Zed

Owned by guilherme zicari and Created By guilherme zicari - version: 0.0

Powered by Dundoc

1. Game Introduction

Project Zed se trata de um Adventure Action Game com uma historia linear que o jogador vai experienciar em multiplos Atos, o jogo oferece um mundo 2D que se passa era medieval fictícia para explorar e dungeons que proporcionarão desafios ao jogador.

Adventure Action Game

Um Adventure Action Game é um hibrido entre os 2 tipos de jogos que compões seu nome , dentre as suas principais características estão exploração , combate dinâmico e puzzles.

GamePlay

O jogador será capaz de acompanhar uma historia envolvente enquanto desfruta de mecânicas de combate simples e dungeons criadas para apresentar um desafio ao jogador.

Target audience & platforms

O jogo é recomendado para qualquer publico alvo mas terá como plataforma apenas PC

Look and Feel

- ## Story

  Em uma pequena vila nos arredores do castelo de Zurik, vivia Lank e seu Avô Zenk, durante anos ambos Lank e seu avô viveram em paz e tranquilidade como fazendeiros provendo para o Reino de Anillia mas um dia Zenk desapareceu, e agora cabe a Lank descobrir o que aconteceu com seu avô.

- ## How to Play

  O jogo apresenta mecânicas simples , possuindo o botão "Espaço" como o botão de ataque e interação, o jogador deverá procurar por pistas que o levem ao seu avô e no caminho utilizar uma espada para lidar com inimigos que queiram impedi-lo.

---

2. # Technical

- ## System Requirements

  Requisitos Mínimos:

  OS: Windows Vista ou melhor.

  Processador: 2 Ghz

  Memória: 2gb de RAM.

  Placa de vídeo: 256mb de memória gráfica.

  DirectX : Versão 10 ou superior.

  Armazenamento: 200mb ou superior

- ## Game Architechture

  O Primeiro ato se trata de uma busca pelas pistas que seu Avô possa ter deixado para trás enquanto o jogador devera combater os inimigos em seu caminho.

- ## User Interface (controls)

Espaço : Tecla responsável tanto pela interação entre jogador e objetos no cenário quanto tecla usada para atacar.

---

3. # Artwork (design)

Os Sprites usados no jogo foram adquiridos em https://opengameart.org/ , todos possuem licensa **CC0,**

- ## Heads Up Display (HUD)

- ## Characters

  - ### Lank

    Lank, o protagonista , um jovem destemido e sagaz que busca encontrar seu avô.

Lank Cresceu em uma casa nos arredores do castelo de Zurik como um fazendeiro, seus pais morreram quando Lank ainda era pequeno demais para se lembrar e não se sabe de nenhuma outro parente além de seu avô.

Em seu tempo livre Lank brincava escondido com uma velha espada que seu avô mantinha guardada no armário,uma velha lembrança de sua juventude.

Lank possui 13 diferentes animações utilizando 43 Sprites no total.

- Zenk

Avô de Lank , um ancião que já passou de seu auge, seus olhos mostram sabedoria que somente o tempo presenteia, mas ainda assim não são capazes de esconder a dor que carrega consigo.

Após a morte de sua filha e genro criou Lank sozinho e o considera seu maior tesouro.

Inimigos

- Loggy

Especie de Ent Nativa do Reino de Anallia, normalmente permanecem em sono perpetuo até que suas raízes sintam movimento na proximidade.

Sua dieta consiste de nutrientes do solo, mas não se importam de complementa-la com um ou dois aventureiros descuidados.

- ## Batty

Espécie comum de Morcegos comum ao Reino de Anallia, se alimentam de insetos e sangue.

Comumente encontrados em cavernas a sul do Castelo.

- ## Skelly

Skellys são criaturas ilógicas que não possuem carne ou sangue, compostas meramente de ossos não se sabe como estas vem a existir ou mesmo se possuem senciência.

Algum dizem que a morte é inevitável, estas criaturas entretanto mostram que uma alternativa indesejável é possível.

