Project

- Ability at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\AbilitysSkillsAndBuffsItems\Abilitys\Ability.cs:
Code of file Ability:

```
þÿusing UnityEngine;Ð
public abstract class Ability : ScriptableObjectÐ
{Ð
    public string abilityName;Ð
    public string abilityDescription;Ð
    public Sprite icon;Ð
    public float baseDamage;Ð
    public float strengthScaling;Ð
    public float intelligenceScaling;Ð
    Ð
    public float cooldown;Ð
    public string animationName;Ð
    public float lastTimeUsed = 0;Ð
    public float ActivateDelayTime = 0;Ð
Ð
    Ð
Ð
    public abstract void OnAbilityObjectHit(AbilityObject abilityObject,
GameObject target);Ð
    public abstract void Activate(AbilityData abilityData);Ð
Ð
    public virtual void PreActivateAbility(AbilityData abilityData){Ð
Ð
    }Ð
    protected virtual void StartActivation(AbilityData abilityData) { }Ð
    protected virtual void UpdateActivation(AbilityData abilityData) { }Ð
    protected virtual void EndActivation(AbilityData abilityData) { }Ð
Ð
     public delegate void AbilityEvent(Ability ability);Ð
     public delegate void AbilityObjectEvent(AbilityObject abilityObject,
GameObject target);Ð
Ð
     public event AbilityObjectEvent OnAbilityObjectSpawnedEvent;Ð
     public event AbilityObjectEvent OnAbilityObjectHitEvent;Ð
  Ð
    public event AbilityEvent OnAbilityActivated;Ð
    public float getLastTimeUsed()Ð
    {Ð
        return lastTimeUsed;Ð
    }Ð
    public float setLastTimeUsed(float time)Đ
    {Đ
        return lastTimeUsed = time;Đ
    }Đ
    protected void RaiseOnObjectSpawned(AbilityObject
abilityObject,GameObject target)Đ
    {Đ
        OnAbilityObjectSpawnedEvent?.Invoke(abilityObject,null);Đ
    }Đ
   Đ
    protected void RaiseOnObjectHit(AbilityObject abilityObject, GameObject
target)Đ
    {Đ
        OnAbilityObjectHitEvent?.Invoke(abilityObject, target);Đ
    }Đ
    protected void RaiseOnAbilityActivated()Đ
    {Đ
        OnAbilityActivated?.Invoke(this);Đ
    }Đ
    Đ
Đ
}Đ
Đ
public class AbilityDataĐ
{Đ
    public GameObject Target;Đ
    public CharacterStats CasterStats;Đ
    public AbilityController CasterController;Đ
    public CharacterCombatController CasterCombatController;Đ
    public float damage;Đ
    public float projectileSpeed;Đ
    public float stunDuration;Đ
Đ
}
```

- AbilityControllData at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\AbilitysSkillsAndBuffsItems\Abilitys\AbilityControllData.cs
:
Code of file AbilityControllData:

```
þÿusing UnityEngine;Đ
class AbilityControllDataĐ
{Đ
    public string type;Đ
    public Vector3 direction;Đ
    public GameObject target;Đ
    public Vector3 targetPosition;Đ
}Ð
Ð
```

- AbilityObject at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\AbilitysSkillsAndBuffsItems\Abilitys\AbilityObject.cs:

Code of file AbilityObject:

```
þÿusing System;Ð
using UnityEngine;Ð
Ð
public class AbilityObject : MonoBehaviourÐ
{Ð
    public event Action<GameObject> OnHit;Ð
    public event Action OnUpdate;Ð
    public event Action OnSpawn;Ð
    public event Action OnDelete;Ð
    public AbilityData data;Ð
Ð
    public bool shouldDestroy=false;Ð
Ð
    public bool deleteOnCollision = true;Ð
    public bool deleteOnTimer = false;Ð
    float timer = 0f;Ð
    public float timerMax = 5f;Ð
Ð
    Ð
Ð
    public Ability ParentAbility { get; set; }Ð
Ð
    protected virtual void HandleOnHit(GameObject target)Ð
    {Ð
        // Trigger OnHit event with target as parameterÐ
        OnHit?.Invoke(target);Ð
  Ð
        if(deleteOnCollision){Ð
            shouldDestroy = true;Ð
        }Ð
    }Ð
Ð
    private void Update()Ð
    {Ð
        // Trigger OnHit event with target as parameterÐ
        OnUpdate?.Invoke();Ð
        if(deleteOnTimer){Ð
            timer += Time.deltaTime;Ð
            if(timer >= timerMax){Ð
```

```
                timer = 0f;Đ
                HandleOnDelete();Đ
            }Đ
        }Đ
  Đ
    }Đ
Đ
    protected void HandleOnSpawn()Đ
    {Đ
        // Perform any initialization or setup for the ability object hereĐ
        // Trigger OnSpawn eventĐ
        OnSpawn?.Invoke();Đ
        Đ
  Đ
    }Đ
Đ
    protected void HandleOnDelete()Đ
    {Đ
        // Perform any cleanup or deactivation for the ability object hereĐ
        // Trigger OnDelete eventĐ
        OnDelete?.Invoke();Đ
        Destroy(gameObject);Đ
        Đ
    }Đ
Đ
    public void Awake()Đ
    {Đ
        HandleOnSpawn();Đ
    }Đ
Đ
    private void OnTriggerEnter(Collider collision)Đ
    {Đ
        Debug.Log("OnTriggerEnter");Đ
        if(data == null){Đ
            Debug.LogError("AbilityObject data is null");Đ
            return;Đ
        }Đ
        // Get target HealthController from collided objectĐ
        if (data.CasterStats != null)Đ
        {Đ
            if (data.CasterStats.gameObject.name == collision.gameObject.name)Đ
            {Đ
                return;Đ
            }Đ
            if (gameObject.name == collision.gameObject.name)Đ
            {Đ
                return;Đ
            }Đ
        }Đ
Đ
            Đ
            // Call HandleOnHit method with target as parameterĐ
            ParentAbility?.OnAbilityObjectHit(this, collision.gameObject);Đ
            HandleOnHit(collision.gameObject);Đ
Đ
    }Đ
}Đ
public interface IBouncingAbilityObjectĐ
{Đ
     float BounceIntensity { get; set; }Đ
    float BounceDuration { get; set; }Đ
    void Bounce(GameObject target);Đ
}Đ
Đ
public interface IPiercingAbilityObjectĐ
{Đ
    void Pierce(GameObject target);Đ
}Đ
Đ
public interface IHomingAbilityObjectĐ
{Đ
    void Home(GameObject target);Đ
}
```

- BaseProjectileObject at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\AbilitysSkillsAndBuffsItems\Abilitys\BaseProjectileObject.cs:

Code of file BaseProjectileObject:

```
using System.Collections;Đ
using UnityEngine;Đ
public class BaseProjectileObject : AbilityObject, IBouncingAbilityObject,
IPiercingAbilityObjectĐ
{Đ
    public float BounceIntensity { get; set; }Đ
    public float BounceDuration { get; set; }Đ
Đ
    public int bounceCount;Đ
    public int pierceCount;Đ
Đ
Đ
    protected override void HandleOnHit(GameObject target)Đ
    {Đ
    // Apply damage to the targetĐ
        if (data.CasterStats != null)Đ
        {Đ
            HealthController targetStats = target.GetComponent<HealthController>();Đ
            if (targetStats != null)Đ
            {Đ
                float damage = data.damage;Đ
                targetStats.TakeDamage(damage, data.CasterStats.gameObject);Đ
            }Đ
        }Đ
Đ
        // Handle Bounce and Pierce logicĐ
        shouldDestroy = deleteOnCollision;Đ
        if (bounceCount > 0)Đ
        {Đ
            Bounce(target);Đ
        }Đ
        else if (pierceCount > 0)Đ
        {Đ
            Pierce(target);Đ
        }Đ
Đ
        if (shouldDestroy)Đ
        {Đ
            HandleOnDelete();Đ
        }Đ
       Đ
    }Đ
Đ
    public void Bounce(GameObject target)Đ
    {Đ
        Đ
        shouldDestroy = false;Đ
        bounceCount--;Đ
Đ
        Vector3 bounceDirection = Vector3.Reflect(transform.forward,
target.transform.up);Đ
        transform.forward = bounceDirection;Đ
Đ
        Rigidbody rb = GetComponent<Rigidbody>();Đ
        rb.velocity = bounceDirection * data.projectileSpeed;Đ
Đ
        Đ
    }Đ
```

```
Đ
    public void Pierce(GameObject target)Đ
    {Đ
Đ
        pierceCount--;Đ
Đ
        shouldDestroy = false;Đ
    }Đ
Đ
   Đ
}
```

- DefaultProjectileAbility at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\AbilitysSkillsAndBuffsItems\Abilitys\DefaultProjectileAbility.cs:

Code of file DefaultProjectileAbility:

```
using System.Collections;Đ
using UnityEngine;Đ
Đ
// Base Projectile Ability classĐ
[CreateAssetMenu(menuName = "Abilities/DefaultProjectileAbility")]Đ
public class DefaultProjectileAbility : AbilityĐ
{Đ
    public GameObject projectilePrefab;Đ
    public float projectileSpeed = 5f;Đ
Đ
    public override void OnAbilityObjectHit(AbilityObject abilityObject,
GameObject target)Đ
    {Đ
        if(abilityObject.data.CasterStats != null)Đ
        {Đ
            HealthController targetStats = target.GetComponent<HealthController>();Đ
            if (targetStats != null)Đ
            {Đ
                float damage = abilityObject.data.damage;Đ

targetStats.TakeDamage(damage,abilityObject.data.CasterStats.gameObject);Đ
            }Đ
        }Đ
         RaiseOnObjectHit(abilityObject,target);Đ
  Đ
Đ
    }Đ
Đ
    public override void Activate(AbilityData abilityData)Đ
    {Đ
```

```
        if (abilityData.CasterStats == null) return;Ð
Ð
        Transform firePoint =
abilityData.CasterStats.GetComponent<AbilityController>().firePoint;Ð
        Ð
        GameObject projectileInstance = Instantiate(projectilePrefab,
firePoint.position, firePoint.rotation);Ð
        BaseProjectileObject abilityObject =
projectileInstance.GetComponent<BaseProjectileObject>();Ð
        RaiseOnObjectSpawned(abilityObject,null);Ð
Ð
        Rigidbody rb = projectileInstance.GetComponent<Rigidbody>();Ð
        rb.velocity = firePoint.forward * projectileSpeed;Ð
Ð
        Ð
        abilityObject.ParentAbility = this;Ð
        abilityObject.data = abilityData;Ð
        abilityData.projectileSpeed = projectileSpeed;Ð
    }Ð
}Ð
Ð
// Base Projectile Object classÐ
Ð
```

- DefaultSkill at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\AbilitysSkillsAndBuffsItems\Abilitys\DefaultSkill.cs:
Code of file DefaultSkill:

```
þÿÐ
using UnityEngine;Ð
Ð
[CreateAssetMenu(fileName = "Skill", menuName = "Skill/Skill", order = 1)]Ð
public class DefaultSkill : SkillÐ
{Ð
Ð
}
```

- FireBall at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\AbilitysSkillsAndBuffsItems\Abilitys\FireBall.cs:
Code of file FireBall:

```
using UnityEngine;Ð
[CreateAssetMenu(menuName = "Abilities/Fireball")]Ð
class FireBall : DefaultProjectileAbility {Ð
    //Expecting BaseProjectileObject to be a prefabÐ
  public override void OnAbilityObjectHit(AbilityObject abilityObject, GameObject
target) {Ð
    if (abilityObject.data.CasterStats != null) {Ð
```

```
      HealthController targetHealth = target.GetComponent<HealthController>();Ð
      if (targetHealth != null) {Ð
        float damage = abilityObject.data.damage;Ð

targetHealth.TakeDamage(damage,abilityObject.data.CasterStats.gameObject);Ð
      }Ð
    }Ð
    RaiseOnObjectHit(abilityObject, target);Ð
  }Ð
Ð
    public override void Activate(AbilityData abilityData) {Ð
    if (abilityData.CasterStats == null) return;Ð
Ð
    Transform firePoint =
abilityData.CasterStats.GetComponent<AbilityController>().firePoint;Ð
Ð
    GameObject projectileInstance = Instantiate(projectilePrefab, firePoint.position,
firePoint.rotation);Ð
    BaseProjectileObject abilityObject =
projectileInstance.GetComponent<BaseProjectileObject>();Ð
    RaiseOnObjectSpawned(abilityObject, null);Ð
Ð
    Rigidbody rb = projectileInstance.GetComponent<Rigidbody>();Ð
    rb.velocity = firePoint.forward * projectileSpeed;Ð
Ð
    abilityObject.ParentAbility = this;Ð
    abilityObject.data = abilityData;Ð
    abilityData.projectileSpeed = projectileSpeed;Ð
  }Ð
}Ð
```

- ShieldBash at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\AbilitysSkillsAndBuffsItems\Abilitys\ShieldBash.cs:
Code of file ShieldBash:

```
using UnityEngine;Ð
[CreateAssetMenu(menuName = "Abilities/ShieldBash")]Ð
public class ShieldBash : AbilityÐ
{Ð
    public GameObject prefabAbilityObject;Ð
    public float stunDuration = 2f;Ð
    public ShieldBash(){Ð
        abilityName = "Shield Bash";Ð
        baseDamage = 50;Ð
        strengthScaling = 0.5f;Ð
        intelligenceScaling = 0.5f;Ð
        animationName = "Shield Bash animation";Đ
        Đ
    }Đ
Đ
    public override void OnAbilityObjectHit(AbilityObject abilityObject,
GameObject target)Đ
    {Đ
        if (abilityObject.data.CasterStats != null)Đ
        {Đ
            HealthController targetStats = target.GetComponent<HealthController>();Đ
            if (targetStats != null)Đ
            {Đ
                float damage = abilityObject.data.damage;Đ

targetStats.TakeDamage(damage,abilityObject.data.CasterStats.gameObject);Đ
Đ
                if (abilityObject.data.stunDuration >= 0f)Đ
                {Đ

targetStats.GetComponent<IStunnable>().Stun(abilityObject.data.stunDuration);Đ
                }Đ
            }Đ
        }Đ
        RaiseOnObjectHit(abilityObject, target);Đ
    }Đ
Đ
    public override void Activate(AbilityData abilityData)Đ
    {Đ
        if (abilityData.CasterStats == null) return;Đ
Đ
        Transform casterTransform = abilityData.CasterStats.transform;Đ
        Vector3 forwardDirection = casterTransform.forward;Đ
Đ
        GameObject abilityObjectInstance = Instantiate(prefabAbilityObject,
casterTransform.position + forwardDirection, Quaternion.identity);Đ
        AbilityObject abilityObject =
abilityObjectInstance.GetComponent<AbilityObject>();Đ
        RaiseOnObjectSpawned(abilityObject, null);Đ
Đ
        Rigidbody rb = abilityObjectInstance.GetComponent<Rigidbody>();Đ
        rb.velocity = forwardDirection * abilityData.projectileSpeed;Đ
Đ
        abilityObject.data = abilityData;Đ
        abilityData.Target = null;Đ
        abilityData.projectileSpeed = 0f;Đ
        abilityObject.ParentAbility = this;Ð
        abilityData.stunDuration = stunDuration;Ð
    }Ð
}Ð
Ð
Ð
```

- SimpleStrike at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\AbilitysSkillsAndBuffsItems\Abilitys\SimpleStrike.cs:
Code of file SimpleStrike:

```
þÿusing UnityEngine;Ð
Ð
[CreateAssetMenu(menuName = "Abilities/SimpleStrike")]Ð
public class SimpleStrike : AbilityÐ
{Ð
    // SimpleStrike specific properties, if anyÐ
    public GameObject MeelePrefab;Ð
    public float lifeTime = 0.5f;Ð
Ð
    public override void OnAbilityObjectHit(AbilityObject abilityObject,
GameObject target)Ð
    {Ð
        HealthController healthController =
target.GetComponent<HealthController>();Ð
        if (healthController != null)Ð
        {Ð
Ð
            healthController.TakeDamage(abilityObject.data.damage,abilityObject.data.
CasterStats.gameObject);Ð
        }Ð
    }Ð
Ð
Ð
    public override void Activate(AbilityData abilityData)Ð
    {Ð
        GameObject meleeStrikeInstance = Instantiate(MeelePrefab,
abilityData.CasterStats.transform.position, Quaternion.identity);Ð
        MeleeStrikeObject abilityObject =
meleeStrikeInstance.AddComponent<MeleeStrikeObject>();Ð
Ð
        abilityObject.ParentAbility = this;Ð
        abilityObject.data = abilityData;Ð
        Destroy(meleeStrikeInstance, lifeTime);Ð
    }Ð
Ð
Ð
}Ð
Ð
public class MeleeStrikeObject : AbilityObjectÐ
{Ð
    private void OnTriggerEnter(Collider collision)Ð
    {Ð
        if (data.CasterStats != null)Ð
        {Ð
            if (data.CasterStats.gameObject.name == collision.gameObject.name)Ð
            {Ð
                return;Ð
            }Ð
            if (gameObject.name == collision.gameObject.name)Ð
            {Ð
                return;Ð
            }Ð
        }Ð
Ð
        HandleOnHit(collision.gameObject);Ð
        ParentAbility.OnAbilityObjectHit(this, collision.gameObject);Ð
Ð
    }Ð
}Ð
```

