DOCUMENTAÇÃO DA BIBLIOTECA

THINKLIB

Biblioteca de Mecânicas de Jogos para Unity.

THINKLIB

Think TEd Lab

UEA UNIVERSIDADE DO ESTADO DO AMAZONAS

Sumário

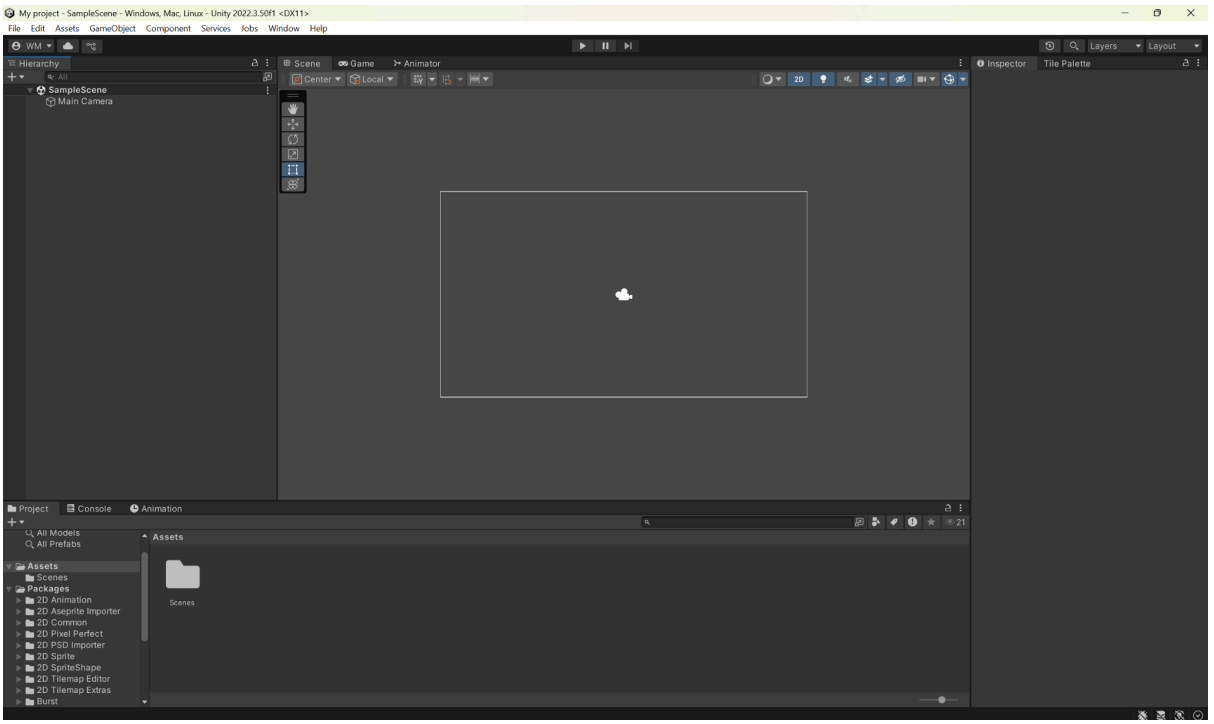
Introdução

ThinkLib é uma biblioteca projetada para simplificar e padronizar o desenvolvimento de mecânicas de jogos na plataforma Unity. Criada por Izaque Rolim, Ícaro Benarrós e Waldecir Martins, estudantes de Licenciatura em Computação, sob a orientação da professora doutora Fernanda Pires, ThinkLib oferece funcionalidades reutilizáveis que promovem boas práticas de programação, garantindo eficiência e organização. Seu objetivo é facilitar a implementação de mecânicas essenciais, para categorias de jogos como plataforma, top-down e point-and-click, contribuindo para o desenvolvimento de jogos estruturados e escaláveis.

