

---

# Final Fantasy VI: Remake

## 1. APRESENTAÇÃO DO JOGO

### 1.1 Descrição

**Final Fantasy VI: Remake** é um jogo, perceptível pelo nome, baseado em *Final Fantasy VI*, um RPG eletrônico de 1994, dirigido por Yoshinori Kitase e Hiroyuki Ito, que revolucionou o mercado de Videogames da época. Ao executar o programa, o jogo inicia uma tela com o modo de **navegação**, que será melhor explicado no seguinte tópico, seu objetivo é chegar na caverna no canto superior direito.

### 1.2 Terra Brandford

Terra é uma garota meio humana, meio Esper (entidade mágica muito poderosa), que foi capturada pelo Império e escravizada para lutar utilizando seus poderes mágicos. O jogo representa o momento em que a menina consegue fugir de sua prisão e busca desesperadamente refúgio em alguma caverna. Uma possível continuação do jogo poderá continuar a história sobre os obstáculos enfrentados pela heroína dentro da caverna.

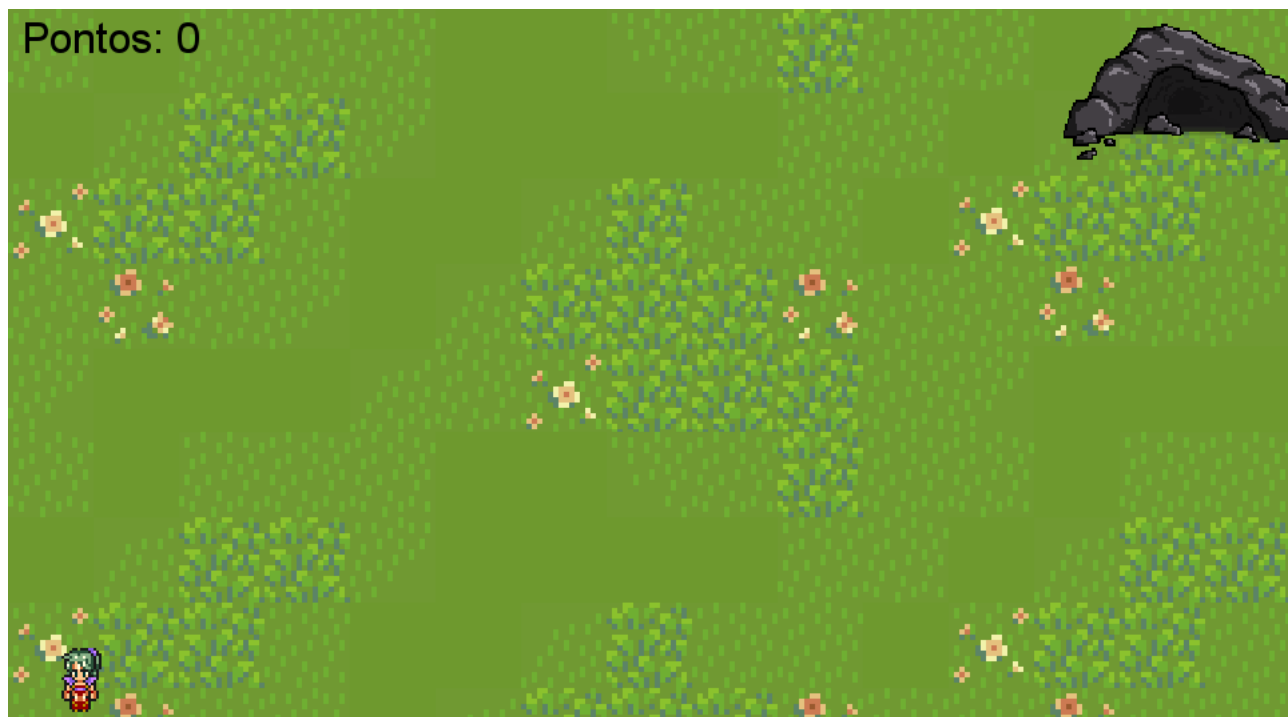


Após anos de treino como escrava no Império, Terra desenvolveu poderosas habilidades de gelo que herdou de seu pai. O lançamento de uma bola de gelo, por exemplo, pode causar um grande estrago em monstros e humanos, além de deixar o alvo lento. Além disso, Terra também possui o poder disparar uma rajada congelante, uma poderosa e fria habilidade que pode congelar inimigos e causar grandes quantidades de dano.

