

Unity

Power Ups



Power Ups

Um "power-up" em videogames é um item ou habilidade especial que concede ao jogador vantagens temporárias. Essas vantagens podem incluir saúde extra, aumento de força, velocidade aumentada ou até mesmo invencibilidade por um curto período de tempo. Os power-ups são fundamentais para superar desafios, derrotar inimigos e progredir no jogo, tornando a experiência mais emocionante e dinâmica. Eles são geralmente representados por ícones ou objetos visuais no jogo, que os jogadores devem coletar para desfrutar de seus benefícios temporários.

Power Ups

Vamos desenvolver e implementar os Power Ups no nosso projeto.

Crie uma pasta **PowerUp** dentro da pasta de scripts.

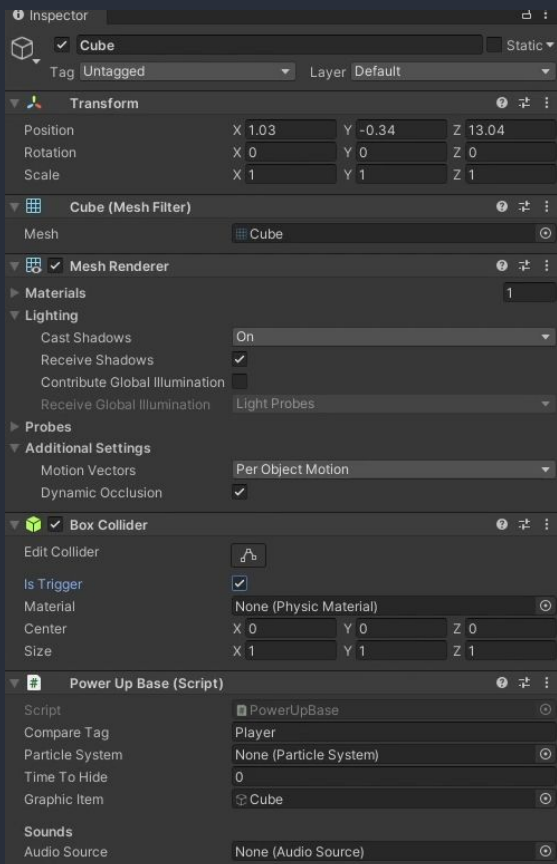
Então crie um script chamado **PowerUpBase**

Power Ups

Precisamos fazê-lo ser coletável agora. Para isso, estenda da classe **ItemCollectableBase**.

```
public class PowerUpBase : ItemCollectableBase
{
}
```

Power Ups



Adicione um cubo na cena, e adicione o PowerUpBase nele. Atualize as referências necessários, mude o colider para ser Trigger.

Power Ups

```
[Header("Power Up")]
public float duration;

protected override void OnCollect()
{
    base.OnCollect();
    StartPowerUp();
}

protected virtual void StartPowerUp()
{
    Debug.Log("Start Power Up");
    Invoke(nameof(EndPowerUp), duration);
}

protected virtual void EndPowerUp()
{
    Debug.Log("End Power Up");
}
```

Power Ups

Remova os componentes de mesh do cubo, e adicione dentro do objeto um outro cubo. Faremos isso pra deixar o visual separado do componente de lógica. Dessa maneira é mais fácil alterar qualquer visual futuramente.

Power Up Velocidade

Agora vamos trabalhar na implementação do nosso powerup de velocidade.

Crie um script com o nome de **PowerUpSpeedUp**. Configure-o como abaixo:

```
public class PowerUpSpeedUp : PowerUpBase
{
}
```

Power Up Velocidade

Agora adicione o seguinte código

```
[Header("Power Up Speed Up")]
public float amountToSpeed;

protected override void StartPowerUp()
{
    base.StartPowerUp();
    PlayerController.Instance.PowerUpSpeedUp(amountToSpeed);
    PlayerController.Instance.SetPowerUpText("Speed up");
}

protected override void EndPowerUp()
{
    base.EndPowerUp();
    PlayerController.Instance.ResetSpeed();
    PlayerController.Instance.SetPowerUpText("");
}
```

Power Up Velocidade

Adicione o código abaixo no PlayerController:

```
public void SetPowerUpText(string s)
{
    uiTextPowerUp.text = s;
}

public void PowerUpSpeedUp(float f)
{
    _currentSpeed = f;
}

public void ResetSpeed()
{
    _currentSpeed = speed;
}
```