Guia de Instalação

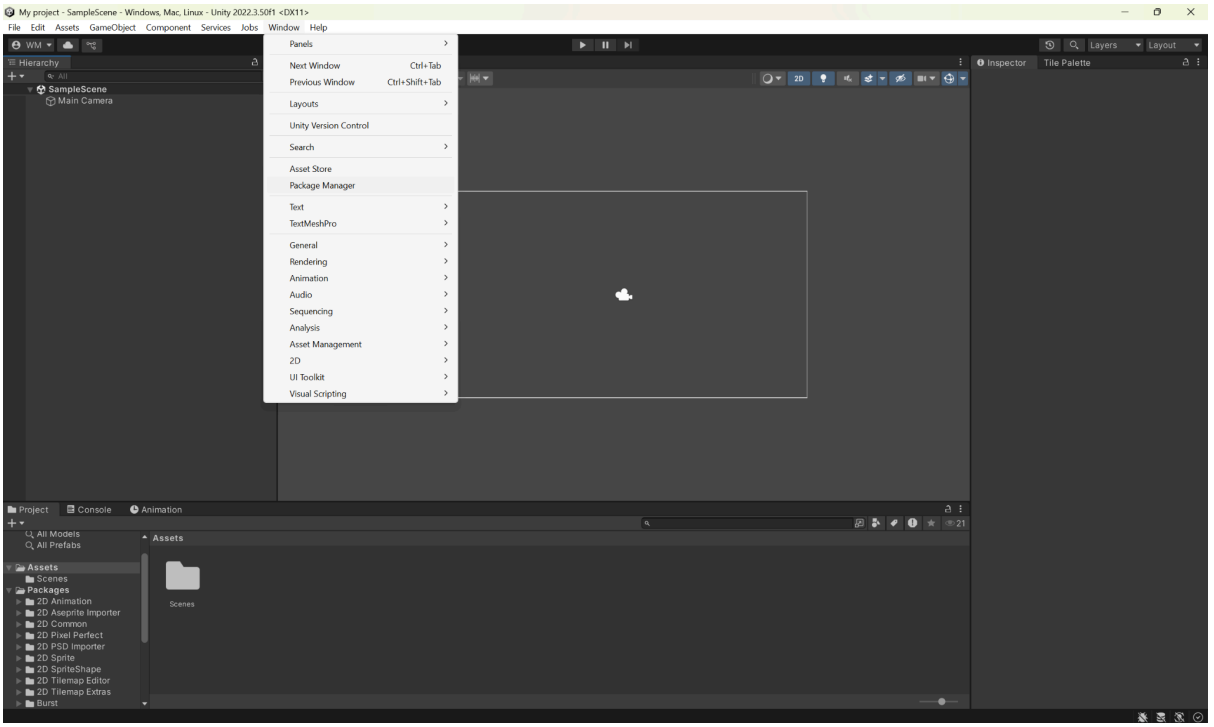
Para integrar ThinkLib ao seu projeto Unity, siga os passos abaixo:

1. Abra o Unity e carregue o projeto



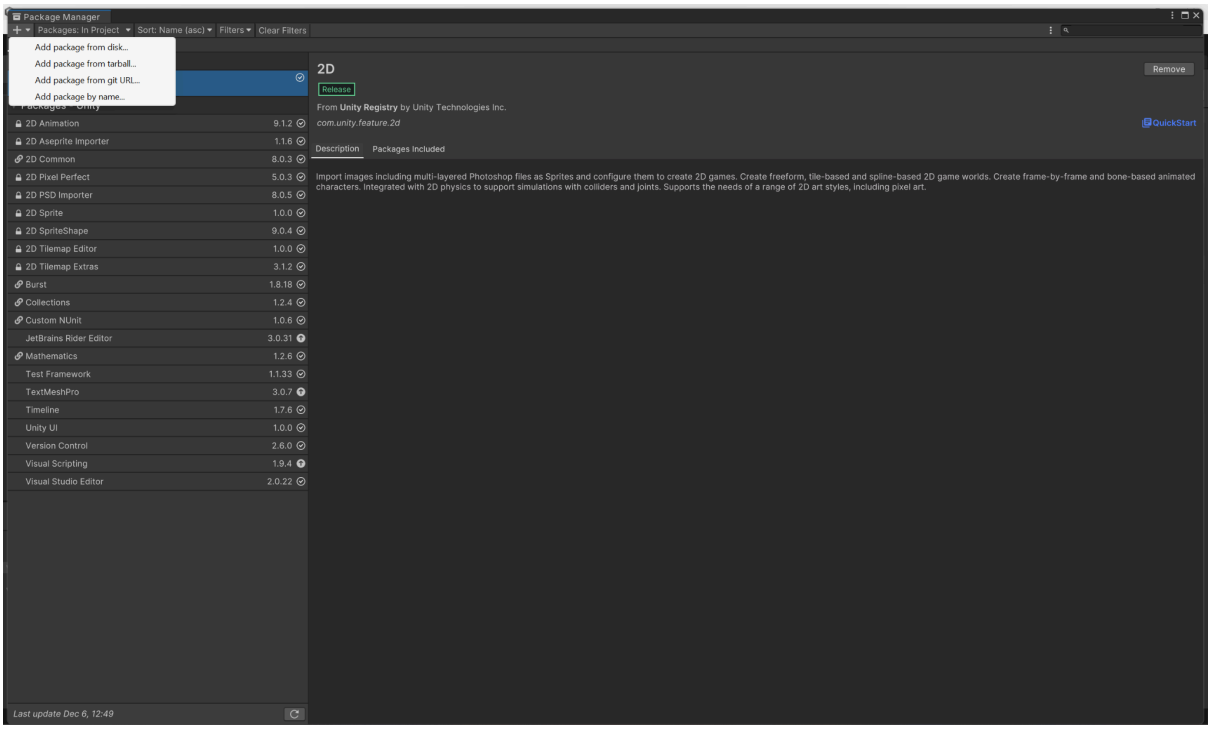
Certifique-se de que o projeto no qual deseja utilizar ThinkLib está configurado e aberto no Unity Editor.