- Buff at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\AbilitysSkillsAndBuffsItems\Buffs\Buff.cs:
Code of file Buff:

```
þÿusing System.Collections.Generic;Ð
using UnityEngine;Ð
[System.Serializable]Ð
public class Buff : ScriptableObjectÐ
{Ð
    public Buff()Ð
    {Ð
      statModifier = new StatsModifier();Ð
    }Ð
    public StatsModifier statModifier;Ð
    public string buffName;Ð
    public float duration;Ð
    public bool stackable;Ð
    public int maxStacks;Ð
Ð
    private event System.Action<BuffInstance> OnApply;Ð
    private event System.Action<BuffInstance> OnFade;Ð
    private event System.Action<BuffInstance> OnHit;Đ
Đ
    public virtual void InvokeOnApply(BuffInstance buffInstance)Đ
    {Đ
        OnApply?.Invoke(buffInstance);Đ
    }Đ
Đ
    public virtual void InvokeOnFade(BuffInstance buffInstance)Đ
    {Đ
        OnFade?.Invoke(buffInstance);Đ
    }Đ
Đ
    public virtual void InvokeOnHit(BuffInstance buffInstance)Đ
    {Đ
        OnHit?.Invoke(buffInstance);Đ
    }Đ
Đ
    public List<string> GetEventTypes()Đ
    {Đ
        List<string> eventTypes = new List<string>();Đ
Đ
        if (OnApply != null)Đ
        {Đ
            eventTypes.Add("OnApply");Đ
        }Đ
Đ
        if (OnFade != null)Đ
        {Đ
            eventTypes.Add("OnFade");Đ
        }Đ
Đ
        if (OnHit != null)Đ
        {Đ
            eventTypes.Add("OnHit");Đ
        }Đ
Đ
        return eventTypes;Đ
    }Đ
}Đ
```

- BuffInstance at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\AbilitysSkillsAndBuffsItems\Buffs\BuffInstance.cs:
Code of file BuffInstance:

```
þÿusing UnityEngine;Đ
Đ
```

```
public class BuffInstanceÐ
{Ð
    public Buff buff;Ð
    public GameObject target;Ð
    public int currentStacks;Ð
    public float remainingDuration;Ð
    StatsModifier characterStats;Ð
Ð
    public BuffInstance(Buff buff, GameObject target, int initialStacks, float
initialDuration)Ð
    {Ð
        this.buff = buff;Ð
        this.target = target;Ð
        this.currentStacks = initialStacks;Ð
        this.remainingDuration = initialDuration;Ð
        characterStats = target.GetComponent<BuffSystem>().TotalstatsModifier;Ð
    }Ð
Ð
    public void Update()Ð
    {Ð
        remainingDuration -= Time.deltaTime;Ð
Ð
        if (remainingDuration <= 0)Ð
        {Ð
            OnBuffFade();Ð
            target.GetComponent<BuffSystem>().RemoveBuff(buff); // add this lineÐ
            return;Ð
        }Ð
Ð
        // Perform any other update logic specific to the buffÐ
    }Ð
Ð
    public void Refresh(float duration)Ð
    {Ð
        remainingDuration = duration;Ð
    }Ð
Ð
    public void AddStack()Ð
    {Ð
        currentStacks++;Ð
        OnBuffApply();Ð
    }Ð
Ð
 Ð
Ð
```

```
    public void OnBuffApply()Ð
    {Ð
        // Perform any actions or apply stat changes when the buff is appliedÐ
        if (buff.statModifier != null)Ð
        {Ð
            characterStats.Add(buff.statModifier);Ð
            target.GetComponent<CharacterStats>().UpdateSubStats();Ð
        }Ð
        buff.InvokeOnApply(this);Ð
    }Ð
Ð
    public void OnBuffFade()Ð
    {Ð
        // Perform any actions or apply stat changes when the buff is appliedÐ
        if (buff.statModifier != null)Ð
        {Ð
            characterStats.Sub(buff.statModifier);Ð
            target.GetComponent<CharacterStats>().UpdateSubStats();Ð
        }Ð
        buff.InvokeOnFade(this);Ð
        Ð
    }Ð
Ð
    public void OnBuffHit()Ð
    {Ð
        // Perform any actions or apply effects when the buff "hits" (e.g., dealing
damage or applying a debuff)Ð
        buff.InvokeOnHit(this);Ð
    }Ð
}Ð
```

- IStatsProvider at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\AbilitysSkillsAndBuffsItems\IStatsProvider.cs:
Code of file IStatsProvider:

```
þÿpublic interface IStatsProviderÐ
{Ð
    CharacterStats GetCharacterStats();Ð
}Ð
```

- Item at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\AbilitysSkillsAndBuffsItems\items\Item.cs:
Code of file Item:

```
þÿusing UnityEngine;Ð
Ð
public abstract class Item : ScriptableObjectÐ
```

```
{Đ
    public string itemName;Đ
    public string description;Đ
    public Sprite icon;Đ
}Đ
Đ
Đ
[System.Serializable]Đ
public class EquipableItem : ItemĐ
{Đ
    public EquipManager.EquipmentType equipmentType;Đ
    public float strengthBonus;Đ
    public float intelligenceBonus;Đ
    public float dexterityBonus;Đ
    public float enduranceBonus;Đ
    public float wisdomBonus;Đ
Đ
    public StatsModifier subStatsModifier;Đ
}Đ
```

- BouceSkill at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\AbilitysSkillsAndBuffsItems\Skills\BouceSkill.cs:
Code of file BouceSkill:

- FireballMasterySkill at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\AbilitysSkillsAndBuffsItems\Skills\FireballMasterySkill.cs:
Code of file FireballMasterySkill:

```
using UnityEngine;Đ
[CreateAssetMenu(fileName = "FireBallMastery", menuName = "Skill/
FireBallMastery", order = 1)]Đ
public class FireballMasterySkill : SkillĐ
{Đ
    [SerializeField]Đ
    private GameObject explosionPrefab;Đ
Đ
    public override void ApplySkill(CharacterStats playerStats)Đ
    {Đ
        Debug.Log("Apply Skill");Đ
        FireBall fireballAbility = GetFireballAbility(playerStats);Đ
        if (fireballAbility != null)Đ
        {Đ
            Debug.Log("Fireball Ability found");Đ
            fireballAbility.OnAbilityObjectHitEvent += ExplodeOnHit;Đ
        }Đ
    }Đ
```

```
Đ
    public override void RemoveSkill(CharacterStats playerStats)Đ
    {Đ
        FireBall fireballAbility = GetFireballAbility(playerStats);Đ
        if (fireballAbility != null)Đ
        {Đ
            fireballAbility.OnAbilityObjectHitEvent -= ExplodeOnHit;Đ
        }Đ
    }Đ
Đ
    private FireBall GetFireballAbility(CharacterStats playerStats)Đ
    {Đ
        AbilityController abilityController =
playerStats.GetComponent<AbilityController>();Đ
        return abilityController.learnedAbilitys.Find(a => a is FireBall) as FireBall;Đ
    }Đ
Đ
    private void ExplodeOnHit(AbilityObject abilityObject, GameObject target)Đ
    {Đ
        Debug.Log("EXPLODE ON Hit");Đ
        ApplyDamageToTargets(abilityObject.transform.position, 2f,
abilityObject.data.damage * 0.5f);Đ
        InstantiateExplosion(abilityObject.transform.position);Đ
    }Đ
Đ
    private void ApplyDamageToTargets(Vector3 position, float radius, float
damage)Đ
    {Đ
   Đ
    }Đ
Đ
    private void InstantiateExplosion(Vector3 position)Đ
    {Đ
        if (explosionPrefab != null)Đ
        {Đ
            GameObject explosion = Instantiate(explosionPrefab, position,
Quaternion.identity);Đ
            // Add additional logic for the explosion, such as configuring the
explosion's lifetime or assigning its parentĐ
        }Đ
        elseĐ
        {Đ
            Debug.LogWarning("No explosion prefab assigned to
FireballMasterySkill.");Đ
        }Đ
    }Đ
}Đ
```

- Skill at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\AbilitysSkillsAndBuffsItems\Skills\Skill.cs:

Code of file Skill:

```
þÿusing System.Collections.Generic;Đ
using UnityEngine;Đ
Đ
public abstract class Skill : ScriptableObjectĐ
{Đ
    public string skillName;Đ
    public List<Archetype> archTypes;Đ
    public StatsModifier statModifier;Đ
Đ
    public virtual void ApplySkill(CharacterStats characterStats)Đ
    {Đ
        // Implement skill-specific behavior in derived classesĐ
    }Đ
Đ
    public virtual void RemoveSkill(CharacterStats characterStats)Đ
    {Đ
        // Implement skill-specific behavior in derived classesĐ
    }Đ
    public virtual void OnSpawnAbilityObject(AbilityObject abilityObject,
AbilityData abilityData)Đ
    {Đ
Đ
    }Đ
    Đ
}Đ
Đ
Đ
public enum ArchetypeĐ
{Đ
    Strength,Đ
    Intelligence,Đ
    Dexterity,Đ
    Endurance,Đ
   WisdomĐ
}
```

- SkillNode at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\AbilitysSkillsAndBuffsItems\Skills\SkillNode.cs:

Code of file SkillNode:

```
þÿusing System.Collections.Generic;Đ
```

```
using UnityEngine;Ð
Ð
[CreateAssetMenu(fileName = "SkillNode", menuName = "SkillTree/SkillNode",
order = 0)]Ð
public class SkillNode : ScriptableObjectÐ
{Ð
    public string skillName;Ð
    public string skillDescription;Ð
    public int skillPointCost;Ð
    public Sprite icon;Ð
    public List<Archetype> mainStatRequirement;Ð
    public List<int> mainStatValue;Ð
    public Skill skill;Ð
    public SkillNode prerequisiteSkill;Ð
    public bool isUnlocked = false;Ð
Ð
Ð
}Ð
```

- SkillNodeFactory at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\AbilitysSkillsAndBuffsItems\Skills\SkillNodeFactory.cs:
Code of file SkillNodeFactory:

```
using System.Collections.Generic;Ð
using UnityEditor;Ð
using UnityEngine;Ð
public static class SkillNodeFactory{Ð
    Ð
     public static SkillNode CreateSkillNode(SkillNodeFactoryDataClass data){Ð
        //set save pathÐ
Ð
         SkillNode skillNode = ScriptableObject.CreateInstance<SkillNode>();Ð
         skillNode.name = data.skillName+"SkillNode";Ð
         skillNode.skillName = data.skillName;Ð
         skillNode.skillDescription = data.skillDescription;Ð
         skillNode.skillPointCost = data.skillPointCost;Ð
         skillNode.icon = data.icon;Ð
         skillNode.mainStatRequirement = data.mainStatRequirement;Ð
         skillNode.mainStatValue = data.mainStatValue;Ð
         skillNode.skill = data.skill;Ð
         skillNode.prerequisiteSkill = data.prerequisiteSkill;Ð
         skillNode.isUnlocked = data.isUnlocked;Ð
Ð
         AssetDatabase.CreateAsset(skillNode, "Assets/Resources/
SkillNodes/"+skillNode.name+".asset");Ð
         return skillNode;Ð
    }Ð
Ð
    Ð
    Ð
Ð
}Ð
public class SkillNodeFactoryDataClass{Ð
        public string skillName;Ð
        public string skillDescription;Ð
        public int skillPointCost;Ð
        public Sprite icon;Ð
        public List<Archetype> mainStatRequirement;Ð
        public List<int> mainStatValue;Ð
        public Skill skill;Ð
        public SkillNode prerequisiteSkill;Ð
        public bool isUnlocked;Ð
        public SkillNodeFactoryDataClass(string skillName, string skillDescription,
int skillPointCost, Sprite icon, List<Archetype> mainStatRequirement, List<int>
mainStatValue, Skill skill, SkillNode prerequisiteSkill, bool isUnlocked){Ð
            this.skillName = skillName;Ð
            this.skillDescription = skillDescription;Ð
            this.skillPointCost = skillPointCost;Ð
            this.icon = icon;Ð
            this.mainStatRequirement = mainStatRequirement;Ð
            this.mainStatValue = mainStatValue;Ð
            this.skill = skill;Ð
            this.prerequisiteSkill = prerequisiteSkill;Ð
            this.isUnlocked = isUnlocked;Ð
        }Ð
}
```

- SkillTree at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\AbilitysSkillsAndBuffsItems\SkillTree.cs:

Code of file SkillTree:

```
þÿusing System.Collections.Generic;Ð
using UnityEngine;Ð
Ð
[CreateAssetMenu(fileName = "SkillTree", menuName = "SkillTree/SkillTree",
order = 1)]Ð
public class SkillTree : ScriptableObjectÐ
{Ð
    public List<SkillNode> skillNodes;Ð
    public SkillTree()Ð
    {Ð
        skillNodes = new List<SkillNode>();Ð
    }Ð
Đ
    public void AddSkillNode(SkillNode skillNode)Đ
    {Đ
        skillNodes.Add(skillNode);Đ
    }Đ
    internal bool IsVisible(SkillNode skillNode)Đ
    {Đ
        return true;Đ
    }Đ
    private void Awake()Đ
    {Đ
        resetAllNodes();Đ
    }Đ
    public void resetAllNodes()Đ
    {Đ
        foreach (SkillNode node in skillNodes)Đ
        {Đ
            node.isUnlocked = false;Đ
Đ
        }Đ
    }Đ
}Đ
Đ
```

- StatsModifier at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\AbilitysSkillsAndBuffsItems\StatsModifier.cs:
Code of file StatsModifier:

```
þÿ[System.Serializable]Đ
public class StatsModifierĐ
{Đ
    public float Strength;Đ
    public float Intelligence;Đ
    public float Dexterity;Đ
    public float Endurance;Đ
    public float Wisdom;Đ
Đ
    public float attackSpeed;Đ
    public float criticalChance;Đ
    public float criticalDamage;Đ
    public float spellCriticalChance;Đ
    public float spellCriticalDamage;Đ
    public float cooldown;Đ
    public float dodgeChance;Đ
    public float armor;Đ
    public float magicResistance;Đ
    public float maxLife;Đ
    public float maxMana;Đ
    public float lifeRegen;Đ
    public float manaRegen;Đ
    public StatsModifier(Đ
        float strength = 0f,Đ
        float intelligence = 0f,Đ
        float dexterity = 0f,Đ
        float endurance = 0f,Đ
        float wisdom = 0f,Đ
        float attackSpeed = 0f,Đ
        float criticalChance = 0f,Đ
        float criticalDamage = 0f,Đ
        float spellCriticalChance = 0f,Đ
        float spellCriticalDamage = 0f,Đ
        float cooldown = 0f,Đ
        float dodgeChance = 0f,Đ
        float armor = 0f,Đ
        float magicResistance = 0f,Đ
        float maxLife = 0f,Đ
        float maxMana = 0f,Đ
        float lifeRegen = 0f,Đ
        float manaRegen = 0fĐ
    )Đ
    {Đ
        Strength = strength;Đ
        Intelligence = intelligence;Đ
        Dexterity = dexterity;Đ
        Endurance = endurance;Đ
        Wisdom = wisdom;Đ
        this.attackSpeed = attackSpeed;Đ
        this.criticalChance = criticalChance;Đ
        this.criticalDamage = criticalDamage;Đ
        this.spellCriticalChance = spellCriticalChance;Đ
        this.spellCriticalDamage = spellCriticalDamage;Đ
        this.cooldown = cooldown;Đ
        this.dodgeChance = dodgeChance;Đ
        this.armor = armor;Đ
        this.magicResistance = magicResistance;Đ
        this.maxLife = maxLife;Đ
        this.maxMana = maxMana;Đ
        this.lifeRegen = lifeRegen;Đ
        this.manaRegen = manaRegen;Đ
    }Đ
Đ
```

```
    public void Add(StatsModifier other)Đ
    {Đ
        Strength += other.Strength;Đ
        Intelligence += other.Intelligence;Đ
        Dexterity += other.Dexterity;Đ
        Endurance += other.Endurance;Đ
        Wisdom += other.Wisdom;Đ
Đ
        attackSpeed += other.attackSpeed;Đ
        criticalChance += other.criticalChance;Đ
        criticalDamage += other.criticalDamage;Đ
        spellCriticalChance += other.spellCriticalChance;Đ
        spellCriticalDamage += other.spellCriticalDamage;Đ
        cooldown += other.cooldown;Đ
        dodgeChance += other.dodgeChance;Đ
        armor += other.armor;Đ
        magicResistance += other.magicResistance;Đ
        maxLife += other.maxLife;Đ
        maxMana += other.maxMana;Đ
        lifeRegen += other.lifeRegen;Đ
        manaRegen += other.manaRegen;Đ
    }Đ
    public void Sub(StatsModifier other)Đ
    {Đ
        Strength -= other.Strength;Đ
        Intelligence -= other.Intelligence;Đ
Đ
        Dexterity -= other.Dexterity;Đ
        Endurance -= other.Endurance;Đ
        Wisdom -= other.Wisdom;Đ
        Đ
        attackSpeed -= other.attackSpeed;Đ
        criticalChance -= other.criticalChance;Đ
        criticalDamage -= other.criticalDamage;Đ
        spellCriticalChance -= other.spellCriticalChance;Đ
        spellCriticalDamage -= other.spellCriticalDamage;Đ
        cooldown -= other.cooldown;Đ
        dodgeChance -= other.dodgeChance;Đ
        armor -= other.armor;Đ
        magicResistance -= other.magicResistance;Đ
        maxLife -= other.maxLife;Đ
        maxMana -= other.maxMana;Đ
        lifeRegen -= other.lifeRegen;Đ
        manaRegen -= other.manaRegen;Đ
    }Đ
```

```
}Ð
```