---

### 1.3 Modo de navegação

O modo de **navegação** do jogo permite que o jogador movimente Terra por um cenário de uma grande planície de grama alta pressionando as teclas *W*, *A*, *S* ou *D* para se movimentar para cima, esquerda, baixo ou direita, respectivamente. No canto superior esquerdo é exibida a pontuação do jogador, que pode ser obtida após derrotar monstros. Além disso, no canto superior esquerdo do cenário pode-se notar uma intrigante entrada de caverna, que marca o objetivo do jogador, porém o caminho não será fácil. Por conta da densa vegetação do ambiente, muitos monstros poderosos se escondem nesse lugar, o que certamente irá atrapalhar o caminho da nossa heroína. No total, 20 monstros estão espalhados pelo cenário de maneira aleatória, cada um com um tipo, força e poderes distintos. Ao encontrar um monstro, o jogador é apresentado ao modo de **batalha**, onde será necessário que o jogador utilize a inteligência, juntamente com os poderes da personagem para derrotar os inimigos e sobreviver, antes que o Império a encontre.



Ao chegar à caverna, o jogador vence o jogo e é direcionado para a tela de vitória. Se o jogador obtiver uma pontuação maior do que seu recorde, ou não existirem recordes armazenados, uma mensagem alertará sobre o bom desempenho e essa pontuação será armazenada para jogatinas futuras. O sistema de recordes possibilitará que o jogador tenha vontade de jogar mais vezes para melhorar sua pontuação.

## 1.4 Modo de batalha

No modo de batalha, o monstro finalmente é apresentado para o jogador, o inimigo pode apresentar diferentes tipos, poderes e forças, o que depende somente da sorte (ou azar) da protagonista. Esse modo também apresenta uma visão da planície por um outro ângulo, ademais, é apresentado um menu de opções e uma barra de energia.

O menu de opções permite que o jogador escolha uma ação para realizar em relação ao inimigo encontrado. É possível navegar por esse menu utilizando as setas para cima e para baixo do teclado e escolher a ação pressionando a tecla ENTER. Ao escolher o “Atacar”, Terra lançará uma bola de gelo poderosa que causará um dano constante sempre que for utilizada. Além disso, sempre que usar um ataque, Terra ficará mais poderosa, recebendo um ponto de energia. Ao acumular três pontos de energia, Terra poderá utilizar “Especial”, um ataque muito poderoso que dispara uma rajada congelante que pode causar um grande estrago no inimigo. Por outro lado, se a batalha estiver muito complicada, a personagem pode tentar fugir do monstro, mas a fuga não é garantida. Sempre que a heroína executar um ataque (normal ou especial) e o inimigo não perder todos os pontos de vida, ou falhar na tentativa de fuga, o monstro ataca de volta.



---

## 1.5 Monstros

Existem três tipos diferentes de monstro nas planícies, cada um com diferentes níveis de poder, pontos de vida, e estilos de ataque. Sempre que os pontos de vida do inimigo chegam a zero ou menos, a pontuação é aumentada com base nos pontos de vida do monstro e o jogador retorna para o modo de **navegação**, podendo continuar sua aventura. Porém, se os pontos de vida da heroína acabarem, o jogador perde e vai para a tela de *GAME OVER*.

### 1.5.1 Bandido das areias



Esse monstro é um simples mercenário das areias que abandonou sua terra natal no deserto para se tornar um caçador de recompensas, assassinando friamente diversas pessoas em troca de algumas moedas de ouro. Esse é o inimigo mais fraco encontrado no jogo, tanto em relação a pontos de vida como nível de poder, além de usar equipamentos pesados que deixam seu movimento lento e facilitam a fuga da personagem.

Dica: Ótimo inimigo para conseguir pontos de energia.

### 1.5.2 Dragão Saphira



Esse enorme dragão é extremamente territorialista e ataca tudo que chega perto de sua área. Suas escamas funcionam como uma espessa armadura que o protege dos mais diversos tipos de ataque. Além disso, esse monstro dispara imensas bolas de fogo, que se o alvo não estiver atento, pode causar queimaduras de até quarto grau. O Dragão Saphira não costuma sair de perto do seu território, o que faz com que a chance de fuga do jogador seja mediana.