Power Up Velocidade

Ainda no PlayerController:

```
private float _currentSpeed;  
  
private void Start()  
{  
    _startPosition = transform.position;  
    ResetSpeed();  
}
```

Power Up Velocidade

Ainda no PlayerController, altere a linha abaixo, dentro do método Update

```
transform.Translate(transform.forward * _currentSpeed * Time.deltaTime);
```

Power Up Velocidade

Transforme o PlayerController em um Singleton usando o component criado do pacote Ebac

```
using Ebac.Core.Singleton;  
  
public class PlayerController : Singleton<PlayerController>  
{  
}
```

Power Up Velocidade

Transforme o PlayerController em um Singleton usando o component criado do pacote Ebac

```
using Ebac.Core.Singleton;  
  
public class PlayerController : Singleton<PlayerController>  
{  
}
```

Power Up Velocidade

Duplicate algum powerup já criado, e altere o script para o script que acabamos de criar.

Power Up Invencível

Vamos implementar agora o Power Up invencível. Crie uma classe com nome de PowerUpInvencible. Adicione o seguinte código:

```
public class PowerUpInvencible : PowerUpBase
{
    protected override void StartPowerUp()
    {
        base.StartPowerUp();
        PlayerController.Instance.SetPowerUpText("Invencible");
        PlayerController.Instance.SetInvencible();
    }

    protected override void EndPowerUp()
    {
        base.EndPowerUp();
        PlayerController.Instance.SetInvencible(false);
        PlayerController.Instance.SetPowerUpText("");
    }
}
```

Power Up Invencível

Dentro de OnTriggerEnter, ajuste o código para o seguinte:

```
public bool invencible = false;

private void OnCollisionEnter(Collision collision)
{
    if(other.transform.tag == tagToChecEndLine)
    {
        if(!invencible) EndGame();
    }
}

private void OnTriggerEnter(Collider other)
{
    if(other.transform.tag == tagToChecEndLine)
    {
        if(!invencible) EndGame();
    }
}
```

Power Up Invencível

Dentro de OnTriggerEnter, ajuste o código para o seguinte:

```
public void SetInvencible(bool b)
{
    invencible = b;
}
```

Power Up Invencível

Duplicate algum powerup já criado, e altere o script para o script que acabamos de criar.

Power Up Invencível

Crie um TextMesh dentro do player. Dentro de PlayerController:

```
public void SetPowerupText(text t)
{
    uiTextPowerup.text = t;
}
```

Power Up MudarAltura

Crie um novo script com nome de PowerUpHeight. Adicione o seguinte código:

```
[Header("Power Up Height")]
public float amountHeight = 2;
public float animationDuration = .1f;
public DG.Tweening.Ease ease = DG.Tweening.Ease.OutBack;

protected override void StartPowerUp()
{
    base.StartPowerUp();
    PlayerController.Instance.ChangeHeight(amountHeight, duration, animationDuration, ease);
}
```

Power Up MudarAltura

Dentro do PlayerController, adicione o seguinte método:

```
public void ChangeHeight(float amount, float duration, float animationDuration, Ease ease)
{
    var p = transform.position;
    p.y = _startPosition.y + amount;
    transform.position = p;
}
```

Power Up MudarAltura

Ainda no PlayerController:

```
private Vector3 _startPosition;  
  
private void Start()  
{  
    _startPosition = transform.position;  
    ResetSpeed();  
}
```

Power Up MudarAltura

Ainda no PlayerController:

```
public void ResetHeight()
{
    var p = transform.position;
    p.y = _startPosition.y;
    transform.position = p;
}
```

Power Up MudarAltura

Vamos agora adicionar algumas animações por código. Primeiro, baixe e instale este pacote:

[DOTween \(HOTween v2\)](#)

Power Up MudarAltura

No PlayerController:

```
using DG.Tweening;

public void ChangeHeight(float amount, float duration, float animationDuration, Ease ease)
{
    /*var p = transform.position;
    p.y = _startPosition.y + amount;
    transform.position = p;*/

    transform.DOMoveY(_startPosition.y + amount,
        animationDuration).SetEase(ease);//.OnComplete(ResetHeight);a
    Invoke(nameof(ResetHeight), duration);
}
```

Power Up MudarAltura

Ainda no PlayerController:

```
public void ResetHeight()
{
    transform.DOMoveY(_startPosition.y, .1f);
}
```