- VisualEffectManager at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\GlobalManager\VisualEffectManager.cs:
Code of file VisualEffectManager:

```
using System.Collections.Generic;Ð
using UnityEngine;Ð
Ð
[CreateAssetMenu(fileName = "VisualEffectManager", menuName =
"ScriptableObjects/VisualEffectManager", order = 1)]Ð
public class VisualEffectManager : ScriptableObjectÐ
{Ð
    [System.Serializable]Ð
    public struct VisualEffectÐ
    {Ð
        public string effectName;Ð
        public GameObject effectPrefab;Ð
    }Ð
Ð
    public List<VisualEffect> visualEffects;Ð
Ð
    public GameObject GetEffectPrefab(string effectName)Ð
    {Ð
        foreach (var effect in visualEffects)Ð
        {Ð
            if (effect.effectName == effectName)Ð
            {Ð
                return effect.effectPrefab;Ð
            }Ð
        }Ð
        Debug.LogError($"No effect with name {effectName} found!");Ð
        return null;Ð
    }Ð
}Ð
```

- AbilityController at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\AbilityController.cs:
Code of file AbilityController:

```
using System.Collections;Ð
using System.Collections.Generic;Ð
Ð
using UnityEngine;Ð
public class AbilityController : MonoBehaviourÐ
{Ð
    public Transform firePoint;Ð
    public List<Ability> learnedAbilitys;Ð
    public List<(string,float)> lastTimeAbilityUsed;Ð
    private IStatsProvider statsProvider;Ð
    private AnimationController animationController;Ð
Ð
    private void Awake()Ð
    {Ð
        statsProvider = GetComponent<IStatsProvider>();Ð
        lastTimeAbilityUsed = new List<(string, float)>();Ð
        animationController = GetComponent<AnimationController>();Ð
    }Ð
Ð
    public void CastAbility(Ability ability, AbilityData abilityData)Ð
    {Ð
            ability.PreActivateAbility(abilityData);Ð
            animationController.PlayAnimation(ability.animationName);Ð
Ð
        if(ability.ActivateDelayTime == 0)Ð
        {Ð
Ð
            ability.Activate(abilityData);Ð
        }Ð
        elseÐ
        {Ð
Ð
            StartCoroutine(CastAfterDelay(ability, abilityData));Ð
        }Ð
    }Ð
    public IEnumerator CastAfterDelay(Ability ability, AbilityData abilityData)Ð
    {Ð
        yield return new WaitForSeconds(animationController.returnAnimationDelay(
ability.animationName));Ð

Debug.Log(animationController.returnAnimationDelay(ability.animationName)+"
DELAY");Ð
        ability.Activate(abilityData);Ð
    }Ð
    public void AddAbility(Ability ability)Ð
    {Ð
        learnedAbilitys.Add(ability);Ð
    }Ð
    public bool checkCooldown(string abilityName,float cooldown){Ð
        foreach ((string,float) paar in lastTimeAbilityUsed)Ð
        {Ð
            if(paar.Item1 == abilityName){Ð
```

```
                if(Time.time - paar.Item2 < cooldown){Đ
                    return false;Đ
                }Đ
            }Đ
        }Đ
        return true;Đ
    }Đ
    public void setCooldown(string abilityName,float cooldown){Đ
        bool found = false;Đ
        for (int i = 0; i < lastTimeAbilityUsed.Count; i++)Đ
        {Đ
            if(lastTimeAbilityUsed[i].Item1 == abilityName){Đ
                lastTimeAbilityUsed[i] = (abilityName,Time.time);Đ
                found = true;Đ
            }Đ
        }Đ
        if(!found){Đ
            lastTimeAbilityUsed.Add((abilityName,Time.time));Đ
        }Đ
    }Đ
}Đ
Đ
```

- AIController at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\AIController.cs:
Code of file AIController:

```
using System;Đ
using UnityEngine;Đ
using UnityEngine.AI;Đ
Đ
public class AIController : MonoBehaviourĐ
{Đ
    public AIState currentState;Đ
    public IdleState idleState;Đ
    public FollowState followState;Đ
    public AssistState assistState;Đ
    public PatrolState patrolState;Đ
Đ
    private NavMeshAgent navMeshAgent;Đ
Đ
    public ChaseState chaseState;Đ
    public AttackState attackState;Đ
Đ
    public Transform target;Đ
    public float aggroRadius;Đ
    public string aggroTag;Đ
    public float attackInterval;Đ
    public Ability attackAbility;Đ
    public  float attackRange;Đ
Đ
    private void Start()Đ
    {Đ
        navMeshAgent = GetComponent<NavMeshAgent>();Đ
        currentState = idleState;Đ
        PatrolStateMonoBehaviour patrolStateMonoBehaviour =
GetComponent<PatrolStateMonoBehaviour>();Đ
        if (patrolStateMonoBehaviour != null)Đ
        {Đ
            patrolState.waypoints = new
System.Collections.Generic.List<Transform>();Đ
            foreach (GameObject g in patrolStateMonoBehaviour.waypoints)Đ
            {Đ
                patrolState.waypoints.Add(g.transform);Đ
            }Đ
        }Đ
        animator = GetComponent<Animator>();Đ
    }Đ
    public NavMeshAgent getNavMeshAgent()Đ
    {Đ
        return navMeshAgent;Đ
    }Đ
    Animator animator;Đ
    private void Update()Đ
    {Đ
                if(navMeshAgent==null){Đ
            return;Đ
        }Đ
        if(animator==null){Đ
            return;Đ
        }Đ
        currentState.UpdateState(this);Đ
        //if navemeshagent is moving,set animator to moveĐ
        if (navMeshAgent.velocity.magnitude > 0)Đ
        {Đ
            animator.SetFloat("Speed", 1);Đ
Đ
        }Đ
        elseĐ
        {Đ
            animator.SetFloat("Speed", 0);Đ
        }Đ
Đ
    }Đ
Đ
    public void ChangeState(AIState newState)Đ
    {Đ
        currentState.ExitState(this);Đ
        currentState = newState;Đ
        newState.EnterState(this);Đ
    }Đ
public void checkForAggro()Đ
{Đ
    Collider[] colliders = Physics.OverlapSphere(transform.position, aggroRadius);Đ
    foreach (Collider collider in colliders)Đ
    {Đ
        if (!string.IsNullOrEmpty(collider.tag) && collider.CompareTag("Player"))Đ
        {Đ
            target = collider.gameObject.transform;Đ
            ChangeState(chaseState);Đ
            break;Đ
        }Đ
    }Đ
Đ
}Đ
    float nextAttackTime=0;Đ
public void attack(){Đ
Đ
            if (target != null)Đ
        {Đ
            float distanceToTarget = Vector3.Distance(transform.position,
target.position);Đ
Đ
            if (distanceToTarget <= attackRange)Đ
            {Đ
                navMeshAgent.isStopped=true;Đ
                GetComponent<Animator>().SetFloat("Speed", 0);Đ
                Đ
                    // Use attack abilityĐ

GetComponent<CharacterCombatController>().PerformAbility(attackAbility,
target.gameObject);Đ
                    Đ
                Đ
            }Đ
            elseĐ
            {Đ
                // Transition to another state if needed, for example, ChaseĐ
                ChangeState(chaseState);Đ
            }Đ
        }Đ
}Đ
Đ
    internal void SetAIController(AIController aiController)Đ
    {Đ
        currentState = aiController.currentState;Đ
        idleState = aiController.idleState;Đ
        followState = aiController.followState;Đ
        assistState = aiController.assistState;Đ
        patrolState = aiController.patrolState;Đ
        chaseState = aiController.chaseState;Đ
        attackState = aiController.attackState;Đ
        target = aiController.target;Đ
        aggroRadius = aiController.aggroRadius;Đ
        aggroTag = aiController.aggroTag;Đ
        attackInterval = aiController.attackInterval;Đ
        attackAbility = aiController.attackAbility;Đ
        attackRange = aiController.attackRange;Đ
Đ
Đ
    }Đ
}Đ
Đ
public abstract class AIState : ScriptableObjectĐ
{Đ
    public abstract void EnterState(AIController aiController);Đ
    public abstract void UpdateState(AIController aiController);Đ
    public abstract void ExitState(AIController aiController);Đ
}Đ
Đ
```

- AssistState at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\AiStates\AssistState.cs:
Code of file AssistState:

```
þÿusing UnityEngine;Đ
Đ
[CreateAssetMenu(menuName = "AI/States/AssistState")]Đ
public class AssistState : AIStateĐ
{Đ
    public Transform target;Đ
    public Ability assistAbility;Đ
    public float assistRange = 10f;Đ
Đ
    public override void EnterState(AIController aiController)Đ
    {Đ
    }Đ
Đ
    public override void UpdateState(AIController aiController)Đ
    {Đ
        float distanceToTarget = Vector3.Distance(aiController.transform.position, 
target.position);Đ
Đ
        if (distanceToTarget <= assistRange)Đ
        {Đ
            // aiController.GetComponent<AbilityController>().UseAbility(assistAbility);Đ
        }Đ
        elseĐ
        {Đ
            aiController.ChangeState(aiController.followState);Đ
        }Đ
    }Đ
Đ
    public override void ExitState(AIController aiController)Đ
    {Đ
    }Đ
}Đ
```

- AttackState at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\AiStates\AttackState.cs:
Code of file AttackState:

```
þÿusing UnityEngine;Đ
[CreateAssetMenu(menuName = "AI/States/AttackState")]Đ
public class AttackState : AIStateĐ
{Đ
    public Transform target;Đ
    public Ability attackAbility;Đ
    public float attackRange = 5f;Đ
    public float attackInterval = 1f;Đ
Đ
    private float nextAttackTime;Đ
Đ
    public override void EnterState(AIController aiController)Đ
    {Đ
        nextAttackTime = Time.time;Đ
    }Đ
Đ
```

```
    public override void UpdateState(AIController aiController)Đ
    {Đ
        aiController.attack();Đ
    }Đ
Đ
    public override void ExitState(AIController aiController)Đ
    {Đ
        // Clean up or reset any variables if neededĐ
    }Đ
}Đ
```

- ChaseState at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\AiStates\ChaseState.cs:
Code of file ChaseState:

```
þÿusing UnityEngine;Đ
[CreateAssetMenu(menuName = "AI/States/ChaseState")]Đ
public class ChaseState : AIStateĐ
{Đ
    public float chaseSpeed = 6f;Đ
    public float stoppingDistance = 5f;Đ
Đ
    public override void EnterState(AIController aiController)Đ
    {Đ
        aiController.GetComponent<UnityEngine.AI.NavMeshAgent>().speed =
chaseSpeed;Đ
    }Đ
Đ
    public override void UpdateState(AIController aiController)Đ
    {Đ
       Transform target = aiController.target;Đ
Đ
        if (target != null)Đ
        {Đ
            float distanceToTarget = Vector3.Distance(aiController.transform.position,
target.position);Đ
Đ
            if (distanceToTarget > stoppingDistance)Đ
            {Đ
                aiController.GetComponent<UnityEngine.AI.NavMeshAgent>().SetDestin
ation(target.position);Đ
            }Đ
            elseĐ
            {Đ
                // Transition to another state if needed, for example, AttackĐ
                aiController.ChangeState(aiController.attackState);Đ
            }Đ
        }Đ
    }Đ
Đ
    public override void ExitState(AIController aiController)Đ
    {Đ
        // Clean up or reset any variables if neededĐ
    }Đ
}Đ
```

- FollowState at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\AiStates\FollowState.cs:
Code of file FollowState:

```
þÿusing UnityEngine;Đ
Đ
[CreateAssetMenu(menuName = "AI/States/FollowState")]Đ
public class FollowState : AIStateĐ
{Đ
    public Transform target;Đ
    public float stoppingDistance = 2f;Đ
Đ
    public override void EnterState(AIController aiController)Đ
    {Đ
    }Đ
Đ
    public override void UpdateState(AIController aiController)Đ
    {Đ
        float distanceToTarget = Vector3.Distance(aiController.transform.position,
target.position);Đ
Đ
        if (distanceToTarget > stoppingDistance)Đ
        {Đ
            aiController.GetComponent<UnityEngine.AI.NavMeshAgent>().SetDestinati
on(target.position);Đ
        }Đ
        elseĐ
        {Đ
            aiController.GetComponent<UnityEngine.AI.NavMeshAgent>().ResetPath();Đ
        }Đ
    }Đ
Đ
    public override void ExitState(AIController aiController)Đ
    {Đ
    }Đ
}Đ
```

- IdleState at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\AiStates\IdleState.cs:
Code of file IdleState:

```
þÿusing UnityEngine;Ð
Ð
[CreateAssetMenu(menuName = "AI/States/IdleState")]Ð
public class IdleState : AIStateÐ
{Ð
    public float idleDuration = 3f;Ð
Ð
    private float idleTime;Ð
Ð
    public override void EnterState(AIController aiController)Ð
    {Ð
        idleTime = Time.time + idleDuration;Ð
    }Ð
Ð
    public override void UpdateState(AIController aiController)Ð
    {Ð
        if (Time.time > idleTime)Ð
        {Ð
            aiController.ChangeState(aiController.patrolState);Ð
        }Ð
    }Ð
Ð
    public override void ExitState(AIController aiController)Ð
    {Ð
    }Ð
}Ð
```

- PatrolState at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\AiStates\PatrolState.cs:
Code of file PatrolState:

```
þÿusing System.Collections.Generic;Ð
using UnityEngine;Ð
[CreateAssetMenu(menuName = "AI/States/PatrolState")]Ð
public class PatrolState : AIStateÐ
{Ð
    public List<Transform> waypoints;Ð
    public float patrolSpeed = 3f;Ð
    public float waitTime = 3f;Ð
Ð
    private int currentWaypoint;Ð
    private float waitEndTime;Ð
Đ
    public override void EnterState(AIController aiController)Đ
    {Đ
        aiController.GetComponent<UnityEngine.AI.NavMeshAgent>().speed = 
patrolSpeed;Đ
        currentWaypoint = 0;Đ
    }Đ
Đ
    public override void UpdateState(AIController aiController)Đ
    {   Đ
        aiController.checkForAggro();Đ
Đ
        UnityEngine.AI.NavMeshAgent agent = 
aiController.GetComponent<UnityEngine.AI.NavMeshAgent>();Đ
Đ
        if (waypoints.Count > 0)Đ
        {Đ
            if (!agent.pathPending && agent.remainingDistance < 0.5f)Đ
            {Đ
                if (Time.time > waitEndTime)Đ
                {Đ
                    currentWaypoint = (currentWaypoint + 1) % waypoints.Count;Đ
                    agent.SetDestination(waypoints[currentWaypoint].position);Đ
                    waitEndTime = Time.time + waitTime;Đ
                }Đ
            }Đ
        }Đ
    }Đ
Đ
    public override void ExitState(AIController aiController)Đ
    {Đ
        // Clean up or reset any variables if neededĐ
    }Đ
}
```
- PatrolStateMonoBehaviour at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\AiStates\PatrolStateMonoBehaviour.cs:
Code of file PatrolStateMonoBehaviour:
```
þÿusing UnityEngine;Đ
Đ
public class PatrolStateMonoBehaviour : MonoBehaviourĐ
{Đ
    public GameObject[] waypoints;Đ
}Đ
```