### 1.5.2 Artema



Artema, também conhecido como “Última Arma”, possui uma origem desconhecida. Esse monstro possui uma reluzente armadura, além de afiadas garras e poderes elétricos. Esse inimigo pode se energizar para lançar grandes projéteis de eletricidade que podem facilmente satisfazer a demanda elétrica de todo um bairro. Ser atingido por um projétil como esse pode causar grandes problemas, e, além disso, seus instintos animais raramente deixam sua presa fugir. Tenha cuidado!

---

### 3. DESCRIÇÃO DO CÓDIGO

**Linhas 1 a 11:** Inclusão das bibliotecas utilizadas no código do jogo.

**Linhas 13 a 36:** Definição de constantes e variáveis globais utilizadas no jogo.

**Linhas 37 a 45:** Funções gerais do jogo.

***random:*** Gera um número aleatório entre 0 e a constante  $n-1$ .

***randInt:*** Gera um número aleatório de um número mínimo até um número máximo, incluindo número negativos.

#### 3.1 Estruturas

**Linhas 47 a 56:** Estrutura do personagem controlado pelo jogador (Terra), com os campos da posição cartesiana de  $x$  e  $y$ , a quantidade de pontos de vida, o dano de ataque, a variável que guarda qual ação está selecionada durante o modo de batalha (*ATACAR*, *ESPECIAL* ou *FUGIR*), uma variável que guarda se o jogador pressionou a tecla ENTER para executar uma ação, e, por fim, dois campos que guardam a posição antiga do  $x$  e do  $y$  do jogador antes de encontrar um monstro.

**Linhas 57 a 64:** Estrutura dos monstros, com os campos  $x$  e  $y$  da posição cartesiana do monstro no modo de **navegação**, seus pontos de vida atuais, o máximo de vida que já teve, o dano de ataque, e, por fim, o tipo do monstro, que pode variar entre *BANDIDO*, *DRAGAO* ou *CACHORRO*.

**Linhas 65 a 70:** Estrutura do projétil (tiro), com os campos  $x$  e  $y$  da sua posição cartesiana no modo de **batalha**, uma variável que armazena se o projétil foi ativado e um campo com a velocidade do tiro.

#### 3.2 Inicializações

**Linhas 72 a 86:** *initGlobais:* Função que inicializa a altura e largura do menu de opções no modo **batalha**, o tamanho da fonte e carrega o arquivo *arial.ttf* da fonte Arial e define que será usado o tamanho 32.

**Linhas 87 a 96:** *initTerra:* Função que inicializa as variáveis da estrutura da protagonista.

**Linhas 98 a 103:** *initProjatilMonstro:* Função que inicializa as variáveis da estrutura de projétil de um monstro.

**Linhas 98 a 103:** *initProjatil:* Função que, de forma similar a função *initProjatilMonstro*, inicializa as variáveis da estrutura de projétil do jogador.

---

### 3.3 Processos do Projétil do Monstro

**Linhas 113 a 168:** *desenhaProjatilMonstro*: Função que desenha o projétil do monstro, somente se o projétil estiver ativo. Toda vez que a função é ativa, a variável de animação aumenta dois frames. Além disso, se o monstro encontrado for um *BANDIDO*, a função desenha um ataque de espada diferente com base no valor da variável de animação. Se o monstro encontrado for um *CACHORRO* ou um *DRAGAO*, a função fará um processo semelhante, porém, com outras imagens para a animação.

**Linhas 169 a 172:** *atiraProjatilMonstro*: Função que ativa o projétil do monstro (atira).

**Linhas 173 a 188:** *atualizaProjatilMonstro*: Função que, se o projétil estiver ativo, muda a posição x do projétil em relação a sua velocidade. Se o projétil chegar até o personagem do jogador, o projétil é desativado, sua posição é redefinida para a inicial, os pontos de vida do jogador são subtraídos pelo dano de ataque do monstro e a variável de animação volta para 0. Se os pontos de vida do jogador acabarem, a função retorna 1, caso contrário, retorna 0.

### 3.4 Processos do Projétil do Jogador

**Linhas 113 a 168:** *desenhaProjatil*: Função que, de forma similar a função *desenhaProjatilMonstro*, desenha o projétil do jogador, somente se o projétil estiver ativo. Se o jogador estiver executando um ataque *ESPECIAL*, a função desenha um ataque com diferentes imagens, com base no valor da variável de animação. Se o jogador estiver executando *ATACAR*, a função fará um processo semelhante, porém, com outras imagens para a animação.