Power Up Coletar Moedas

Vamos criar um novo script chamado **PowerUpCoins**.

```
public class PowerUpCoins : PowerUpBase
{
    [Header("Coin Collector")]
    public float sizeAmount = 7;

    protected override void StartPowerUp()
    {
        base.StartPowerUp();
        PlayerController.Instance.ChangeCoinCollectorSize(sizeAmount);
    }

    protected override void EndPowerUp()
    {
        base.EndPowerUp();
        PlayerController.Instance.ChangeCoinCollectorSize(1);
    }
}
```

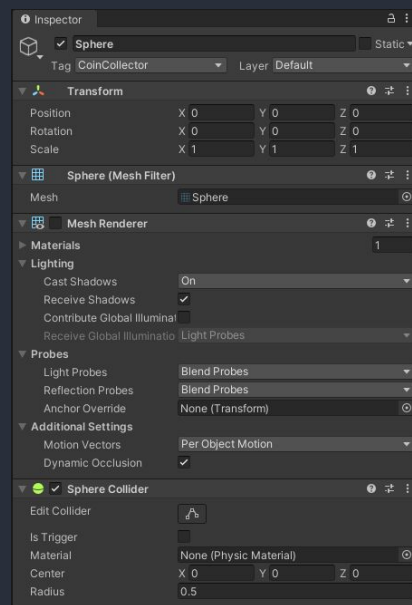
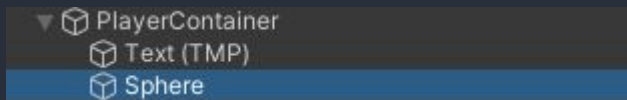
Power Up Coletar Moedas

No PlayerController:

```
[Header("Coin Setup")]  
public GameObject coinCollector;  
  
public void ChangeCoinCollectorSize(float amount)  
{  
    coinCollector.transform.localScale = Vector3.one * amount;  
}
```

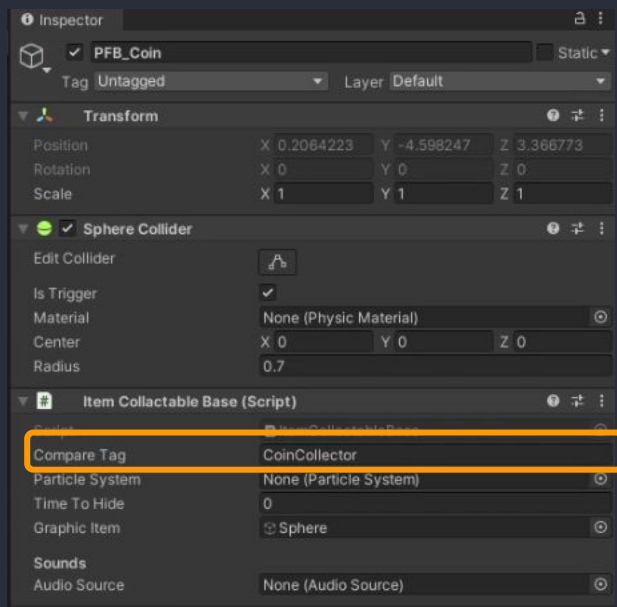
Power Up Coletar Moedas

Vamos mudar a maneira que o player coleta as moedas. Crie uma esfera dentro do player com as seguintes configurações. Crie uma nova tag para esse objeto, e dê o nome de CoinCollector.



Power Up Coletar Moedas

Atualize a configuração no PFB_Coin



Power Up Coletar Moedas

Atualize o script ItemCollectableCoin:

```
public Collider collider;
public bool collect = false;
public float lerp = 5f;
public float minDistance = 1f;

private void Start()
{
    CoinsAnimationManager.Instance.RegisterCoin(this);
}

protected override void OnCollect()
{
    base.OnCollect();
    collider.enabled = false;
    collect = true;
    PlayerController.Instance.Bounce();
}
```

Power Up Coletar Moedas

Atualize o script ItemCollectableCoin:

```
protected override void Collect()
{
    OnCollect();
}

private void Update()
{
    if(collect)
    {
        transform.position = Vector3.Lerp(transform.position,
        PlayerController.Instance.transform.position, lerp * Time.deltaTime);

        if(Vector3.Distance(transform.position, PlayerController.Instance.transform.position) <
        minDistance)
        {
            Hideltens();
            Destroy(gameObject);
        }
    }
}
```