2. Acesse o Package Manager



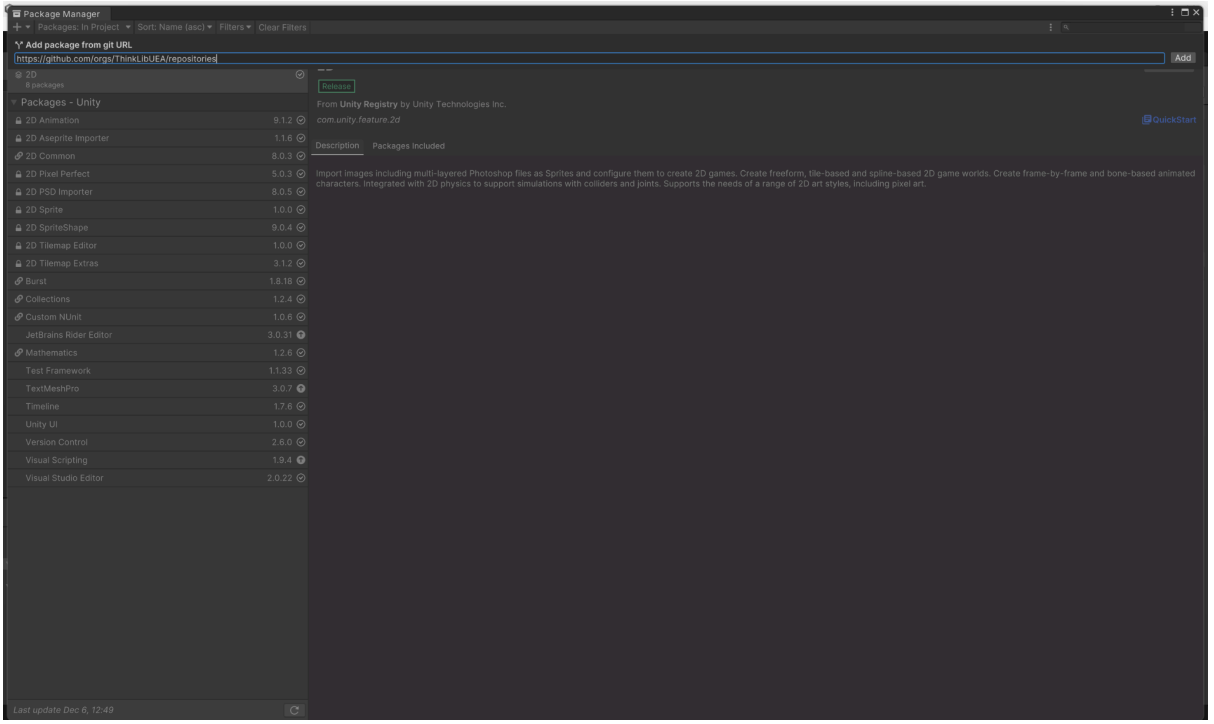
No menu superior do Unity Editor, vá em Window > Package Manager. Isso abrirá a interface de gerenciamento de pacotes.

3. Adicione um pacote por URL do Git



No canto superior esquerdo do Package Manager, clique no botão + e selecione a opção Add package from Git URL.

4. Insira o endereço do repositório



a. Digite o seguinte URL no campo fornecido:
b. https://github.com/orgs/ThinkLibUEA/repositories
c. Em seguida, pressione Enter ou clique em Add.

5. Confirme a instalação

O Unity realizará o download da biblioteca e a adicionará ao seu projeto. Após a conclusão, ThinkLib estará disponível para uso.