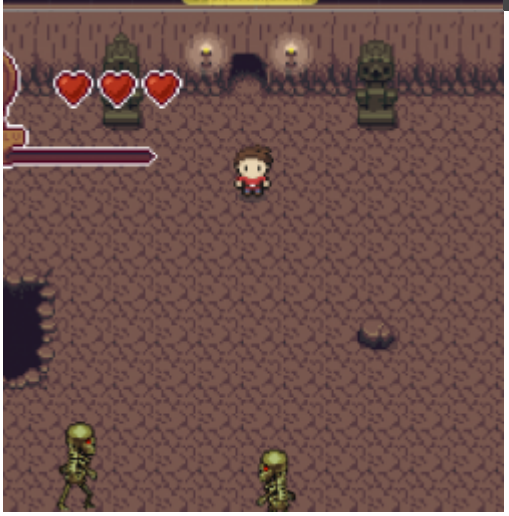
Level Design

Todos os mapas são compostos por quadrados de 25x25 pixels, existem 3 principais ambientações para os mapas:

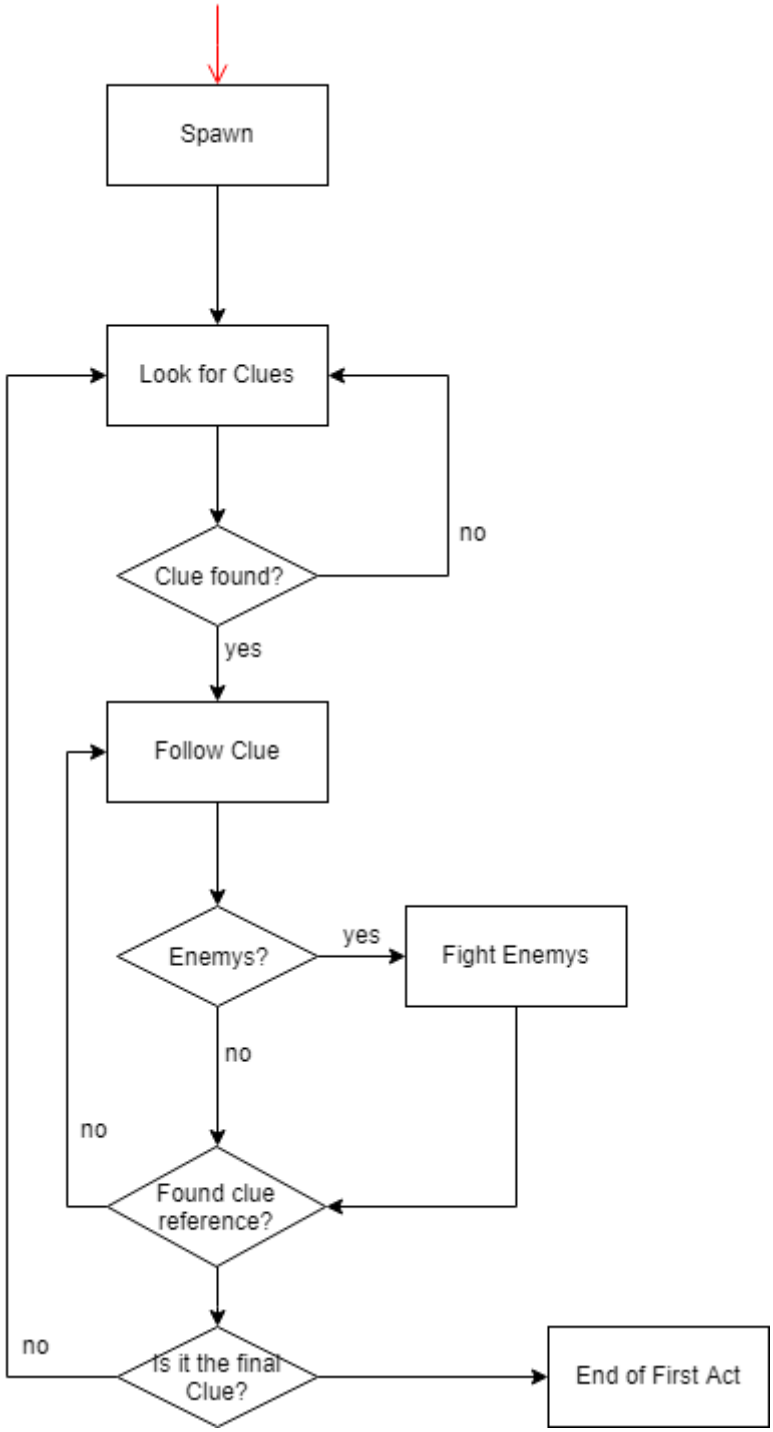
-Ambientes internos , como o interior de uma casa

-A Dungeon

-A floresta em que Lank vive e seus arredores, incluindo os diversos lagos e rios.



Fluxograma - ATO 1



---

4. Development

◦ Scripts C# e Classes

O jogo feito em Unity utiliza Scripts em linguagem C# e modelagem Orientada a Objetos.

▪ Player_Movement

Script responsável pelas ações do jogador e administrar seu Status constantemente.