**Linhas 241 a 244:** *atiraProjatil*: Função que, de forma similar a função *atiraProjatilMonstro*, ativa o projétil do jogador (atira).

**Linhas 246 a 262:** *atualizaProjatil*: Função que, de forma similar a função *atualizaProjatilMonstro*, se o projétil estiver ativo, muda a posição x do projétil em relação a sua velocidade. Se o projétil chegar até o monstro, o projétil é desativado, sua posição é redefinida para a inicial e os pontos de vida do monstro são subtraídos pelo dano de ataque do jogador. Se os pontos de vida do monstro acabarem, a pontuação aumenta com base nos pontos de vida máximos do monstro, e a função retorna *NAVEGACAO*, caso contrário, chama a função *atiraProjatilMonstro* e retorna *BATALHA*.

**Linhas 263 a 281:** *atualizaProjatilEspecial*: Função que, de forma similar a função *atualizaProjatil*, se o projétil estiver ativo, muda a posição x do projétil em relação a sua velocidade. Se o projétil chegar até o monstro, o projétil é desativado, sua posição é redefinida para a inicial e os pontos de vida do monstro são subtraídos pelo dano de ataque do jogador **multiplicado por 3**. Se os pontos de vida do monstro acabarem, a pontuação aumenta com base nos pontos de vida

---

máximos do monstro, e a função retorna *NAVEGACAO*, caso contrário, chama a função *atiraProjetoMonstro* e retorna *BATALHA*.

### 3.5 Criação dos Monstros

**Linhas 284 a 287:** *dist*: Função que retorna a distância entre dois pontos no plano cartesiano.

**Linhas 288 a 293:** *colisaoCoordenadas*: Função que retorna 1 se a distância entre as coordenadas de dois monstros for menor ou igual a 100 e, caso contrário, retorna 0.

**Linhas 294 a 302:** *colisaoMonstros*: Função que retorna 1 se existe colisão entre as coordenadas de todos os monstros já criados e o monstro que está sendo criado e retorna 0 caso contrário.

**Linhas 303 a 322:** *criaMonstro*: Função que cria um monstro com um tipo e uma posição aleatória. Dependendo do tipo do monstro são criados diferentes valores para os pontos de vida e dano de ataque.

### 3.6 Modo de Navegação

**Linhas 327 a 345:** *desenhaCenarioNaveg*: Função que desenha uma imagem como o fundo do cenário, uma caverna no canto superior esquerdo da tela e escreve a pontuação atual do jogador.

**Linhas 346 a 356:** *desenhaTerraNaveg*: Função que desenha a imagem do personagem do jogador e, se o personagem estiver virado para a direita, a função inverte a imagem.

**Linhas 358 a 381:** *processaTeclaNaveg*: Função que salva o x e o y do jogador antes de serem alterados, e, dependendo da tecla pressionada, W, S, A ou D, o personagem desloca 20 posições para cima, baixo, esquerda ou direita e retorna 1, 2, 3 ou 4.

**Linhas 382 a 389:** *detectouMonstro*: Função que retorna 1 se o jogador colidiu com algum monstro e 0 caso contrário.

**Linhas 390 a 395:** *chegouObjetivo*: Função que retorna 1 se o jogador chegou ao objetivo no canto superior direito ou 0 caso contrário.

### 3.7 Modo de Batalha

**Linhas 399 a 427:** *desenhaCenarioBatalha*: Função que desenha a imagem do fundo do cenário de batalha, cria as opções do modo de batalha, desenha o cursor do menu de opções e escreve a pontuação no canto superior esquerdo.

**Linhas 428 a 434:** *desenhaTerraBatalha*: Função que desenha a imagem do personagem no modo de batalha.

---

**Linhas 435 a 444:** *desenhaVidaTerra*: Função que desenha a barra de vida do jogador, variando da cor verde para amarelo e vermelho, dependendo da quantidade de pontos de vida.

**Linhas 446 a 463:** *desenhaMonstroBatalha*: Função que desenha a imagem do monstro no modo de batalha, dependendo do tipo do monstro a imagem é diferente.