- AnimationController at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\AnimationController.cs:
Code of file AnimationController:

```
þÿusing System.Collections.Generic;Đ
using UnityEngine;Đ
Đ
public class AnimationController : MonoBehaviourĐ
{Đ
    [SerializeField] private Animator animator;Đ
    [SerializeField] private List<VisualEffectData> visualEffectDataList;Đ
Đ
    [SerializeField] private List<(string, float)> animationCastDelays;Đ
    public const string attackAnimationName = "attack";Đ
    public const string OneHandSwordLightAttack1AnimationName = 
"1HandSwordLightAttack1";Đ
    public const string OneHandSwordLightAttack2AnimationName = 
"1HandSwordLightAttack2";Đ
    public const string OneHandSwordLightAttack3AnimationName = 
"1HandSwordLightAttack3";Đ
Đ
    public const string idleAnimationName = "idle";Đ
Đ
Đ
    private Dictionary<string, GameObject> visualEffects;Đ
Đ
    private void Awake()Đ
    {Đ
        // Initialize the visualEffects dictionary.Đ
        initAnimationDelays();Đ
        visualEffects = new Dictionary<string, GameObject>();Đ
        foreach (VisualEffectData effectData in visualEffectDataList)Đ
        {Đ
            visualEffects.Add(effectData.name, effectData.visualEffectPrefab);Đ
        }Đ
    }Đ
Đ
    public void PlayAnimation(string animationName)Đ
    {Đ
        // Play the specified animation.Đ
        if (animationName == "attack")Đ
        {Đ
            animator.SetTrigger("attack");Đ
            return;Đ
        }Đ
Đ
        animator.Play(animationName);Đ
    }Đ
Đ
    public void ApplyVisualEffect(string effectName, Vector3 position, Quaternion
rotation)Đ
    {Đ
        // Instantiate the specified visual effect at the given position and rotation.Đ
        if (visualEffects.TryGetValue(effectName, out GameObject effectPrefab))Đ
        {Đ
            Instantiate(effectPrefab, position, rotation);Đ
        }Đ
        elseĐ
        {Đ
            Debug.LogWarning($"Visual effect '{effectName}' not found.");Đ
        }Đ
    }Đ
    public void initAnimationDelays()Đ
    {Đ
        animationCastDelays = new List<(string, float)>();Đ
        animationCastDelays.Add(("attack", 0.11f));Đ
        animationCastDelays.Add(("1HandSwordLightAttack1", 0.11f));Đ
        animationCastDelays.Add(("1HandSwordLightAttack2", 0.07f));Đ
        animationCastDelays.Add(("1HandSwordLightAttack3", 0.06f));Đ
Đ
    }Đ
    public float returnAnimationDelay(string animationName)Đ
    {Đ
        foreach ((string, float) paar in animationCastDelays)Đ
        {Đ
            if (paar.Item1 == animationName)Đ
            {Đ
                return paar.Item2;Đ
            }Đ
        }Đ
        return 0;Đ
    }Đ
    Đ
}Đ
Đ
[System.Serializable]Đ
public class VisualEffectDataĐ
{Đ
    public string name;Đ
    public GameObject visualEffectPrefab;Đ
```

```
}Ð

- BuffSystem at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My
project\Assets\Scripts\PlayerAndUnitsComponent\BuffSystem.cs:
Code of file BuffSystem:
þÿusing System.Collections.Generic;Ð
using UnityEngine;Ð
Ð
public class BuffSystem : MonoBehaviourÐ
{Ð
    public Dictionary<string, BuffInstance> activeBuffs;Ð
    public List <string> buffsToRemove;Ð
    private Dictionary<string, System.Action<BuffInstance>> eventHandlers;Ð
Ð
    public StatsModifier TotalstatsModifier;Ð
     Ð
Ð
    private void Awake()Ð
    {Ð
        buffsToRemove = new List<string>();Ð
        activeBuffs = new Dictionary<string, BuffInstance>();Ð
        eventHandlers = new Dictionary<string, System.Action<BuffInstance>>();Ð
    }Ð
Ð
    private void Update()Ð
    {Ð
        foreach (BuffInstance buffInstance in activeBuffs.Values)Ð
        {Ð
            buffInstance.Update();Ð
        }Ð
        removeBuffs();Ð
    }Ð
    private void removeBuffs(){Ð
        foreach (string buffName in buffsToRemove)Ð
        {Ð
            BuffInstance buffInstance = activeBuffs[buffName];Ð
            RemoveEventHandlers(buffInstance.buff);Ð
            activeBuffs.Remove(buffName);Ð
        }Ð
        buffsToRemove.Clear();Ð
    }Ð
    public void AddBuff(Buff buff, GameObject target)Ð
    {Ð
        if(buff==null){Ð
            Debug.LogError("buff is null");Ð
            return;Ð
        }Ð
        if (activeBuffs.ContainsKey(buff.buffName))Ð
        {Ð
            BuffInstance existingBuff = activeBuffs[buff.buffName];Ð
Ð
            if (buff.stackable && existingBuff.currentStacks < buff.maxStacks)Ð
            {Ð
                existingBuff.AddStack();Ð
                existingBuff.Refresh(buff.duration);Ð
            }Ð
            elseÐ
            {Ð
                existingBuff.Refresh(buff.duration);Ð
            }Ð
        }Ð
        elseÐ
        {Ð
            BuffInstance newBuff = new BuffInstance(buff, target, 1, buff.duration);Ð
            activeBuffs.Add(buff.buffName, newBuff);Ð
            AddEventHandlers(buff);Ð
            newBuff.OnBuffApply();Ð
        }Ð
    }Ð
Ð
    public void RemoveBuff(Buff buff)Ð
    {Ð
        buffsToRemove.Add(buff.buffName);Ð
Ð
    }Ð
Ð
    public BuffInstance GetBuffInstance(string buffName)Ð
    {Ð
        if (activeBuffs.ContainsKey(buffName))Ð
        {Ð
            return activeBuffs[buffName];Ð
        }Ð
        return null;Ð
    }Ð
Ð
    private void AddEventHandlers(Buff buff)Ð
    {Ð
        List<string> eventTypes = buff.GetEventTypes();Ð
Ð
        foreach (string eventType in eventTypes)Ð
        {Đ
            if (!eventHandlers.ContainsKey(eventType))Đ
            {Đ
                eventHandlers.Add(eventType, (BuffInstance buffInstance) => { });Đ
            }Đ
Đ
            System.Action<BuffInstance> eventHandler = null;Đ
            switch (eventType)Đ
            {Đ
                case "OnApply":Đ
                    eventHandler = buff.InvokeOnApply;Đ
                    break;Đ
                case "OnFade":Đ
                    eventHandler = buff.InvokeOnFade;Đ
                    break;Đ
                case "OnHit":Đ
                    eventHandler = buff.InvokeOnHit;Đ
                    break;Đ
            }Đ
Đ
            if (eventHandler != null)Đ
            {Đ
                eventHandlers[eventType] += eventHandler;Đ
            }Đ
        }Đ
    }Đ
Đ
    private void RemoveEventHandlers(Buff buff)Đ
    {Đ
        List<string> eventTypes = buff.GetEventTypes();Đ
Đ
        foreach (string eventType in eventTypes)Đ
        {Đ
            System.Action<BuffInstance> eventHandler = null;Đ
            switch (eventType)Đ
            {Đ
                case "OnApply":Đ
                    eventHandler = buff.InvokeOnApply;Đ
                    break;Đ
                case "OnFade":Đ
                    eventHandler = buff.InvokeOnFade;Đ
                    break;Đ
                case "OnHit":Đ
                    eventHandler = buff.InvokeOnHit;Đ
                    break;Đ
            }Đ
Đ
            if (eventHandler != null)Đ
            {Đ
                eventHandlers[eventType] -= eventHandler;Đ
            }Đ
        }Đ
    }Đ
Đ
    public void CallEventHandlers(string eventType, BuffInstance buffInstance)Đ
    {Đ
        if (eventHandlers.ContainsKey(eventType))Đ
        {Đ
            eventHandlers[eventType]?.Invoke(buffInstance);Đ
        }Đ
    }Đ
}Đ
```

- ButtonWithToolTip at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\ButtonWithToolTip.cs:
Code of file ButtonWithToolTip:

```
þÿusing UnityEngine;Đ
using UnityEngine.EventSystems;Đ
using UnityEngine.UI;Đ
Đ
public class ButtonWithToolTip : MonoBehaviour, IPointerEnterHandler,
IPointerExitHandlerĐ
{Đ
    public SkillNode skillNode;Đ
    private PlayerController playerController;Đ
    public UIManager uiManager;Đ
    private GameObject toolTipObject;Đ
Đ
    private Button button;Đ
Đ
    private void Awake()Đ
    {Đ
        playerController = FindObjectOfType<PlayerController>();Đ
        uiManager = FindObjectOfType<UIManager>();Đ
        toolTipObject = uiManager.tooltip;Đ
        toolTipObject.SetActive(false);Đ
Đ
        button = GetComponent<Button>();Đ
        button.onClick.AddListener(TryLearn);Đ
        if(skillNode!=null){Đ
```

```
            GetComponent<Image>().sprite = skillNode.icon;Ð
        }Ð
    }Ð
Ð
    public void OnPointerEnter(PointerEventData eventData)Ð
    {Ð
        ShowToolTip();Ð
    }Ð
Ð
    public void OnPointerExit(PointerEventData eventData)Ð
    {Ð
        HideToolTip();Ð
    }Ð
Ð
    private void ShowToolTip()Ð
    {Ð
        uiManager.OpenToolTip(skillNode,
gameObject.GetComponent<RectTransform>().position);Ð
    }Ð
Ð
    private void HideToolTip()Ð
    {Ð
        uiManager.CloseToolTip();Ð
    }Ð
Ð
    private void TryLearn()Ð
    {Ð
        playerController.TryUnlockSkillNode(skillNode);Ð
    }Ð
}Ð
```

- CharacterCombatController at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\CharacterCombatController.cs:
Code of file CharacterCombatController:

```
þÿusing System.Collections;Ð
using System.Collections.Generic;Ð
using UnityEngine;Ð
public class CharacterCombatController : MonoBehaviour, IStatsProviderÐ
{Ð
    public CharacterStats characterStats;Ð
    public AbilityController abilityController;Ð
    public AnimationController animationController;Ð
    public IStunnable stunnable;Ð
    public ComboController comboController;Ð
```

```
Đ
Đ
    private void Start()Đ
    {Đ
        characterStats = GetComponent<CharacterStats>();Đ
        stunnable = GetComponent<IStunnable>();Đ
        abilityController = GetComponent<AbilityController>();Đ
        animationController = GetComponent<AnimationController>();Đ
        comboController = new ComboController();Đ
    }Đ
Đ
    public void PerformAbility(Ability ability, GameObject target)Đ
    {Đ
        if(stunnable.isStunned())Đ
        {Đ
            return;Đ
        }Đ
        if(abilityController.checkCooldown(ability.name,ability.cooldown)==false)Đ
        {Đ
            return;Đ
        }Đ
        PlayerController playerController = GetComponent<PlayerController>();Đ
        if (playerController != null)Đ
        {Đ
           playerController.faceIndirectionOfCamera();Đ
        }Đ
        float damageAbility = ability.baseDamage + (ability.strengthScaling * 
characterStats.strength) + (ability.intelligenceScaling * 
characterStats.intelligence);Đ
        float critChance = characterStats.criticalChance;Đ
        if (Random.Range(0f, 1f) <= critChance)Đ
        {Đ
            damageAbility *= 2;Đ
        }Đ
Đ
Đ
        AbilityData abilityData = new AbilityDataĐ
        {Đ
            CasterStats = characterStats,Đ
           Target = target,Đ
            damage = damageAbility,Đ
            CasterController = abilityController,Đ
            CasterCombatController = thisĐ
            // ... other fieldsĐ
        };Đ
        abilityController.setCooldown(ability.name,ability.cooldown);Đ
        comboController.UpdateComboController();Đ
        abilityController.CastAbility(ability, abilityData);Đ
    }Đ
    public CharacterStats GetCharacterStats()Đ
    {Đ
        return characterStats;Đ
    }Đ
Đ
}Đ
```

- CharacterStats at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\CharacterStats.cs:
Code of file CharacterStats:

```
þÿusing System;Đ
using System.Collections;Đ
using UnityEngine;Đ
[Serializable]Đ
public class CharacterStats : MonoBehaviourĐ
{Đ
    // MainStatsĐ
    public float strength;Đ
    public float intelligence;Đ
    public float dexterity;Đ
    public float endurance;Đ
    public float wisdom;Đ
Đ
    // SubStatsĐ
Đ
    public float criticalChance;Đ
    public float criticalDamage;Đ
    public float attackSpeed;Đ
Đ
    public float spellCriticalChance;Đ
    public float spellCriticalDamage;Đ
Đ
    public float cooldown;Đ
Đ
    public float maxLife;Đ
    public float maxMana;Đ
    public float lifeRegen;Đ
    public float manaRegen;Đ
Đ
    public float armor;Đ
    public float magicResistance;Đ
```

```
Đ
    public float dodgeChance;Đ
Đ
    public int unspentStatPoints;Đ
Đ
    public event Action StatsChanged;Đ
    private EquipManager equipManager;Đ
    private SkillController skillController;Đ
    private BuffSystem buffSystem;Đ
Đ
    private void Awake()Đ
    {Đ
        buffSystem = GetComponent<BuffSystem>();Đ
        equipManager = GetComponent<EquipManager>();Đ
        skillController = GetComponent<SkillController>();Đ
    }Đ
    private void Start()Đ
    {Đ
        // Initialize unspentStatPoints or load from saved game dataĐ
        unspentStatPoints = 10;Đ
        StartCoroutine(InitializeCharacterStats());        Đ
    }Đ
    private IEnumerator InitializeCharacterStats()Đ
    {Đ
        yield return new WaitUntil(() => equipManager != null);Đ
        UpdateSubStats();Đ
Đ
                Đ
         HealthController healthController = GetComponent<HealthController>();Đ
         if(healthController != null){Đ
                healthController.updateHealth();Đ
         }Đ
         ManaController manaController = GetComponent<ManaController>();Đ
            if(manaController != null){Đ
                    manaController.updateMana();Đ
            }Đ
        Đ
    }Đ
    public void AddStatPoints(int amount)Đ
    {Đ
        unspentStatPoints += amount;Đ
        StatsChanged?.Invoke();Đ
    }Đ
Đ
    public void UpdateSubStats()Đ
    {Ð
Ð
        strength += equipManager.TotalStrength;Ð
        intelligence += equipManager.TotalIntelligence;Ð
        dexterity += equipManager.TotalDexterity;Ð
        endurance += equipManager.TotalEndurance;Ð
        wisdom += equipManager.TotalWisdom;Ð
Ð
Ð
        criticalChance = 0.02f * dexterity;Ð
        criticalDamage = 1.5f + (0.14f * dexterity);Ð
        attackSpeed = 1 + (0.01f * strength * dexterity);Ð
Ð
        spellCriticalChance = 0.02f * intelligence;Ð
        spellCriticalDamage = 1.5f + (0.14f * intelligence);Ð
Ð
        armor = 1.5f * endurance;Ð
        magicResistance = 1.5f * endurance;Ð
Ð
Ð
        // Calculate substats based on main stats + equipment bonuses.Ð
        maxLife = 100 + 20 * endurance;Ð
        maxMana = 100 + 20 * wisdom;Ð
        lifeRegen = 1 + 0.25f * endurance;Ð
        manaRegen = 0.5f + 0.25f * wisdom;Ð
Ð
        dodgeChance = 0.009f * dexterity;Ð
Ð
        AddStatBonuses(equipManager.TotalStatModier);Ð
        AddStatBonuses(skillController.totalStatsModier);Ð
        AddStatBonuses(buffSystem.TotalstatsModifier);Ð
Ð
Ð
        StatsChanged?.Invoke();Ð
    }Ð
Ð
    public void AddStatBonuses(StatsModifier statModifier)Ð
    {Ð
Ð
        attackSpeed += statModifier.attackSpeed;Ð
        criticalChance += statModifier.criticalChance;Ð
        criticalDamage += statModifier.criticalDamage;Ð
        spellCriticalChance += statModifier.spellCriticalChance;Ð
        spellCriticalDamage += statModifier.spellCriticalDamage;Ð
        cooldown += statModifier.cooldown;Ð
        dodgeChance += statModifier.dodgeChance;Đ
        armor += statModifier.armor;Đ
        magicResistance += statModifier.magicResistance;Đ
        maxLife += statModifier.maxLife;Đ
        maxMana += statModifier.maxMana;Đ
        lifeRegen += statModifier.lifeRegen;Đ
        manaRegen += statModifier.manaRegen;Đ
    }Đ
    public void RemoveStatBonuses(StatsModifier statModifier)Đ
    {Đ
        attackSpeed -= statModifier.attackSpeed;Đ
        criticalChance -= statModifier.criticalChance;Đ
        criticalDamage -= statModifier.criticalDamage;Đ
        spellCriticalChance -= statModifier.spellCriticalChance;Đ
        spellCriticalDamage -= statModifier.spellCriticalDamage;Đ
        cooldown -= statModifier.cooldown;Đ
        dodgeChance -= statModifier.dodgeChance;Đ
        armor -= statModifier.armor;Đ
        magicResistance -= statModifier.magicResistance;Đ
        maxLife -= statModifier.maxLife;Đ
        maxMana -= statModifier.maxMana;Đ
        lifeRegen -= statModifier.lifeRegen;Đ
        manaRegen -= statModifier.manaRegen;Đ
    }Đ
Đ
Đ
    public void IncreaseStat(Archetype stateType, int amount)Đ
    {Đ
        if (unspentStatPoints >= amount)Đ
        {Đ
            switch (stateType)Đ
            {Đ
                case Archetype.Strength:Đ
                    strength += amount;Đ
                    break;Đ
                case Archetype.Intelligence:Đ
                    intelligence += amount;Đ
                    break;Đ
                case Archetype.Dexterity:Đ
                    dexterity += amount;Đ
                    break;Đ
                case Archetype.Endurance:Đ
                    endurance += amount;Đ
                    break;Đ
                case Archetype.Wisdom:Đ
                wisdom += amount;Đ
                break;Đ
            default:Đ
                Debug.LogWarning("Invalid stat name.");Đ
                return;Đ
        }Đ
Đ
        unspentStatPoints -= amount;Đ
        UpdateSubStats();Đ
    }Đ
    elseĐ
    {Đ
        Debug.LogWarning("Not enough stat points.");Đ
    }Đ
}Đ
Đ
internal void SetStats(CharacterStats stats)Đ
{Đ
    strength = stats.strength;Đ
    intelligence = stats.intelligence;Đ
    dexterity = stats.dexterity;Đ
    endurance = stats.endurance;Đ
    wisdom = stats.wisdom;Đ
    equipManager = GetComponent<EquipManager>();Đ
    UpdateSubStats();Đ
}Đ
}Đ
```