Caso encontre problemas durante a instalação, verifique se:

- O Unity está conectado à internet.
- O link do repositório está correto.
- A versão do Unity é compatível com ThinkLib.

Jogos de Plataforma - Movimentação

Métodos

Método	Parâmetros	Descrição
SetTeclasMovimentacao	List<KeyCode> listaTeclasDireita, List<KeyCode> listaTeclasEsquerda	Configura as listas de teclas que controlam o movimento do personagem para a direita e para a esquerda
SetAnimacaoParado	Animation animacao	Define o clipe de animação exibido enquanto o personagem está em movimento
SetAnimacaoMovimento		
SetVelocidade	Float velocidade	Especifica a velocidade padrão de movimentação do personagem
SetCorrer	KeyCode teclaCorrer float velocidade, Animation animacao	Define a tecla que o personagem vai correr, a velocidade da corrida e a animação da corrida.
SetDash	KeyCode teclaDash, Animation animacao	Define a tecla responsável por executar o dash e a animação correspondente ao momento do dash,

Exemplo Prático - Configurando Movimentação

```
using UnityEngine;
using System.Collections.Generic;

public class KeyCodeExample : MonoBehaviour
{
    public AnimationClip animacaoPersonagem, animacaoCorrer, animacaoDash;
    public Movimentoacao movimentacao;

    void Start()
    {
        List<KeyCode> teclasDireita = new List<KeyCode>
        {
            KeyCode.D,
            KeyCode.RightArrow
        };

        List<KeyCode> teclasEsquerda = new List<KeyCode>
        {
            KeyCode.A,
            KeyCode.LeftArrow
        };

        movimentacao.SetTeclasMovimentacao(teclasDireita, teclasEsquerda);
        movimentacao.SetAnimacao(animacaoPersonagem);
        movimentacao.SetVelocidade(4f);
        movimentacao.SetCorrer(KeyCode.LeftShift, 10f, animacaoCorrer); // Opcional
        movimentacao.SetDash(KeyCode.Q, animacaoDash); // Opcional
    }
}
```

Jogos de Plataforma - Pulo

Métodos

Método	Parâmetros	Descrição
SetTeclaPulo	KeyCode teclaPulo	Define a tecla que o personagem irá efetuar o pulo.
SetIsPuloDuplo	Boolean isPuloDuplo	Ativa ou desativa a funcionalidade de pulo duplo. Se habilitado, o personagem poderá realizar um segundo pulo enquanto ainda estiver no ar.
SetForcaPulo	Float forcaPulo	Define a força do pulo do personagem. Valores maiores resultam em pulos mais altos.
SetTagChao	String tag	Define a tecla do objeto e é considerado o “chão”, é importante para o bom funcionamento do pulo duplo.

Exemplo Prático - Configurando Pulo

```
using UnityEngine;

public class KeyCodeExample : MonoBehaviour
{
    public Pulo pulo;

    void Start()
    {
        pulo.SetTeclaPulo(KeyCode.Space);
        pulo.SetIsPuloDuplo(true);
        pulo.SetForcaPulo(100f);
        pulo.SetTagChao("ground");
    }
}
```

Jogos de Plataforma - Ataque com Projétil

Métodos

Método	Parâmetros	Descrição
SetProjetil	GameObject objetoDisparado	Define o objeto do jogo que será utilizado como projétil.
SetTeclaDisparo	KeyCode teclaDisparo	Define a tecla que ativará o disparo do projétil.
SetPosicaoInicialProjetil	Transform posInicial	Configura a posição inicial do projétil na cena. Essa posição geralmente corresponde ao ponto de origem, como a mão ou arma do personagem
SetVelocidadeProjetil	Float velocidadeProjetil	Define a velocidade de movimento do projétil.
SetTempoMaximoProjetil	Float tempoMaximoProjetil	Determina o tempo máximo, em segundos, que o projétil permanecerá ativo na cena.

		O valor padrão é 10 segundos, se não ajustado.
SetAnimacaoProjetil	Animation animacao	Atribui um clipe de animação específico ao projétil
SetAnimacaoPersonagem	Animation animacaoPersonagem	Define o clipe de animação aplicado ao personagem no momento do disparo.