**Linhas 446 a 463:** *desenhaMonstroBatalha*: Função que desenha a imagem do monstro no modo de batalha, dependendo do tipo do monstro a imagem é diferente.

**Linhas 464 a 473:** *desenhaVidaMonstro*: Função que desenha a barra de vida do monstro, variando da cor verde para amarelo e vermelho, dependendo da quantidade de pontos de vida.

**Linhas 474 a 488:** *desenhaManaBar*: Função que desenha a barra de mana do jogador e a quantidade de mana atual.

**Linhas 489 a 533:** *processaTeclaBatalha*: Função que, se não houver nenhum projétil ativo, permite o jogador navegar pelo menu de opções utilizando as setas do teclado e selecionar a ação pressionando ENTER.

**Linhas 534 a 594:** *processaAcaoTerra*: Função que, se o jogador tiver executado uma ação, desativa a execução (para a função não ser executada mais de uma vez). Além disso, se a ação for *ATACAR*, a função *atiraProjetoil* é chamada, e a mana é aumentada em um. Em caso semelhante, se a ação for *ESPECIAL*, a função *atiraProjetoil* é chamada e a mana é zerada. Nesses dois casos, se a vida do inimigo for menor ou igual a zero, a função retorna *NAVEGACAO*. Se a ação do jogador for *FUGIR*, a função *randInt* será chamada, e, para cada tipo de monstro, um valor mínimo é requisitado. Se o valor for maior do que o necessário para fugir, a função retorna *NAVEGACAO*, caso contrário a função *atiraProjetoilMonstro* é chamada.

## **3.8 Função Main**

### **3.8.1 Rotinas de inicialização**

**Linhas 598 a 600:** Cria a tela, a fila de eventos e o temporizador do jogo.

**Linhas 604 a 608:** Inicializa a biblioteca Allegro.

**Linhas 610 a 614:** Inicializa o módulo de primitivas do Allegro.

**Linhas 616 a 620:** Inicializa o módulo que permite carregar imagens no jogo.

**Linhas 622 a 627:** Cria um temporizador que incrementa uma unidade a cada frame por segundo.

**Linhas 629 a 635:** Cria uma tela com dimensões de 960 x 540.



---

**Linhas 637 a 641:** Instala o teclado.

**Linhas 643 a 647:** Instala o áudio.

**Linhas 649 a 656:** Inicializa os módulos Allegro que permite carregar fontes, imagens, áudios e reserva 4 espaços para áudios.

**Linhas 658 a 662:** Inicializa o módulo Allegro que entende arquivos ttf de fontes de texto.

**Linhas 664 a 677:** Cria a fila de eventos do jogo e registra na fila eventos de tela, eventos de tempo e eventos de teclado.

**Linha 680:** Inicia o temporizador.

### 3.8.2 Rotinas Específicas do Jogo

**Linha 683:** Variável que guarda o modo de jogo (NAVEGACAO ou BATALHA).

**Linhas 684 a 689:** Criação das estruturas.

**Linha 691:** Geração de seed aleatória para as funções rand.

**Linhas 693 a 702:** Criação dos vinte monstros juntamente com a verificação se existe colisão entre eles, utilizando as funções *criaMonstro* e *colisaoMonstros*. Se houver colisão, o monstro é criado novamente.

**Linhas 693 a 702:** Criação dos vinte monstros juntamente com a verificação se existe colisão entre eles, utilizando as funções *criaMonstro* e *colisaoMonstros*. Se houver colisão, o monstro é criado novamente.

**Linhas 704 a 716:** Criação de algumas variáveis importantes. A identidade do monstro encontrado, a direção que o personagem está olhando, a quantidade de energia, frames de animação, auxiliares no desenho da barra de vida, a pontuação, o recorde, e variáveis que armazenam se o jogo está sendo executado ou não e se o jogador venceu.

**Linhas 718 a 722:** Abertura do arquivo que armazena o recorde e armazenamento do recorde.

**Linhas 724 a 780:** Carrega todas as imagens e sprites utilizados no jogo.

**Linhas 782 a 786:** Carrega todas as músicas utilizadas no jogo.

**Linhas 788 a 793:** Inicialização das estruturas.

**Linha 796:** Inicia a música do modo de **navegação**.

---

### 3.8.2 Execução do Jogo

**Linha 799:** Início do while do jogo.

**Linhas 801 a 806:** Toda vez que o tempo do temporizador aumenta (**eventos do temporizador**), as seguintes funções são chamadas.