- ComboController at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\ComboController.cs:
Code of file ComboController:

```
using System;Đ
using System.Collections.Generic;Đ
using UnityEngine;Đ
[System.Serializable]Đ
public class ComboController{Đ
    public List<ComboCounter> comboCounterList;Đ
Đ
    public ComboController(){Đ
        comboCounterList = new List<ComboCounter>();Đ
    }Đ
    public void UpdateComboController(){Đ
        foreach (ComboCounter comboCounter in comboCounterList)Đ
        {Đ
            comboCounter.UpdateComboCounter();Đ
        }Đ
    }Đ
    public void IncreaseComboCounter(string comboName){Đ
        bool found = false;Đ
        foreach (ComboCounter comboCounter in comboCounterList)Đ
        {Đ
            if(comboCounter.ComboName == comboName){Đ
                comboCounter.IncreaseComboCounter();Đ
                found = true;Đ
            }Đ
        }Đ
        if(!found){Đ
            comboCounterList.Add(new ComboCounter(1f,comboName));Đ
        }Đ
    }Đ
    public int GetComboCounter(string comboName){Đ
        foreach (ComboCounter comboCounter in comboCounterList)Đ
        {Đ
            if(comboCounter.ComboName == comboName){Đ
                return comboCounter.GetComboCounter();Đ
            }Đ
        }Đ
        return 0;Đ
    }Đ
Đ
    internal void ResetComboCounter(string comboName)Đ
    {Đ
        foreach (ComboCounter comboCounter in comboCounterList)Đ
        {Đ
            if(comboCounter.ComboName == comboName){Đ
                comboCounter.ResetComboCounter();Đ
            }Đ
        }Đ
    }Đ
}Đ
[System.Serializable]Đ
public class ComboCounter{Đ
    Đ
    public string ComboName;Đ
    public int comboCounter;Đ
    public float comboTimer;Đ
    public float comboTimeLimit;Đ
    public ComboCounter(float comboTimeLimit,string comboName){Đ
        this.comboTimeLimit = comboTimeLimit;Đ
        comboCounter = 0;Đ
        comboTimer = 0;Ð
        ComboName = comboName;Ð
    }Ð
    public void UpdateComboCounter(){Ð
        comboTimer += Time.deltaTime;Ð
        if(comboTimer >= comboTimeLimit){Ð
            comboCounter = 0;Ð
        }Ð
    }Ð
    public void IncreaseComboCounter(){Ð
        UpdateComboCounter();Ð
        comboCounter++;Ð
        comboTimer = 0;Ð
    }Ð
    public int GetComboCounter(){Ð
        return comboCounter;Ð
    }Ð
    public void ResetComboCounter(){Ð
        comboCounter = 0;Ð
        comboTimer = 0;Ð
    }Ð
}
```

- EquipManager at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\EquipManager.cs:

Code of file EquipManager:

```
þÿusing System;Ð
using System.Collections.Generic;Ð
using UnityEngine;Ð
[Serializable]Ð
public class EquipManager : MonoBehaviourÐ
{Ð
    public enum EquipmentType { Weapon, Shield, Helmet, ChestArmor, LegArmor,
Boots, Ring, Wrist }Ð
Ð
    public Dictionary<EquipmentType, EquipableItem> equippedItems = new
Dictionary<EquipmentType, EquipableItem>();Ð
Ð
    // Properties to store the total stats from all equipped items.Ð
    public StatsModifier TotalStatModier;Ð
    public float TotalStrength = 0;Ð
    public float TotalIntelligence= 0;Ð
    public float TotalDexterity= 0;Ð
    public float TotalEndurance= 0;Ð
    public float TotalWisdom=0 ;Ð
    // Add more stat properties as needed.Ð
```

```
Đ
    public void EquipItem(EquipmentType type, EquipableItem item)Đ
    {Đ
        if (equippedItems.ContainsKey(type))Đ
        {Đ
            UnequipItem(type);Đ
        }Đ
Đ
        equippedItems[type] = item;Đ
        ApplyItemStats(item);Đ
    }Đ
Đ
    public void UnequipItem(EquipmentType type)Đ
    {Đ
        if (!equippedItems.ContainsKey(type)) return;Đ
Đ
        EquipableItem item = equippedItems[type];Đ
        RemoveItemStats(item);Đ
        equippedItems.Remove(type);Đ
    }Đ
Đ
    private void ApplyItemStats(EquipableItem item)Đ
    {Đ
        TotalStrength += item.strengthBonus;Đ
        TotalIntelligence += item.intelligenceBonus;Đ
        TotalDexterity += item.dexterityBonus;Đ
        TotalEndurance += item.enduranceBonus;Đ
        TotalWisdom += item.wisdomBonus;Đ
Đ
        TotalStatModier.Add(item.subStatsModifier);Đ
Đ
        // Add more stat effects as needed.Đ
    }Đ
Đ
    private void RemoveItemStats(EquipableItem item)Đ
    {Đ
        TotalStrength -= item.strengthBonus;Đ
        TotalIntelligence -= item.intelligenceBonus;Đ
        TotalDexterity -= item.dexterityBonus;Đ
        TotalEndurance -= item.enduranceBonus;Đ
        TotalWisdom -= item.wisdomBonus;Đ
Đ
        TotalStatModier.Sub(item.subStatsModifier);Đ
Đ
        // Remove more stat effects as needed.Đ
    }Ð
Ð
    internal void SetEquipManager(EquipManager equipManager)Ð
    {Ð
        equippedItems = equipManager.equippedItems;Ð
        TotalStatModier = equipManager.TotalStatModier;Ð
        TotalStrength = equipManager.TotalStrength;Ð
        TotalIntelligence = equipManager.TotalIntelligence;Ð
        TotalDexterity = equipManager.TotalDexterity;Ð
        TotalEndurance = equipManager.TotalEndurance;Ð
        TotalWisdom = equipManager.TotalWisdom;Ð
        Ð
Ð
    }Ð
}Ð
```

- ExperienceSystem at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\ExperienceSystem.cs:
Code of file ExperienceSystem:

```
þÿusing System;Ð
Ð
public class ExperienceSystemÐ
{Ð
    public int CurrentXP { get; private set; }Ð
    public int Level { get; private set; }Ð
    public int XpToNextLevel { get; private set; }Ð
Ð
    public event Action LevelUpEvent;Ð
    public event Action<int> ExperienceGained;Ð
Ð
    public ExperienceSystem()Ð
    {Ð
        CurrentXP = 0;Ð
        Level = 1;Ð
        UpdateXpToNextLevel();Ð
    }Ð
Ð
    public void AddExperience(int amount)Ð
    {Ð
        CurrentXP += amount;Ð
        ExperienceGained?.Invoke(amount);Ð
Ð
        while (CurrentXP >= XpToNextLevel)Ð
        {Ð
            CurrentXP -= XpToNextLevel;Ð
            LevelUp();Ð
        }Ð
    }Ð
Ð
    private void LevelUp()Ð
    {Ð
        Level++;Ð
        UpdateXpToNextLevel();Ð
        LevelUpEvent?.Invoke();Ð
    }Ð
Ð
    private void UpdateXpToNextLevel()Ð
    {Ð
        XpToNextLevel = CalculateXpForLevel(Level);Ð
    }Ð
Ð
    private int CalculateXpForLevel(int level)Ð
    {Ð
        // Implement your custom XP calculation logic hereÐ
        return (int)Math.Floor(Math.Pow(level, 2) * 100);Ð
    }Ð
}Ð
```

- HealthController at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\HealthController.cs:
Code of file HealthController:

```
using UnityEngine;Ð
Ð
public class HealthController : MonoBehaviourÐ
{Ð
    private CharacterStats characterStats;Ð
    public string Name;Ð
    public float maxHealth;Ð
    public float currentHealth;Ð
    public GameObject damageTextPrefab;Ð
    private QuestSystem questSystem;Ð
Ð
    void UpdateMaxHealth()Ð
    {Ð
        maxHealth = characterStats.maxLife;Ð
    }Ð
    public void updateHealth()Ð
    {Ð
        currentHealth = maxHealth;Ð
    }Ð
    private void Start()Đ
    {Đ
        characterStats = GetComponent<CharacterStats>();Đ
        characterStats.StatsChanged+=UpdateMaxHealth;Đ
        UpdateMaxHealth();Đ
        currentHealth = maxHealth;Đ
        damageTextPrefab = GameObject.Find("DamageTextTemplate");Đ
        Đ
Đ
    }Đ
Đ
    public void TakeDamage(float damage,GameObject from)Đ
    {Đ
        currentHealth -= damage;Đ
        ShowDamageNumbers(damage);Đ
        if (currentHealth <= 0)Đ
        {Đ
            if(from.GetComponent<QuestSystem>() != null)Đ
            {Đ

from.GetComponent<QuestSystem>().UpdateQuestObjective("kill:"+Name);Đ
            }Đ
            Die();Đ
        }Đ
    }Đ
Đ
    private void Die()Đ
    {Đ
        // Implement death behavior, such as playing death animation, dropping loot,
etc.Đ
Đ
        Destroy(gameObject);Đ
    }Đ
    public void ShowDamageNumbers(float damage)Đ
    {Đ
        if (WorldSpaceCanvasController.Instance == null)Đ
        {Đ
            Debug.LogError("WorldSpaceCanvasController instance is not present in
the scene.");Đ
            return;Đ
        }Đ
Đ
        WorldSpaceCanvasController.Instance.SpawnDamageNumber(damage,
transform.position + Vector3.up * 2f);Đ
    }Đ
```

```
Đ
}
- HotkeyController at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My
project\Assets\Scripts\PlayerAndUnitsComponent\HotkeyController.cs:
Code of file HotkeyController:
using UnityEngine;Đ
using System.Collections.Generic;Đ
using System;Đ
Đ
public class HotkeyController : MonoBehaviour{Đ
   Đ
   public List<Hotkey> hotkeys;Đ
   private CharacterCombatController combatController;Đ
   public Dictionary<KeyCode, Hotkey> hotkeyMapping;Đ
   public void Update()Đ
   {Đ
      HandleHotkey();Đ
   }Đ
   public void Start()Đ
   {Đ
      combatController = GetComponent<CharacterCombatController>();Đ
      hotkeys = new List<Hotkey>();Đ
         for (int i = 0; i < 9; i++)Đ
         {Đ
            hotkeys.Add(new Hotkey());Đ
         }Đ
Đ
      hotkeyMapping = new Dictionary<KeyCode, Hotkey>Đ
      {Đ
         { KeyCode.Alpha1, hotkeys[0] },Đ
         { KeyCode.Alpha2, hotkeys[1] },Đ
         { KeyCode.Alpha3, hotkeys[2] },Đ
         { KeyCode.Alpha4, hotkeys[3] },Đ
         { KeyCode.Alpha5, hotkeys[4] },Đ
         { KeyCode.Alpha6, hotkeys[5] },Đ
         { KeyCode.Alpha7, hotkeys[6] },Đ
         { KeyCode.Alpha8, hotkeys[7] },Đ
         { KeyCode.E, hotkeys[8] }Đ
      };Đ
Đ
      Hotkey hotkeyTest = new Hotkey();Đ
      hotkeyTest.ability = combatController.abilityController.learnedAbilitys[0];Đ
      hotkeys[0].ability = combatController.abilityController.learnedAbilitys[0];Đ
   }Đ
   private void HandleHotkey()Đ
```

```
    {Ð
        foreach (KeyValuePair<KeyCode, Hotkey> entry in hotkeyMapping)Ð
        {Ð
            if (Input.GetKeyDown(entry.Key))Ð
            {Ð
                Hotkey hotkey = entry.Value;Ð
                if (hotkey.ability != null)Ð
                {Ð
                    combatController.PerformAbility(hotkey.ability, this.gameObject);Ð
                }Ð
                // else if (hotkey.item != null)Ð
                {Ð
                    // UseItem(hotkey.item);Ð
                }Ð
            }Ð
        }Ð
    }Ð
Ð
internal void SwapHotkeys(int hotkeyIndex1, int hotkeyIndex2)Ð
{Ð
    Hotkey tempHotkey = hotkeys[hotkeyIndex1];Ð
    hotkeys[hotkeyIndex1] = hotkeys[hotkeyIndex2];Ð
    hotkeys[hotkeyIndex2] = tempHotkey;Ð
}Ð
Ð
internal void AssignAbilityToHotkey(int hotkeyIndex, Ability assignedAbility)Ð
{Ð
    hotkeys[hotkeyIndex].ability = assignedAbility;Ð
    hotkeys[hotkeyIndex].item = null;Ð
}Ð
Ð
internal void AssignItemToHotkey(int hotkeyIndex, Item assignedItem)Ð
{Ð
    hotkeys[hotkeyIndex].item = assignedItem;Ð
    hotkeys[hotkeyIndex].ability = null;Ð
}Ð
Ð
}Ð
public class HotkeyÐ
{Ð
    public Ability ability;Ð
    public Item item;Ð
}
```
- IInteractable at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\IInteractable.cs:

```
Code of file IInteractable:
using UnityEngine;Đ
Đ
public interface IInteractableĐ
{Đ
    void Interact(Transform interacter);Đ
}Đ

- Inventory at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My
project\Assets\Scripts\PlayerAndUnitsComponent\Inventory.cs:
Code of file Inventory:
þÿusing System.Collections.Generic;Đ
using UnityEngine;Đ
Đ
public class Inventory : MonoBehaviourĐ
{Đ
    public List<Item> items;Đ
    public QuestSystem questSystem;Đ
      private void Start()Đ
    {Đ
        questSystem = GetComponent<QuestSystem>();Đ
Đ
    }Đ
Đ
    public void AddItem(Item item)Đ
    {Đ
        items.Add(item);Đ
        if(questSystem!=null)  // Check if questSystem is not nullĐ
        {Đ
            Debug.Log("collect:"+item.name);Đ
            questSystem.UpdateQuestObjective("collect:"+item.name); // Call
UpdateQuestObjective method with item idĐ
        }Đ
    }Đ
Đ
    public void RemoveItem(Item item)Đ
    {Đ
        items.Remove(item);Đ
    }Đ
Đ
    public bool HasItem(Item item)Đ
    {Đ
        return items.Contains(item);Đ
    }Đ
}Đ
```

- isStunnableController at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\isStunnableController.cs:
Code of file isStunnableController:

```
using UnityEngine;Ð
public class isStunnableController : MonoBehaviour,IStunnable{Ð
Ð
    public bool stunned;Ð
Ð
    float timeAtStunStart;Ð
    float stunDuration;Ð
Ð
    bool IStunnable.stunned { get => stunned ;}Ð
Ð
    float IStunnable.timeAtStunStart => timeAtStunStart;Ð
Ð
    float IStunnable.stunDuration => stunDuration;Ð
Ð
   VisualEffectController visualEffectController;Ð
Ð
    private void Start()Ð
    {Ð
        visualEffectController = GetComponent<VisualEffectController>();Ð
    }Ð
    public void Stun(float duration){Ð
        stunned = true;Ð
        timeAtStunStart = Time.time;Ð
        stunDuration = duration;Ð
        visualEffectController.SpawnEffect("Stun",duration);Ð
Ð
    }Ð
    public bool isStunned(){Ð
        Ð
            if(Time.time - timeAtStunStart >= stunDuration){Ð
                stunned = false;Ð
            }Ð
        return stunned;Ð
        Ð
    }Ð
}
```

