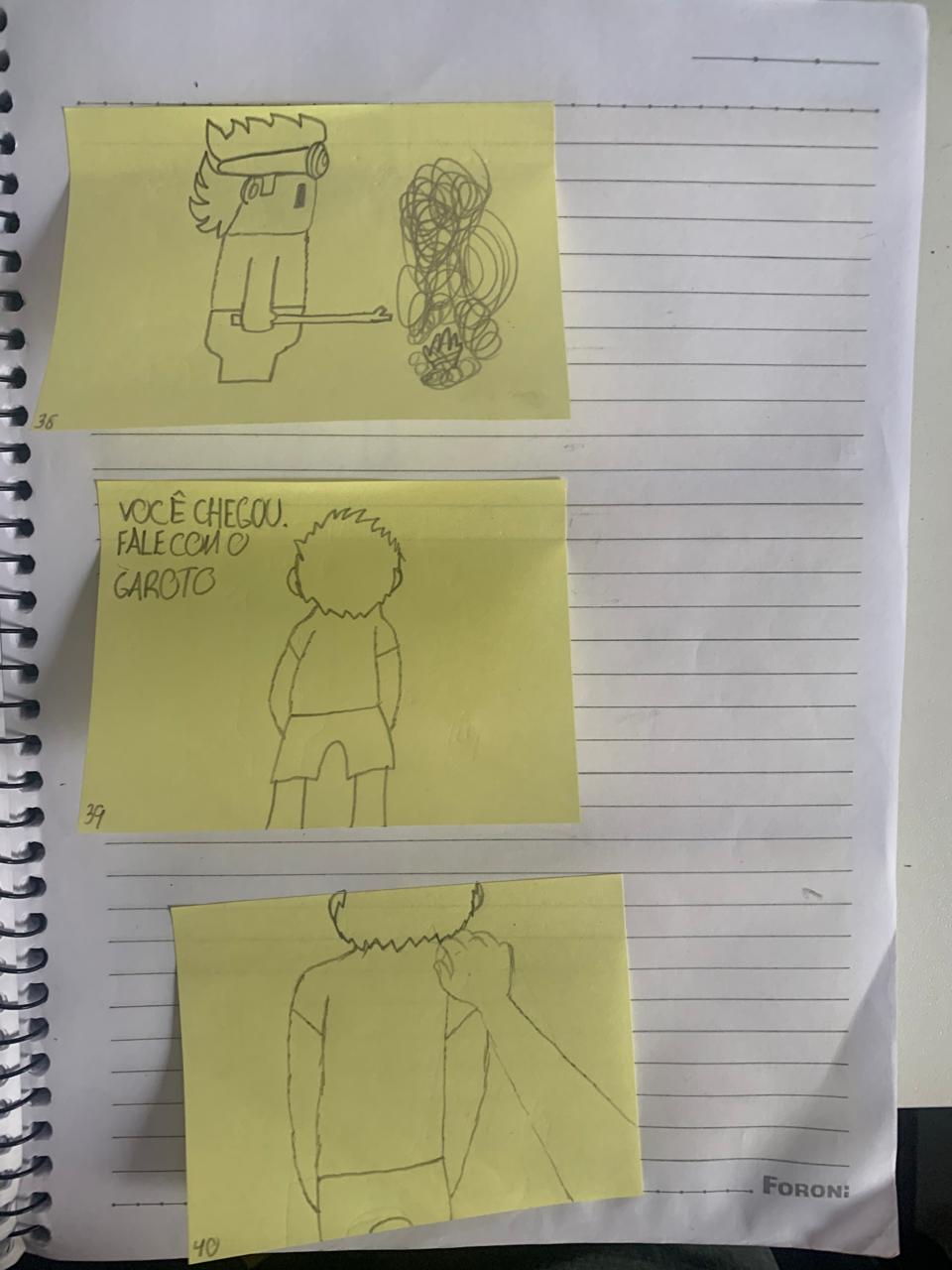
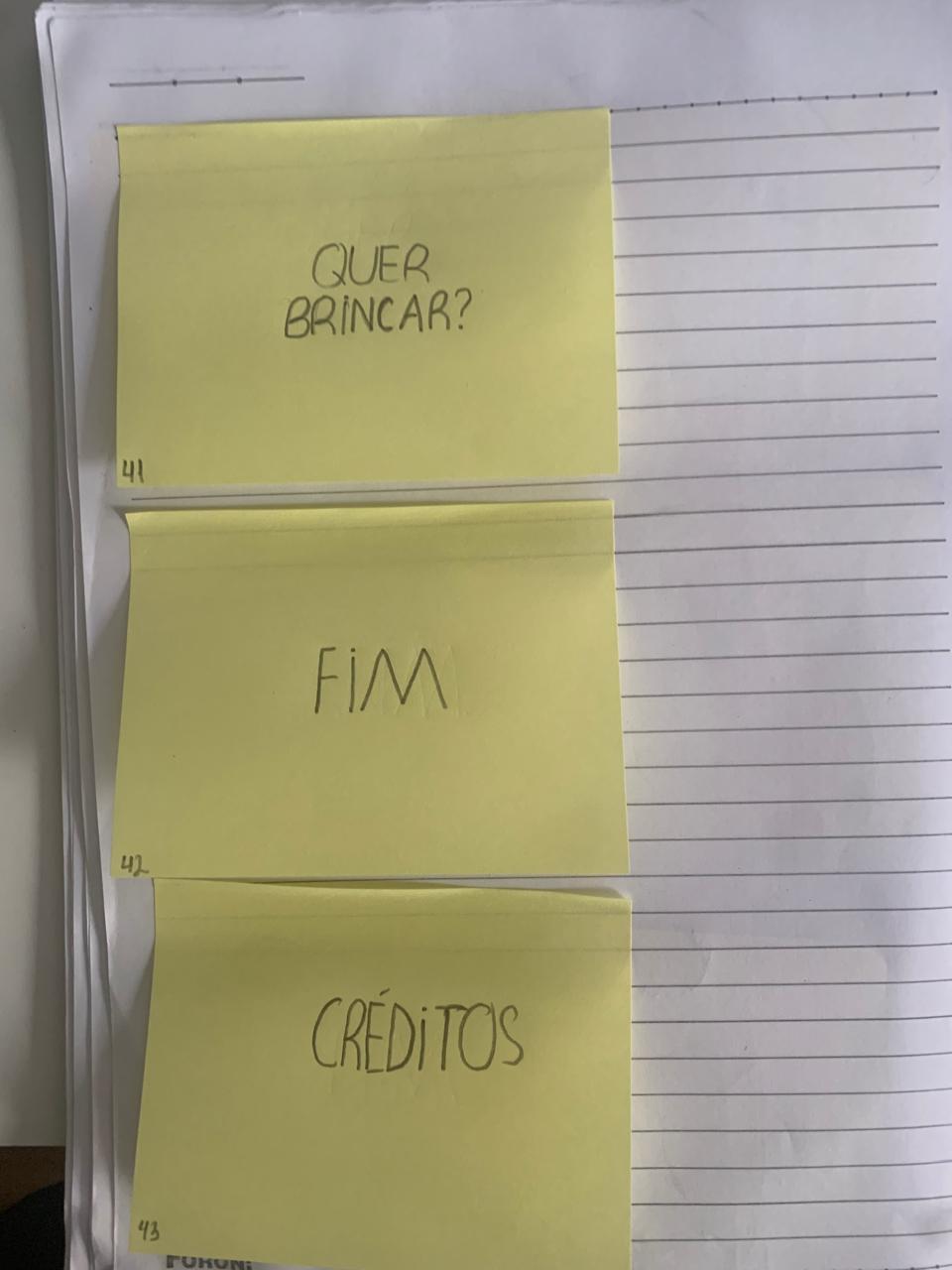
Cena 7 Cena 8

Cena 9 Cena 10

Cena 11 Cena 12

Cena 13 Cena 14