Seus métodos incluem :

void Start()

void Update()

private IEnumerator attackCO // Animação Ataque

void UpdateAnimationAndMove() // animação e movimento

void MoveCharacter() // Movimento

public void Knock(float knockTime, float dmg) // Damage

private IEnumerator knockCO(float knockTime) // Knockback

- # Player_Hit

Responsavel pela interação do ataque do Player com objetos que podem se quebrar , como Potes.

Métodos:

public void OnTriggerEnter2D(Collider2D other)

- # Health_Manager

Administra o HP do jogador e a representação deste na HUD

Métodos:

public void InitHearts()

public void UpdateHearts()

```
public FloatValue playerCurrentHealth;

// Start is called before the first frame update
void Start()
{
    InitHearts();
}
public void InitHearts()
{
    for(int i=0; i< heartContainers.initialValue; i++)
    {
        hearts[i].gameObject.SetActive(true);
        hearts[i].sprite = fullHeart;
    }

}
public void UpdateHearts()
{
    float tempHealth = playerCurrentHealth.RuntimeValue / 2;
    for(int i = 0; i< heartContainers.initialValue; i++)
    {
        if(i<= tempHealth - 1)
        {
            hearts[i].sprite = fullHeart;
        }else if (i >= tempHealth)
        {
            hearts[i].sprite = emptyHeart;
        }
        else
```

- # Room_Move

Administra os limites da camera para que eles não mostre rooms além do que o jogador está.Também responsavel por mostrar o nome do room que o jogador entrou.

Métodos:

private void OnTriggerEnter2D(Collider2D other)

private IEnumerator nomeLugarCO()

```
    {
        cam = Camera.main.GetComponent<Camera_Movement>();

    }

    // Update is called once per frame
    void Update()
    {

    }
    private void OnTriggerEnter2D(Collider2D other)
    {
        if (other.CompareTag("Player") && !other.isTrigger)
        {
            buffer = cameraChange;
            cam.minPosition +=  buffer;
            cam.maxPosition +=   buffer;
            other.transform.position += playerChange;
            buffer.x = 0;
            buffer.y = 0;
                if (show)
            {
                StartCoroutine(nomeLugarCO());
            }
        }

    }
```

- # Scene_Transition

Responsavel pela transição de cenas no jogo.

Existem 3 no total:

-Interior da casa

-Interior da Dungeon

-Mundo principal

Métodos:

public IEnumerator FadeCO() // Administra Fadein e Fadeout da tela branca quando mudando de cena.

private void Awake()

private void OnTriggerEnter2D(Collider2D other)

- # Enemy

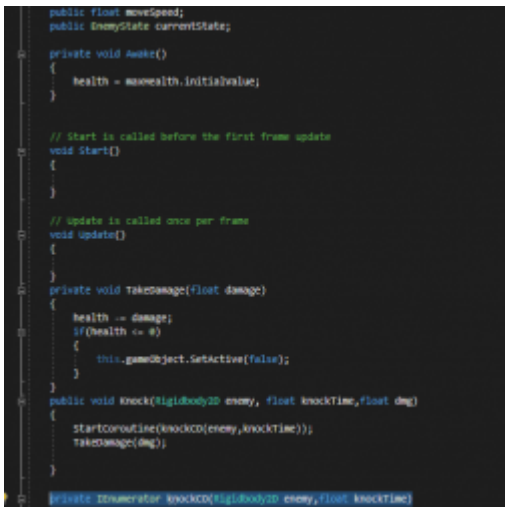
Classe pai de todos os inimigos , inimigos herdaram desta.

Métodos:

public void Knock(Rigidbody2D enemy, float knockTime,float dmg)

private void TakeDamage(float damage)

private IEnumerator knockCO(Rigidbody2D enemy,float knockTime)



- # Loggy

Script do inimigo Loggy , Herda de Enemy:

void checkDistance()

private void SetAnimFloat(Vector2 setter)

private void changeAnim(Vector2 dir)

private void ChangeState(EnemyState newState)

---

5. Audio & Sound F/x

O jogo contêm uma trilha constante para cada Cena:

-Interior da Casa : Windless Slopes.mp3

-OutWorld : Celestial;mp3

-Dungeon: The Arrival (BATTLE II);mp3

E a musica FoggyWoods.mp3 para o menu

O compositor de todas é desconhecido.

---

6. Marketing

Project Zed é um jogo para nostálgicos da era 16 bits , principalmente aqueles que jogaram Zelda A link to the Past e deve ser recebido como uma experiencia rápida que tenta reviver os sentimentos do passado.

◦ Key Features

-Exploração

-Combate In-Real-Time

-Spelunking

-Puzzle