- IStunnable at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\IStunnable.cs:
Code of file IStunnable:

```
//Interface isStunnable if GameObject can be stunned,contain bool isStunnedÐ
```

```
// Path: Assets\Scripts\PlayerAndUnitsComponent\IStunnable.csÐ
using UnityEngine;Ð
public interface IStunnableÐ
{Ð
    bool stunned { get; }Ð
Ð
   Ð
    float timeAtStunStart{ get; }Ð
    float stunDuration{ get ;}Ð
Ð
Ð
    void Stun(float duration);Ð
    public bool isStunned();Ð
}Ð
Ð
```

- ManaController at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\ManaController.cs:
Code of file ManaController:

```
using UnityEngine;Ð
Ð
public class ManaController : MonoBehaviourÐ
{Ð
    public float maxMana;Ð
    public float currentMana;Ð
    private CharacterStats characterStats;Ð
Ð
    private void Start()Ð
    {Ð
        characterStats = GetComponent<CharacterStats>();Ð
        characterStats.StatsChanged += updateMaxMana;Ð
        Ð
        currentMana = maxMana;Ð
Ð
    }Ð
    private void updateMaxMana()Ð
    {Ð
     maxMana = characterStats.maxMana;Ð
    }Ð
    public void updateMana()Ð
    {Ð
        currentMana = maxMana;Ð
    }Ð
    public void UseMana(float manaCost)Ð
    {Ð
```

```
        if (HasSufficientMana(manaCost))Ð
        {Ð
            currentMana -= manaCost;Ð
        }Ð
    }Ð
Ð
    public bool HasSufficientMana(float manaCost)Ð
    {Ð
        return currentMana >= manaCost;Ð
    }Ð
Ð
    public void RegenerateMana(float manaAmount)Ð
    {Ð
        currentMana += manaAmount;Ð
        if (currentMana > maxMana)Ð
        {Ð
            currentMana = maxMana;Ð
        }Ð
    }Ð
Ð
Ð
}Ð
```

- MovementController at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\MovementController.cs:
Code of file MovementController:

```
using UnityEngine;Ð
using UnityEngine.AI;Ð
Ð
public class MovementController : MonoBehaviourÐ
{Ð
    IStunnable stunnable;Ð
    public Transform target;Ð
    public float stoppingDistance = 2f;Ð
Ð
    private NavMeshAgent agent;Ð
Ð
    private void Start()Ð
    {Ð
        agent = GetComponent<NavMeshAgent>();Ð
        stunnable = GetComponent<IStunnable>();Ð
        Debug.Log("stunnable: " + stunnable);Ð
    }Ð
Ð
```

```
    private void Update()Đ
    {Đ
        if (target != null)Đ
        {Đ
            agent.SetDestination(target.position);Đ
            agent.stoppingDistance = stoppingDistance;Đ
        }Đ
        if(stunnable != null && stunnable.isStunned())Đ
        {Đ
            agent.isStopped = true;Đ
        }Đ
        elseĐ
        {Đ
            agent.isStopped = false;Đ
        }Đ
    }Đ
}Đ
```

- PlayerController at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\PlayerController.cs:
Code of file PlayerController:

```
using System.Collections.Generic;Đ
using UnityEngine;Đ
Đ
Đ
Đ
public class PlayerController : MonoBehaviourĐ
{Đ
    [Header("Controller")]Đ
    BuffSystem buffSystem;Đ
    ExperienceSystem experienceSystem;Đ
    CharacterStats characterStats;Đ
    CharacterCombatController combatController;Đ
    SkillController skillController;Đ
    SkillTree skillTree;Đ
    IStunnable stunnable;Đ
    HotkeyController hotkeyController;Đ
    CanGrabController canGrabController;Đ
   TargetingSystem targetingSystem;Đ
  Đ
Đ
Đ
    public Ability Ability1;Đ
Đ
Đ
```

```
Đ
    [Header("Movement")]Đ
    public float moveSpeed = 5f;Đ
    public float rotationSpeed = 720f;Đ
    public float jumpForce = 1f;Đ
    public LayerMask groundLayer;Đ
Đ
    [Header("Camera")]Đ
    public Transform cameraTarget;Đ
    public float cameraDistance = 5f;Đ
    public float cameraHeight = 2f;Đ
    public float cameraRotationSpeed = 2f;Đ
Đ
Đ
    private Rigidbody rb;Đ
    private Animator animator;Đ
    private Vector3 moveDirection;Đ
    private bool isGrounded;Đ
    private Transform mainCamera;Đ
    private float cameraRotationY;Đ
Đ
Đ
    private void Start()Đ
    {Đ
        canGrabController = GetComponent<CanGrabController>();Đ
        combatController = GetComponent<CharacterCombatController>();Đ
        characterStats = GetComponent<CharacterStats>();Đ
        skillController = GetComponent<SkillController>();Đ
        targetingSystem = GetComponent<TargetingSystem>();Đ
Đ
        //EDITOR CODEĐ
        skillController.skillTree.resetAllNodes();Đ
Đ
        rb = GetComponent<Rigidbody>();Đ
        animator = GetComponent<Animator>();Đ
        mainCamera = Camera.main.transform;Đ
        Cursor.lockState = CursorLockMode.Locked;Đ
        Cursor.visible = false;Đ
Đ
Đ
Đ
        skillController.OnSkillUnlocked += UpdateToSkillEvents;Đ
        stunnable = GetComponent<IStunnable>();Đ
    }Đ
Đ
```

```
Ð
   private void Update()Ð
   {Ð
      HandleMovement();Ð
      HandleJump();Ð
      HandleCamera();Ð
      HandleActions();Ð
   }Ð
   private void UpdateToSkillEvents(SkillNode node)Ð
   {Ð
      characterStats.UpdateSubStats();Ð
   }Ð
   GameObject target;Ð
   public void HandleActions(){Ð
Ð
      if(Input.GetKeyDown(KeyCode.E)){Ð
         Ð
         target = targetingSystem.GetTarget();Ð
         if(target == null){return;}Ð
Ð
             if(target.GetComponent<IInteractable>() != null){Ð
                     if(Vector3.Distance(target.transform.position,transform.position)
< 10f)Ð
                     {Ð
                target.GetComponent<IInteractable>().Interact(transform);Ð
                   }Ð
      }Ð
   }Ð
   }Ð
Ð
   private void HandleMovement()Ð
   {Ð
      if(stunnable != null && stunnable.isStunned())Ð
      {Ð
         return;Ð
      }Ð
      float horizontal = Input.GetAxis("Horizontal");Ð
      float vertical = Input.GetAxis("Vertical");Ð
Ð
      moveDirection = mainCamera.forward * vertical + mainCamera.right *
horizontal;Ð
      moveDirection.y = 0f;Ð
      moveDirection.Normalize();Ð
Ð
      if (moveDirection != Vector3.zero)Ð
```

```
        {Ð
            Quaternion targetRotation = Quaternion.LookRotation(moveDirection);Ð
            transform.rotation = Quaternion.RotateTowards(transform.rotation, 
targetRotation, rotationSpeed * Time.deltaTime);Ð
Ð
        }Ð
Ð
        animator.SetFloat("Speed", moveDirection.magnitude);Ð
        rb.MovePosition(rb.position + moveDirection * moveSpeed * Time.deltaTime);Ð
    }Ð
Ð
    private void HandleJump()Ð
    {Ð
        if(stunnable != null && stunnable.isStunned())Ð
        {Ð
            return;Ð
        }Ð
        isGrounded = Physics.Raycast(transform.position, Vector3.down, 0.4f, 
groundLayer);Ð
Ð
        if (Input.GetButtonDown("Jump") && isGrounded)Ð
        {Ð
            rb.AddForce(Vector3.up * jumpForce, ForceMode.Impulse);Ð
        }Ð
Ð
        // animator.SetBool("IsGrounded", isGrounded);Ð
    }Ð
Ð
    private void HandleCamera()Ð
    {Ð
        float mouseX = Input.GetAxis("Mouse X");Ð
        float mouseY = Input.GetAxis("Mouse Y");Ð
Ð
        cameraRotationY -= mouseY * cameraRotationSpeed;Ð
        cameraRotationY = Mathf.Clamp(cameraRotationY, -80f, 80f);Ð
Ð
        mainCamera.RotateAround(cameraTarget.position, Vector3.up, mouseX * 
cameraRotationSpeed);Ð
        mainCamera.localRotation = Quaternion.Euler(cameraRotationY, 
mainCamera.localEulerAngles.y, 0f);Ð
Ð
        Vector3 cameraOffset = new Vector3(0f, cameraHeight, -cameraDistance);Ð
        Vector3 targetPosition = cameraTarget.position + 
mainCamera.TransformDirection(cameraOffset);Ð
Ð
```

```
        mainCamera.position = Vector3.Lerp(mainCamera.position, targetPosition,
Time.deltaTime * rotationSpeed);Đ
        mainCamera.LookAt(cameraTarget);Đ
    }Đ
Đ
    public void faceIndirectionOfCamera()Đ
    {Đ
        transform.rotation = Quaternion.Euler(0f, mainCamera.localEulerAngles.y,
0f);Đ
    }Đ
    public bool TryUnlockSkillNode(SkillNode skillNode)Đ
    {Đ
        if (skillNode == null)Đ
        {Đ
            Debug.LogWarning("Invalid skill node.");Đ
            return false;Đ
        }Đ
        if (skillNode.isUnlocked)Đ
        {Đ
            Debug.LogWarning("Already learned.");Đ
            return false;Đ
        }Đ
Đ
        // Check if the character has enough skill points to unlock the node.Đ
        if (skillController.availableSkillPoints < skillNode.skillPointCost)Đ
        {Đ
            Debug.LogWarning("Not enough skill points.");Đ
            return false;Đ
        }Đ
Đ
        // Check if the required main stat meets the node's requirement.Đ
        bool statRequirementsMet = true;Đ
        for (int i = 0; i < skillNode.mainStatRequirement.Count; i++)Đ
        {Đ
            Archetype statName = skillNode.mainStatRequirement[i];Đ
            int requiredValue = skillNode.mainStatValue[i];Đ
Đ
            switch (statName)Đ
            {Đ
                case Archetype.Strength:Đ
                    if (characterStats.strength < requiredValue) statRequirementsMet =
false;Đ
                    break;Đ
                case Archetype.Intelligence:Đ
                    if (characterStats.intelligence < requiredValue) statRequirementsMet
= false;Đ
                break;Đ
                case Archetype.Dexterity:Đ
                    if (characterStats.dexterity < requiredValue) statRequirementsMet = 
false;Đ
                    break;Đ
                case Archetype.Endurance:Đ
                    if (characterStats.endurance < requiredValue) statRequirementsMet = 
false;Đ
                    break;Đ
                case Archetype.Wisdom:Đ
                    if (characterStats.wisdom < requiredValue) statRequirementsMet = 
false;Đ
                    break;Đ
                default:Đ
                    Debug.LogWarning("Invalid stat name in the skill node.");Đ
                    break;Đ
             }Đ
          }Đ
Đ
        if (!statRequirementsMet)Đ
        {Đ
            Debug.LogWarning("Main stat requirement not met.");Đ
            return false;Đ
        }Đ
Đ
        // Check if the required prerequisite skill has been unlocked.Đ
        if (skillNode.prerequisiteSkill != null && !
skillNode.prerequisiteSkill.isUnlocked)Đ
        {Đ
            Debug.LogWarning("Prerequisite skill not unlocked.");Đ
            return false;Đ
        }Đ
Đ
        // Check if the skill node is visible based on the fog of war mechanic.Đ
        if (!skillController.skillTree.IsVisible(skillNode))Đ
        {Đ
            Debug.LogWarning("Skill node is not visible.");Đ
            return false;Đ
        }Đ
Đ
        // Unlock the skill node.Đ
        skillNode.isUnlocked = true;Đ
Đ
        skillController.LearnSkill(skillNode);Đ
```

```
Ð
Ð
        return true;Ð
    }Ð
    public bool TryUnLearnSkillNode(SkillNode skillNode)Ð
    {Ð
        if (skillNode.isUnlocked == false)Ð
        {Ð
            return false;Ð
        }Ð
        skillNode.isUnlocked = false;Ð
        skillController.UnlearnSkill(skillNode);Ð
        return true;Ð
    }Ð
Ð
}Ð
```

- SkillController at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\SkillController.cs:
Code of file SkillController:

```
þÿusing System.Collections.Generic;Ð
using UnityEngine;Ð
Ð
class SkillController : MonoBehaviourÐ
{Ð
    public List<Skill> activeSkills;Ð
    public SkillTree skillTree;Ð
    public int availableSkillPoints;Ð
    public StatsModifier totalStatsModier;Ð
    public delegate void SkillEvent(SkillNode skillNode);Ð
    public event SkillEvent OnSkillUnlocked;Ð
    public event SkillEvent OnSkillUnlearnd;Ð
Ð
Ð
Ð
    public void LearnSkill(SkillNode skillNode)Ð
    {Ð
Ð
        // Call event to update the UI, etc.Ð
Ð
        activeSkills.Add(skillNode.skill);Ð

skillNode.skill.ApplySkill(this.gameObject.GetComponent<CharacterStats>());Ð
        totalStatsModier.Add(skillNode.skill.statModifier);Ð
        availableSkillPoints -= skillNode.skillPointCost;Ð
        OnSkillUnlocked?.Invoke(skillNode);Ð
Ð
    }Ð
    public void UnlearnSkill(SkillNode skillNode)Ð
    {Ð
        if (activeSkills.Remove(skillNode.skill))Ð
        {Ð
            totalStatsModier.Sub(skillNode.skill.statModifier);Ð
        }Ð
        availableSkillPoints += skillNode.skillPointCost;Ð

skillNode.skill.RemoveSkill(this.gameObject.GetComponent<CharacterStats>());Ð
        OnSkillUnlearnd?.Invoke(skillNode);Ð
    }Ð
Ð
}Ð
Ð
```

- TargetingSystem at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\TargetingSystem.cs:
Code of file TargetingSystem:

```
þÿusing UnityEngine;Ð
Ð
public class TargetingSystem : MonoBehaviourÐ
{Ð
    public Camera playerCamera;Ð
    public LayerMask targetLayerMask;Ð
    public GameObject currentTarget;Ð
    public GameObject crosshair;Ð
    public float maxTargetingDistance = 100f;Ð
    public Material highlightMaterial;Ð
    private GameObject lastTarget;Ð
    private Material originalMaterial;Ð
    public OutlineHighlight outlineHighlightController;Ð
Ð
    private void Update()Ð
    {Ð
        HandleCrosshairTargeting();Ð
        HandleMouseClickTargeting();Ð
        HighlightTarget();Ð
    }Ð
    private void Start()Ð
    {Ð
Ð
    }Ð
```

```
Đ
    private void HandleCrosshairTargeting()Đ
    {Đ
        RaycastHit hit;Đ
        Ray ray = playerCamera.ScreenPointToRay(crosshair.transform.position);Đ
Đ
        if (Physics.Raycast(ray, out hit, maxTargetingDistance, targetLayerMask))Đ
        {Đ
            currentTarget = hit.collider.gameObject;Đ
        }Đ
        elseĐ
        {Đ
            currentTarget = null;Đ
        }Đ
    }Đ
Đ
    private void HandleMouseClickTargeting()Đ
    {Đ
        if (Input.GetMouseButtonDown(0))Đ
        {Đ
            RaycastHit hit;Đ
            Ray ray = playerCamera.ScreenPointToRay(Input.mousePosition);Đ
Đ
            if (Physics.Raycast(ray, out hit, maxTargetingDistance, targetLayerMask))Đ
            {Đ
                currentTarget = hit.collider.gameObject;Đ
            }Đ
        }Đ
    }Đ
Đ
    public GameObject GetTarget()Đ
    {Đ
        return currentTarget;Đ
    }Đ
    private void HighlightTarget()Đ
    {Đ
        if (currentTarget != null)Đ
        {Đ
            outlineHighlightController.target = currentTarget.transform;Đ
            lastTarget = currentTarget;Đ
        }Đ
        elseĐ
        {Đ
            outlineHighlightController.target = null;Đ
        }Đ
    }Ð
Ð
}Ð
```

- VisualEffectController at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\PlayerAndUnitsComponent\VisualEffectController.cs:

Code of file VisualEffectController:

```
using System.Collections.Generic;Ð
using UnityEngine;Ð
Ð
public enum effectUnitPosition{Ð
    overHead,Ð
    underFeet,Ð
}Ð
public class VisualEffectController : MonoBehaviourÐ
{Ð
    public VisualEffectManager visualEffectManager;Ð
Ð
    public Transform positionOverHead;Ð
    public Transform positionUnderFeet;Ð
Ð
    private Transform goalTransform;Ð
    private List<(GameObject,float)> effectInstances = new
List<(GameObject,float)>();Ð
    public void SpawnEffect(string effectName, float effectDuration = 0,
effectUnitPosition effectPosition = effectUnitPosition.overHead)Ð
    {Ð
        if(effectPosition == effectUnitPosition.overHead){Ð
            goalTransform = positionOverHead;Ð
        }Ð
        if(effectPosition == effectUnitPosition.underFeet){Ð
            goalTransform = positionUnderFeet;Ð
        }Ð
Ð
         GameObject effectPrefab =
visualEffectManager.GetEffectPrefab(effectName);Ð
        if (effectPrefab != null)Ð
        {Ð
            GameObject effectInstance = Instantiate(effectPrefab, Vector3.zero,
Quaternion.identity, goalTransform);Ð
            effectInstance.transform.localPosition = Vector3.zero;Ð
            effectInstances.Add((effectInstance,Time.time+effectDuration));Ð
Ð
        }Ð
    }Ð
    void Update(){Ð
        for (int i = effectInstances.Count - 1; i >= 0; i--)Ð
        {Ð
            (GameObject, float) effectInstance = effectInstances[i];Ð
            if (effectInstance.Item2 < Time.time)Ð
            {Ð
                Destroy(effectInstance.Item1);Ð
                effectInstances.RemoveAt(i);Ð
                Debug.Log("effect removed");Ð
            }Ð
        }Ð
    }Ð
    Ð
Ð
}Ð
```

- GameEvent at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\QuestFiles\GameEvent.cs:
Code of file GameEvent:

```
using UnityEngine.Events;Ð
Ð
[System.Serializable]Ð
public class GameEvent : UnityEvent<string> { }Ð
```

- KillObjective at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\QuestFiles\KillObjective.cs:
Code of file KillObjective:

```
using UnityEngine;Ð
Ð
public class KillObjective : QuestObjectiveÐ
{Ð
    [SerializeField]Ð
    public string enemyId;Ð
    [SerializeField]Ð
    public int targetKills;Ð
    [SerializeField]Ð
    public int currentKills;Ð
Ð
    public KillObjective(string id, string description, string enemyId, int targetKills)Ð
    {Ð
        this.id = id;Ð
        this.description = description;Ð
        this.enemyId = enemyId;Ð
        this.targetKills = targetKills;Ð
        this.currentKills = 0;Đ
        this.status = ObjectiveStatus.Incomplete;Đ
    }Đ
Đ
    public override void UpdateProgress(string killedEnemyId)Đ
    {Đ
        if (killedEnemyId == "kill:"+enemyId && status != ObjectiveStatus.Completed)Đ
        {Đ
            currentKills++;Đ
            Debug.LogError("Current Kills: " + currentKills);Đ
            if (currentKills >= targetKills)Đ
            {Đ
                status = ObjectiveStatus.Completed;Đ
            }Đ
        }Đ
    }Đ
    public override string GetObjectiveProgress()Đ
    {Đ
        return currentKills + "/" + targetKills;Đ
    }Đ
Đ
    public override bool IsCompleted()Đ
    {Đ
        return status == ObjectiveStatus.Completed;Đ
    }Đ
}Đ
```

- Quest at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\QuestFiles\Quest.cs:
Code of file Quest:

```
using UnityEngine;Đ
using System.Collections.Generic;Đ
Đ
[CreateAssetMenu(fileName = "Quest", menuName = "ScriptableObjects/Quest",
order = 1)]Đ
public class Quest : ScriptableObjectĐ
{Đ
    public int id;Đ
    public string title;Đ
    public string description;Đ
    public List<QuestObjective> objectives;Đ
    public List<Reward> rewards;Đ
    public QuestStatus status;Đ
Đ
    public Quest(int id, string title, string description)Đ
    {Đ
        this.id = id;Đ
        this.title = title;Đ
        this.description = description;Đ
        this.objectives = new List<QuestObjective>();Đ
        this.rewards = new List<Reward>();Đ
        this.status = QuestStatus.NotStarted;Đ
    }Đ
Đ
    public void AddObjective(QuestObjective objective)Đ
    {Đ
        objectives.Add(objective);Đ
    }Đ
Đ
    public void AddReward(Reward reward)Đ
    {Đ
        rewards.Add(reward);Đ
    }Đ
Đ
    // The missing CheckAndUpdateObjectives methodĐ
    public void CheckAndUpdateObjectives(string objectiveId)Đ
    {Đ
        foreach (QuestObjective objective in objectives)Đ
        {Đ
            if ( objective.status == ObjectiveStatus.Incomplete)Đ
            {Đ
                objective.UpdateProgress(objectiveId);Đ
                if (objective.status == ObjectiveStatus.Completed)Đ
                {Đ
                    CheckQuestCompletion();Đ
                }Đ
                break;Đ
            }Đ
        }Đ
    }Đ
Đ
    private void CheckQuestCompletion()Đ
    {Đ
        bool allObjectivesComplete = true;Đ
        foreach (QuestObjective objective in objectives)Đ
        {Đ
            if (objective.status != ObjectiveStatus.Completed)Đ
            {Đ
                allObjectivesComplete = false;Đ
                break;Đ
            }Đ
        }Đ
Đ
        if (allObjectivesComplete)Đ
        {Đ
            status = QuestStatus.Completed;Đ
        }Đ
    }Đ
}Đ
public enum QuestStatus { NotStarted, InProgress, Completed }
```

- QuestAction at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\QuestFiles\QuestAction.cs:
Code of file QuestAction:

```
using UnityEngine;Đ
Đ
[CreateAssetMenu(fileName = "QuestAction", menuName = "QuestSystem/
QuestAction", order = 1)]Đ
public class QuestAction : ScriptableObjectĐ
{Đ
    public string actionId;Đ
}Đ
```

- QuestGiver at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\QuestFiles\QuestGiver.cs:
Code of file QuestGiver:

```
using UnityEngine;Đ
Đ
public class QuestGiver : MonoBehaviour, IInteractableĐ
{Đ
    [SerializeField] private Quest quest;Đ
 Đ
    [SerializeField] private GameObject interactionIndicator;Đ
Đ
    private bool playerInRange = false;Đ
    private QuestSystem playerQuestSystem;Đ
    private GameObject Interacts;Đ
    private UIManager uiManager;Đ
    public void Start(){Đ
        uiManager = FindObjectOfType<UIManager>();Đ
    }Đ
Đ
    void Update()Đ
    {Đ
        if (playerInRange && Input.GetKeyDown(KeyCode.E))Đ
        {Đ
```

```
            Interact(Interacts.transform);Đ
        }Đ
    }Đ
    public void Interact(Transform interactFrom)Đ
    {Đ
        if(uiManager.questUIPresenter.gameObject.activeInHierarchy ){Đ
            uiManager.hideQuestUiPresenter();Đ
        }Đ
        else{Đ
            uiManager.showQuestUiPresenter(quest);Đ
        }Đ
        Đ
    }Đ
Đ
    void OnTriggerEnter(Collider other)Đ
    {Đ
        if (other.CompareTag("Player"))Đ
        {Đ
            playerInRange = true;Đ
            Interacts = other.gameObject;Đ
            interactionIndicator.SetActive(true);Đ
            playerQuestSystem = other.GetComponent<QuestSystem>();Đ
        }Đ
    }Đ
Đ
    void OnTriggerExit(Collider other)Đ
    {Đ
        if (other.CompareTag("Player"))Đ
        {Đ
            playerInRange = false;Đ
            interactionIndicator.SetActive(false);Đ
           Đ
        }Đ
    }Đ
Đ
Đ
}Đ
```

- QuestObjective at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\QuestFiles\QuestObjective.cs:
Code of file QuestObjective:

```
using UnityEditor;Đ
using UnityEngine;Đ
Đ
public abstract class QuestObjectiveĐ
```

```
{Đ
    public string id;Đ
    public string description;Đ
    public ObjectiveStatus status;Đ
    public abstract bool IsCompleted();Đ
Đ
    public abstract void UpdateProgress(string infoId);Đ
    public abstract string GetObjectiveProgress();Đ
Đ
}Đ
public enum ObjectiveStatus { Completed,Incomplete };Đ
Đ
public class GatherObjective : QuestObjectiveĐ
{Đ
    public string itemId;Đ
    public int targetItems;Đ
    public int currentItems;Đ
Đ
    public GatherObjective(string id, string description, string itemId, int
targetItems)Đ
    {Đ
        this.id = id;Đ
        this.description = description;Đ
        this.itemId = itemId;Đ
        this.targetItems = targetItems;Đ
        this.currentItems = 0;Đ
        this.status = ObjectiveStatus.Incomplete;Đ
    }Đ
Đ
    public override void UpdateProgress(string gatheredItemId)Đ
    {Đ
        if (gatheredItemId == "gather:"+itemId && status !=
ObjectiveStatus.Completed)Đ
        {Đ
            currentItems++;Đ
            if (currentItems >= targetItems)Đ
            {Đ
                status = ObjectiveStatus.Completed;Đ
            }Đ
        }Đ
    }Đ
Đ
    public override bool IsCompleted()Đ
    {Đ
        return status == ObjectiveStatus.Completed;Đ
    }Đ
    public override string GetObjectiveProgress()Đ
    {Đ
        return currentItems + "/" + targetItems;Đ
    }Đ
}Đ
Đ
public class InspectObjective : QuestObjectiveĐ
{Đ
    public string locationId;Đ
    public bool locationInspected;Đ
Đ
    public InspectObjective(string id, string description, string locationId)Đ
    {Đ
        this.id = id;Đ
        this.description = description;Đ
        this.locationId = locationId;Đ
        this.locationInspected = false;Đ
        this.status = ObjectiveStatus.Incomplete;Đ
    }Đ
Đ
    public override void UpdateProgress(string inspectedLocationId)Đ
    {Đ
        if (inspectedLocationId == "visit:"+locationId && !locationInspected)Đ
        {Đ
            locationInspected = true;Đ
            status = ObjectiveStatus.Completed;Đ
        }Đ
    }Đ
Đ
    public override bool IsCompleted()Đ
    {Đ
        return locationInspected;Đ
    }Đ
    public override string GetObjectiveProgress()Đ
    {Đ
        return locationInspected ? "Inspected" : "Not Inspected";Đ
    }Đ
}Đ
Đ
public class ActivateObjective : QuestObjectiveĐ
{Đ
    public string altarId;Đ
    public bool altarActivated;Đ
Đ
```

```
    public ActivateObjective(string id, string description, string altarId)Đ
    {Đ
        this.id = id;Đ
        this.description = description;Đ
        this.altarId = altarId;Đ
        this.altarActivated = false;Đ
        this.status = ObjectiveStatus.Incomplete;Đ
    }Đ
Đ
    public override void UpdateProgress(string activatedAltarId)Đ
    {Đ
        if (activatedAltarId == "activate:"+altarId && !altarActivated)Đ
        {Đ
            altarActivated = true;Đ
            status = ObjectiveStatus.Completed;Đ
        }Đ
    }Đ
Đ
    public override bool IsCompleted()Đ
    {Đ
        return altarActivated;Đ
    }Đ
    public override string GetObjectiveProgress()Đ
    {Đ
        return altarActivated ? "Activated" : "Not Activated";Đ
    }Đ
}Đ
```

- HuntWolvesQuest at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\QuestFiles\QuestS\HuntWolvesQuest.cs:
Code of file HuntWolvesQuest:

```
using UnityEngine;Đ
using System.Collections.Generic;Đ
Đ
[CreateAssetMenu(fileName = "HuntWolves", menuName = "ScriptableObjects/
Quests/HuntWolves", order = 1)]Đ
public class HuntWolvesQuest : QuestĐ
{Đ
    public HuntWolvesQuest() : base(1, "Hunt the Wolves", "The village has been
suffering from frequent wolf attacks. They've asked you to hunt down 10 wolves
and bring back their pelts as proof.")Đ
    {Đ
        // Add a KillObjective to the list of objectivesĐ
        AddObjective(new KillObjective("HuntWolvesObjective", "Hunt 10 Wolves",
"Wolf", 10));Đ
```

```
    }Đ
}Đ
```

- NewBehaviourScript at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\QuestFiles\QuestS\NewBehaviourScript.cs:
Code of file NewBehaviourScript:

```
using System.Collections;Đ
using System.Collections.Generic;Đ
using UnityEngine;Đ
Đ
public class NewBehaviourScript : MonoBehaviourĐ
{Đ
    // Start is called before the first frame updateĐ
    void Start()Đ
    {Đ
        Đ
    }Đ
Đ
    // Update is called once per frameĐ
    void Update()Đ
    {Đ
        Đ
    }Đ
}Đ
```

- QuestSystem at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\QuestFiles\QuestSystem.cs:
Code of file QuestSystem:

```
using UnityEngine;Đ
using System.Collections.Generic;Đ
Đ
public class QuestSystem : MonoBehaviourĐ
{Đ
    public List<Quest> quests;Đ
    public UIManager uiManager;Đ
Đ
 Đ
public Quest GetQuestByID(int questID)Đ
{Đ
    foreach (Quest quest in quests)Đ
    {Đ
        if (quest.id == questID)Đ
        {Đ
            return quest;Đ
        }Đ
    }Ð
    return null;Ð
}Ð
    private void Start()Ð
    {Ð
        quests = new List<Quest>();Ð
    }Ð
Ð
    public void AddQuest(Quest quest)Ð
    {Ð
        uiManager.updateQuestBook();Ð
        quests.Add(quest);Ð
    }Ð
Ð
    public void RemoveQuest(int questId)Ð
    {Ð
Ð
        Quest questToRemove = quests.Find(q => q.id == questId);Ð
        if (questToRemove != null)Ð
        {Ð
          Ð
            quests.Remove(questToRemove);Ð
        }Ð
    }Ð
        public void UpdateQuestObjective(string objectiveId)Ð
    {Ð
        uiManager.updateQuestBook();Ð
        foreach (Quest quest in quests)Ð
        {Ð
            if (quest.status != QuestStatus.Completed)Ð
            {Ð
                quest.CheckAndUpdateObjectives(objectiveId);Ð
            }Ð
        }Ð
    }Ð
Ð
   Ð
}Ð
```

- Reward at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\QuestFiles\Reward.cs:
Code of file Reward:

```
using UnityEngine;Ð
Ð
[System.Serializable]Ð
```

```
public class RewardÐ
{Ð
    public string rewardId;Ð
    public string rewardName;Ð
    public int quantity;Ð
Ð
    public Reward(string rewardId, string rewardName, int quantity)Ð
    {Ð
        this.rewardId = rewardId;Ð
        this.rewardName = rewardName;Ð
        this.quantity = quantity;Ð
    }Ð
}Ð
```

- CharacterStatsUI at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\Ui\CharacterStatsUI.cs:
Code of file CharacterStatsUI:

```
þÿusing UnityEngine;Ð
using UnityEngine.UI;Ð
Ð
public class CharacterStatsUI : MonoBehaviourÐ
{Ð
    public Text unspentStatPointsText;Ð
    public Text strengthText;Ð
    public Text intelligenceText;Ð
    public Text dexterityText;Ð
    public Text enduranceText;Ð
    public Text wisdomText;Ð
Ð
    public Button strengthButton;Ð
    public Button intelligenceButton;Ð
    public Button dexterityButton;Ð
    public Button enduranceButton;Ð
    public Button wisdomButton;Ð
Ð
    public CharacterStats characterStats;Ð
Ð
    private void Start()Ð
    {Ð
        strengthButton.onClick.AddListener(() => IncreaseStat(Archetype.Strength));Ð
        intelligenceButton.onClick.AddListener(() =>
IncreaseStat(Archetype.Intelligence));Ð
        dexterityButton.onClick.AddListener(() => IncreaseStat(Archetype.Dexterity));Ð
        enduranceButton.onClick.AddListener(() =>
IncreaseStat(Archetype.Endurance));Ð
```

```
        wisdomButton.onClick.AddListener(() => IncreaseStat(Archetype.Wisdom));Ð
Ð
        characterStats.StatsChanged += UpdateUI;Ð
        UpdateUI();Ð
    }Ð
Ð
    private void Awake()Ð
    {Ð
        Cursor.visible = true;Ð
        Cursor.lockState = CursorLockMode.None;Ð
    }Ð
Ð
    private void UpdateUI()Ð
    {Ð
        unspentStatPointsText.text = "Unspent Points: " +
characterStats.unspentStatPoints;Ð
        strengthText.text = "Strength: " + characterStats.strength;Ð
        intelligenceText.text = "Intelligence: " + characterStats.intelligence;Ð
        dexterityText.text = "Dexterity: " + characterStats.dexterity;Ð
        enduranceText.text = "Endurance: " + characterStats.endurance;Ð
        wisdomText.text = "Wisdom: " + characterStats.wisdom;Ð
    }Ð
Ð
    private void IncreaseStat(Archetype mainStatType)Ð
    {Ð
        characterStats.IncreaseStat(mainStatType, 1);Ð
    }Ð
}Ð
```