**Linhas 808 a 811:** Muda o frame de animação de sprites.

**Linhas 815 a 824:** Se o jogador está no modo de **navegação**, as funções para desenhar o cenário são chamadas (*desenhaCenarioNaveg* e *desenhaTerraNaveg*).

**Linhas 826 a 839:** Se o jogador está no modo de **navegação**, verifica se o jogador encontrou algum monstro através da função *detectouMonstro* e, caso retorno verdadeiro, as variáveis para identidade do monstro, e desenho de sua barra de vida são salvas, a música do modo de **navegação** é interrompida e a música do modo de **batalha** começa. Por fim, o modo de jogo muda para **batalha**.

**Linhas 841 a 845:** Se o jogador está no modo de **navegação**, a função *chegouObjetivo* é chamada e, caso retorno verdadeiro, o jogo acaba e o jogador vence.

**Linhas 841 a 845:** Se o jogador está no modo de **batalha**, as seguintes funções são chamadas: *desenhaCenarioBatalha*, *desenhaTerraBatalha*, *desenhaMonstroBatalha*, *desenhaVidaTerra*, *desenhaVidaMonstro*, *desenhaManaBar*, *desenhaProjtil* e *desenhaProjtilMonstro*. Além disso, caso as funções *processaAcaoTerra*, *atualizaProjtil* ou *atualizaProjtilEspecial*, retornem *NAVEGACAO*, ou seja, que o jogador fugiu ou derrotou o inimigo, o modo de jogo volta para a **navegação**. Além disso, caso *atualizaProjtilMonstro* seja verdadeiro, ou seja, que a vida do jogador chegou a 0, o jogo acaba e o jogador perde.

**Linha 881:** Atualiza a tela quando houver algo para mostrar.

**Linha 882:** Fim dos **eventos do temporizador**.

**Linhas 883 a 887:** Se o tipo de evento for o fechamento da janela, o jogo para de ser executado.

**Linhas 883 a 887:** Se o tipo de evento for um pressionar de uma tecla, se o modo de jogo for **navegação**, a função *processaTeclaNaveg* é chamada e a direção do personagem é armazenada. Caso o modo de jogo seja **batalha**, a função *processaTeclaBatalha* é chamada.

**Linha 896:** Fim do while do jogo.

### 3.8.3 Tela de Fim de Jogo

**Linhas 898 e 899:** A música é parada e a variável para iniciar uma música é criada.

---

**Linha 900:** Início do while da tela de fim de jogo.

**Linhas 907 a 926:** Se o jogador venceu, a música de vitória inicia e um texto comemorativo é escrito. Além disso, se a pontuação foi maior do que o recorde, outra mensagem comemorativa e o novo recorde aparecem. Caso contrário, se a pontuação não for maior que o recorde, a pontuação é escrita e abaixo o recorde também aparece.

**Linhas 927 a 932:** Se o jogador perdeu, a música de derrota inicia e o texto “GAME OVER” é escrito na tela.

**Linha 935:** Atualiza a tela, quando houver algo para mostrar.

**Linhas 936 a 939:** Se o tipo de evento for o fechamento da tela, a tela de fim de jogo acaba.

**Linha 940:** Fim do while da tela de fim de jogo.

### **3.8.3 Procedimento de Destruição**

**Linhas 942 a 1005:** Finaliza o temporizador, fecha a tela, limpa a memória, destrói todos os bitmaps, sprites, imagens e músicas do jogo.

**Linha 1008:** Fim do main.

## **4. AUTOR**

Carlos Henrique Brito Malta Leão

Sistemas de Informação - UFMG

chbmleao@ufmg.br

PDS1

### **4.1 Agradecimentos**

Pedro Olmo Vaz de Melo

Auxílio no desenvolvimento de sprites: Leandro Guido e João Pedro Pimenta

Sprites Terra, monstros e cenários: *Final Fantasy VI* (1994)

Arquivos do jogo: [drive.google.com/drive/folders/1nKv9y6mx1b\\_X4bfN0lbGVsEhdsQjVHOC?usp=sharing](https://drive.google.com/drive/folders/1nKv9y6mx1b_X4bfN0lbGVsEhdsQjVHOC?usp=sharing)