Exemplo Prático - Configurando Ataque

```
using UnityEngine;

public class KeyCodeExample : MonoBehaviour
{
    public AnimationClip animacaoProjetil;
    public AnimationClip animacaoPersonagem;
    public AtaqueDisparo ataque;
    public GameObject projetil;
    public Transform posicaoInicialProjetil;

    void Start()
    {
        ataque.SetProjetil(projetil);
        ataque.SetTeclaDisparo(KeyCode.Mouse0);
        ataque.SetPosicaoInicialProjetil(posicaoInicialProjetil);
        ataque.SetVelocidadeProjetil(5f);
        ataque.SetTempoMaximoProjetil(30f);
        ataque.SetAnimacaoProjetil(animacaoProjetil);
        ataque.SetAnimacaoPersonagem(animacaoPersonagem);
    }
}
```

Jogos de Plataforma - Ataque Corpo a Corpo

Métodos

Método	Parâmetros	Descrição
SetTeclaAtaque	KeyCode teclaAtaque	Define a tecla de ataque do personagem.
SetAnimacaoAtaque	Animation animation	Define o clipe de animação aplicado ao personagem no momento de ataque.
SetRaioAtaque	Float raioAtaque	Estabelece o alcance do ataque do personagem. Um raio maior permite atingir inimigos mais distantes.
SetTagInimigos	String tag	Especifica a tag dos objetos que sofrerão dano com o ataque.

Exemplo Prático - Configurando Ataque

```
using UnityEngine;
using UnityEngine.UI;

public class KeyCodeExample : MonoBehaviour
{
    public Text textoMoeda;
    public AnimationClip animacaoAtaque;
    public AtaqueDeCorpo ataque;

    void Start()
    {
        ataque.SetTeclaAtaque(KeyCode.Mouse0);
        ataque.SetAnimacaoAtaque(animacaoAtaque);
        ataque.SetRaioAtaque(10f);
        ataque.SetTagInimigos("inimigo");
    }
}
```

Jogos de Plataforma - Sistema de Vida

Métodos

Método	Parâmetros	Descrição
SetBarraDeVida	Image barraDeVida	Define a imagem que irá ser redimensionada conforme a quantidade de vida.
SetVidaMaxima	int vidaMaxima	Defina a quantidade máxima de vida do personagem.
PerderVida	int quantidadeDeVidaPerdida	Reduz a vida atual do personagem com base na quantidade enviada por parâmetro.
GanharVida	int quantidadeDeVidaRestaurada	Aumenta a vida atual do personagem com base na

		quantidade enviada por parâmetro.
SetDerrota	String funcaoDeDerrota	Define o nome do método que será chamado quando a vida do personagem for menor ou igual a zero, pode executar ações como Game Over, etc.

Atributos

Nome	Descrição
GetVidaAtual()	Retorna um inteiro com a quantidade de vida atual.

Exemplo Prático - Configurando Sistema

```
using UnityEngine;
using UnityEngine.UI;

public class KeyCodeExample : MonoBehaviour
{
    public Image barraDeVida;
    public Text textoVida;
    public SistemaDeVidaLib sistema;

    void Start()
    {
        sistema.SetBarraDeVida(barraDeVida);
        sistema.SetVidaMaxima(100f);
        sistema.SetDerrota("ExibirMensagemDerrota");
    }

    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space))
        {
            sistema.PerderVida(10f);
        }

        if (Input.GetKeyDown(KeyCode.LeftShift))
        {
            sistema.GanharVida(10f);
        }

        textoVida.text = sistema.GetVidaAtual().ToString(); // Atualiza o texto com a quantidade de
vida
    }

    void ExibirMensagemDerrota()
    {
        Debug.Log("Você perdeu.");
    }
}
```

Jogos de Plataforma - Sistema de Coleta

Métodos

Método	Parâmetro	Descrição
SetTagObjetoColetavel	String tag	Insira a tag do objeto do jogo que será coletado. Defina essa tag no Inspector
SetFuncaoColeta	String nomeDaFuncao	Insira o nome do método que irá ser chamado ao coletar o item.

Atributos

Property	Function
QtdItemColetados	Retorna um inteiro com a quantidade de itens coletados

Exemplo Prático - Configurando Sistema

```
using UnityEngine;
using UnityEngine.UI;

public class KeyCodeExample : MonoBehaviour
{
    public Text textoMoeda;
    public SistemaDeColeta coleta;

    void Start()
    {
        coleta.SetTagObjetoColetavel("moeda");
        coleta.SetFuncaoColeta("ExibirMensagemDeColeta");
    }

    void Update()
    {
        textoMoeda.text = coleta.QtdItensColetados().ToString(); // Atualiza o texto com a quantidade
de moedas
    }

    void ExibirMensagemDeColeta()
    {
        Debug.Log("Você coletou o item tal.");
    }
}
```