- CharacterUi at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\Ui\CharacterUi.cs:
Code of file CharacterUi:

```
þÿusing UnityEngine;Ð
using UnityEngine.UI;Ð
Ð
public class CharacterUi : MonoBehaviourÐ
{Ð
    public Text unspentStatPointsText;Ð
    public Text strengthText;Ð
    public Text intelligenceText;Ð
    public Text dexterityText;Ð
    public Text enduranceText;Ð
    public Text wisdomText;Ð
Ð
Ð
```

```
Ð
   public Text subStatsPhysical;Ð
   public Text subStatsSpellCasting;Ð
   public Text subStatsDefensive;Ð
   public Text subStatsUniversal;Ð
Ð
   public Button openCharacterStatsMenu;Ð
   public Image unspentStatPoints;Ð
Ð
   public CharacterStats characterStats;Ð
   public UIManager uiManager;Ð
Ð
   private void Start()Ð
   {Ð
      openCharacterStatsMenu.onClick.AddListener(() =>
uiManager.OpenCharacterStatusUI());Ð
      characterStats.StatsChanged += UpdateUI;Ð
      UpdateUI();Ð
   }Ð
Ð
   private void Awake()Ð
   {Ð
      Cursor.visible = true;Ð
      Cursor.lockState = CursorLockMode.None;Ð
   }Ð
Ð
   private void UpdateUI()Ð
   {Ð
      strengthText.text = "Strength: " + characterStats.strength;Ð
      intelligenceText.text = "Intelligence: " + characterStats.intelligence;Ð
      dexterityText.text = "Dexterity: " + characterStats.dexterity;Ð
      enduranceText.text = "Endurance: " + characterStats.endurance;Ð
      wisdomText.text = "Wisdom: " + characterStats.wisdom;Ð
Ð
      subStatsPhysical.text = "Critical Chance: " +
characterStats.criticalChance.ToString("F1") + "%" + "\nCritical Damage: " +
characterStats.criticalDamage + "%" + "\nAttack Speed: " +
characterStats.attackSpeed.ToString("F2");Ð
      subStatsSpellCasting.text = "Spell Crit Chc: " +
characterStats.spellCriticalChance.ToString("F1") + "%" + "\nSpell Crit Dmg: " +
characterStats.spellCriticalDamage + "%" + "\nCooldown: " +
characterStats.cooldown;Ð
      subStatsDefensive.text = "Armor: " + characterStats.armor + "\nMagic Resi:
" + characterStats.magicResistance + "\nDodge Chance: " +
characterStats.dodgeChance.ToString("F1") + "%";Ð
```

```
        subStatsUniversal.text = "Max Life: " + characterStats.maxLife + "\nLife Reg:
" + characterStats.lifeRegen + "\nMax Mana: " + characterStats.maxMana +
"\nMana Reg: " + characterStats.manaRegen;Ð
    }Ð
Ð
}Ð
```

- DamageNumberController at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\Ui\DamageNumberController.cs:
Code of file DamageNumberController:

```
þÿusing TMPro;Ð
using UnityEngine;Ð
Ð
public class DamageNumberController : MonoBehaviourÐ
{Ð
    public TextMeshPro textMeshPro;Ð
    public float floatSpeed = 1f;Ð
    public float duration = 1.5f;Ð
Ð
    private float elapsedTime = 0f;Ð
    private Camera playerCamera;Ð
Ð
    private void Start()Ð
    {Ð
        playerCamera = Camera.main;Ð
    }Ð
Ð
    public void SetDamageValue(float damage)Ð
    {Ð
        if (textMeshPro == null)Ð
        {Ð
            Debug.LogError("TextMeshPro component is not assigned in the
DamageNumberController component.");Ð
            return;Ð
        }Ð
Ð
        textMeshPro.text = damage.ToString();Ð
    }Ð
Ð
    private void Update()Ð
    {Ð
        if (textMeshPro == null)Ð
        {Ð
            Destroy(gameObject);Ð
            return;Ð
        }Đ
Đ
        // Rotate towards player cameraĐ
        if (playerCamera != null)Đ
        {Đ
            FaceCamera();Đ
        }Đ
Đ
        // Float upwardsĐ
        transform.position += Vector3.up * floatSpeed * Time.deltaTime;Đ
Đ
        // Update the elapsed timeĐ
        elapsedTime += Time.deltaTime;Đ
Đ
        // Fade effectĐ
        textMeshPro.alpha = Mathf.Clamp01(1f - (elapsedTime / duration));Đ
Đ
        // Destroy the damage number after the durationĐ
        if (elapsedTime >= duration)Đ
        {Đ
            Destroy(gameObject);Đ
        }Đ
    }Đ
Đ
    private void FaceCamera()Đ
    {Đ
        Vector3 targetDirection = playerCamera.transform.position -
transform.position;Đ
        targetDirection.y = 0;Đ
        Quaternion targetRotation = Quaternion.LookRotation(-targetDirection);Đ
        transform.rotation = Quaternion.Slerp(transform.rotation, targetRotation, 1);Đ
    }Đ
}Đ
```

- GameManager at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\Ui\GameManager.cs:
Code of file GameManager:

```
using UnityEngine;Đ
Đ
public class GameManager : MonoBehaviourĐ
{Đ
    public static GameManager Instance;Đ
Đ
    public enum GameState { InMenu, Playing, Paused, GameOver }Đ
    public GameState currentState;Đ
```

```
Ð
    private void Awake()Ð
    {Ð
        if (Instance == null)Ð
        {Ð
            Instance = this;Ð
            DontDestroyOnLoad(gameObject);Ð
        }Ð
        elseÐ
        {Ð
            Destroy(gameObject);Ð
            return;Ð
        }Ð
Ð
        // Initialize game state and other systems as neededÐ
        currentState = GameState.InMenu;Ð
    }Ð
Ð
    private void Update()Ð
    {Ð
        HandleGameState();Ð
        UpdateCursorVisibility();Ð
    }Ð
Ð
    private void UpdateCursorVisibility()Ð
    {Ð
        // If the game is paused or in a menu, show the cursorÐ
        if (currentState == GameState.Paused || currentState == GameState.InMenu)Ð
        {Ð
            Cursor.visible = true;Ð
            Cursor.lockState = CursorLockMode.None;Ð
        }Ð
        // If the game is playing, hide the cursorÐ
        else if (currentState == GameState.Playing)Ð
        {Ð
            Cursor.visible = false;Ð
            Cursor.lockState = CursorLockMode.Locked;Ð
        }Ð
    }Ð
Ð
    private void HandleGameState()Ð
    {Ð
        switch (currentState)Ð
        {Ð
            case GameState.InMenu:Ð
```

```
                // Handle main menu logicĐ
                break;Đ
            case GameState.Playing:Đ
                // Handle playing state logicĐ
                break;Đ
            case GameState.Paused:Đ
                // Handle paused state logicĐ
                break;Đ
            case GameState.GameOver:Đ
                // Handle game over logicĐ
                break;Đ
        }Đ
    }Đ
Đ
    public void ChangeGameState(GameState newState)Đ
    {Đ
        currentState = newState;Đ
    }Đ
    public void SaveGame()Đ
    {Đ
        // Implement save game logicĐ
    }Đ
Đ
    public void LoadGame()Đ
    {Đ
        // Implement load game logicĐ
    }Đ
    // Implement other methods as needed, such as SaveGame, LoadGame, etc.Đ
}
```
- IDragable at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\Ui\Interfaces\IDragable.cs:
Code of file IDragable:
```
using UnityEngine.EventSystems;Đ
using UnityEngine;Đ
public interface IDragable : IBeginDragHandler, IDragHandler, IEndDragHandlerĐ
{Đ
Đ
    GameObject getDraggedObject();Đ
}Đ
```

- IRecieveDrop at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\Ui\Interfaces\IRecieveDrop.cs:
Code of file IRecieveDrop:
```
using UnityEngine.EventSystems;Đ
using UnityEngine;Đ
```

```
public interface IRecieveDrop : IPointerEnterHandler, IPointerExitHandler,
IDropHandlerĐ
{Đ
    Đ
}
- OutlineHighlight at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My
project\Assets\Scripts\Ui\OutlineHighlight.cs:
Code of file OutlineHighlight:
þÿusing UnityEngine;Đ
using UnityEngine.Rendering;Đ
Đ
[ExecuteInEditMode, ImageEffectAllowedInSceneView]Đ
public class OutlineHighlight : MonoBehaviourĐ
{Đ
    public Material highlightMaterial;Đ
    public Color highlightColor = Color.red;Đ
    public float outlineThickness = 2f;Đ
    public Transform target;Đ
Đ
    private Camera cam;Đ
    private CommandBuffer commandBuffer;Đ
Đ
    private void Start()Đ
    {Đ
        cam = GetComponent<Camera>();Đ
        commandBuffer = new CommandBuffer();Đ
    }Đ
Đ
    private void OnRenderImage(RenderTexture src, RenderTexture dest)Đ
    {Đ
        if (target == null)Đ
        {Đ
            Graphics.Blit(src, dest);Đ
            return;Đ
        }Đ
Đ
        commandBuffer.Clear();Đ
Đ
        var renderTexture = RenderTexture.GetTemporary(src.width, src.height,
src.depth, src.format);Đ
Đ
        commandBuffer.SetRenderTarget(renderTexture);Đ
Đ
        commandBuffer.ClearRenderTarget(true, true, Color.clear);Đ
Đ
```

```
        var meshFilter = target.GetComponent<MeshFilter>();Đ
        if (meshFilter != null)Đ
        {Đ
            commandBuffer.DrawMesh(meshFilter.sharedMesh,
target.localToWorldMatrix, highlightMaterial);Đ
        }Đ
Đ
        highlightMaterial.SetColor("_OutlineColor", highlightColor);Đ
        highlightMaterial.SetFloat("_OutlineThickness", outlineThickness);Đ
Đ
        Graphics.ExecuteCommandBuffer(commandBuffer);Đ
Đ
        Graphics.Blit(renderTexture, dest);Đ
Đ
        RenderTexture.ReleaseTemporary(renderTexture);Đ
    }Đ
}Đ
```

- OverlayUiController at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\Ui\OverlayUiController.cs:
Code of file OverlayUiController:

```
using UnityEngine;Đ
using UnityEngine.UI;Đ
Đ
public class OverlayUiController : MonoBehaviourĐ
{Đ
    private UIManager UIManager;Đ
    [SerializeField] public Text characterNameText;Đ
    [SerializeField] public Slider healthBar;Đ
    [SerializeField] public Slider manaBar;Đ
    [SerializeField] public Text levelText;Đ
    private GameObject player;Đ
Đ
    private HealthController HealthController;Đ
    private ManaController ManaController;Đ
Đ
    private void updateHealthBar()Đ
    {Đ
        healthBar.value = HealthController.currentHealth;Đ
        healthBar.maxValue = HealthController.maxHealth;Đ
    }Đ
    Đ
    private void updateManaBar()Đ
    {Đ
        manaBar.value = (ManaController.currentMana); Đ
        manaBar.maxValue = ManaController.maxMana;Ð
    }Ð
    private void updateHealthAndMana(){Ð
Ð
        updateHealthBar();Ð
        updateManaBar();Ð
    }Ð
    private void Start()Ð
    {Ð
        UIManager = FindObjectOfType<UIManager>();Ð
        player = FindObjectOfType<PlayerController>().gameObject;Ð
        HealthController = player.GetComponent<HealthController>();Ð
        ManaController = player.GetComponent<ManaController>();Ð
Ð
        Ð
    }Ð
    public void Update()Ð
    {Ð
        updateHealthAndMana();Ð
    }Ð
}
```

- PresentQuestUiController at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\Ui\PresentQuestUiController.cs:
Code of file PresentQuestUiController:

```
using UnityEngine;Ð
using UnityEngine.UI;Ð
using TMPro;Ð
Ð
public class PresentQuestUiController : MonoBehaviourÐ
{Ð
    [SerializeField] private TMP_Text questTitle;Ð
    [SerializeField] private TMP_Text questDescription;Ð
    [SerializeField] private Button acceptButton;Ð
    [SerializeField] private Button declineButton;Ð
Ð
    private QuestSystem questSystem;Ð
    void Start()Ð
    {Ð
        questSystem = FindObjectOfType<QuestSystem>();Ð
    }Ð
  Ð
Ð
    public void showQuestInfo(Quest quest,UIManager UIManager)Ð
    {Ð
        questTitle.text = quest.title;Ð
        questDescription.text = quest.description;Ð
        acceptButton.onClick.AddListener(() => questSystem.AddQuest(quest));Ð
        acceptButton.onClick.AddListener(() => UIManager.hideQuestUiPresenter());Ð
        declineButton.onClick.AddListener(() => UIManager.hideQuestUiPresenter());Ð
        Ð
    }Ð
Ð
}Ð
```

- QuestBookUIController at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\Ui\QuestBookUIController.cs:
Code of file QuestBookUIController:

```
using System.Collections.Generic;Ð
using UnityEngine;Ð
using UnityEngine.UI;Ð
using TMPro;Ð
Ð
public class QuestBookUIController : MonoBehaviourÐ
{Ð
    public TMP_Text titleText;Ð
    public TMP_Text descriptionText;Ð
    public TMP_Text objectivesText;Ð
    public ScrollRect questListScrollRect;Ð
    public GameObject questListItemPrefab;Ð
    public Transform questListContent;Ð
    Ð
    public QuestSystem questSystem;Ð
  Ð
Ð
    private void Awake()Ð
    {Ð
       Ð
    }Ð
Ð
    private void Start()Ð
    {  Ð
        UpdateQuestList();Ð
    }Ð
Ð
    public void UpdateQuestList()Ð
    {Ð
        // Clear the quest list contentÐ
        foreach (Transform child in questListContent)Ð
        {Ð
            Destroy(child.gameObject);Ð
        }Ð
Ð
        // Re-populate the quest list contentÐ
        foreach (Quest quest in questSystem.quests)Ð
        {Ð
            GameObject questListItem = Instantiate(questListItemPrefab,
questListContent);Ð
            questListItem.gameObject.SetActive(true);Ð
            questListItem.GetComponentInChildren<TMP_Text>().text = quest.title;Ð
            questListItem.GetComponent<Button>().onClick.AddListener(() =>
ShowQuestInformation(quest));Ð
        }Ð
Ð
        // Reset the scroll position of the quest listÐ
        questListScrollRect.verticalNormalizedPosition = 1f;Ð
    }Ð
Ð
    public void ShowQuestInformation(Quest quest)Ð
    {Ð
        // Set the quest information text fields to the current quest's dataÐ
        titleText.text = quest.title;Ð
        descriptionText.text = quest.description;Ð
        string objectivesString = "";Ð
        foreach(QuestObjective objective in quest.objectives)Ð
        {Ð
            objectivesString += $"-({objective.GetObjectiveProgress()})\n";Ð
        }   Ð
        Ð
    }Ð
}Ð
```

- QuestLogUIController at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\Ui\QuestLogUIController.cs:
Code of file QuestLogUIController:

```
using System.Collections.Generic;Ð
using UnityEngine;Ð
using TMPro;Ð
Ð
public class QuestLogUIController : MonoBehaviourÐ
{Ð
    public TMP_Text questLogText;Ð
    public TMP_Text trackingText;Ð
Ð
    private QuestSystem questSystem;Ð
    private List<Quest> trackingQuests = new List<Quest>();Ð
```

```
Đ
    private void Awake()Đ
    {Đ
        questSystem = FindObjectOfType<QuestSystem>();Đ
        if (questSystem == null)Đ
        {Đ
            Debug.LogError("No QuestSystem found in the scene!");Đ
        }Đ
    }Đ
Đ
    private void Start()Đ
    {Đ
        UpdateQuestLog();Đ
    }Đ
Đ
    public void UpdateQuestLog()Đ
    {Đ
        string questLogString = "";Đ
        foreach (Quest quest in questSystem.quests)Đ
        {Đ
            questLogString += $"[{quest.status}] {quest.title}\n";Đ
            foreach (QuestObjective objective in quest.objectives)Đ
            {Đ
                questLogString += $"- {objective.description}
({objective.GetObjectiveProgress()})\n";Đ
            }Đ
            questLogString += "\n";Đ
        }Đ
        questLogText.text = questLogString;Đ
Đ
        string trackingString = "Tracking: ";Đ
        foreach (Quest quest in trackingQuests)Đ
        {Đ
            trackingString += quest.title + ", ";Đ
        }Đ
        trackingText.text = trackingString.TrimEnd(',', ' ');Đ
    }Đ
Đ
    public void AddQuestToTrack(int questID)Đ
    {Đ
        Quest quest = questSystem.GetQuestByID(questID);Đ
        if (quest != null && !trackingQuests.Contains(quest))Đ
        {Đ
            trackingQuests.Add(quest);Đ
            UpdateQuestLog();Đ
        }Ð
    }Ð
Ð
    public void RemoveQuestToTrack(int questID)Ð
    {Ð
        Quest quest = questSystem.GetQuestByID(questID);Ð
        if (quest != null && trackingQuests.Contains(quest))Ð
        {Ð
            trackingQuests.Remove(quest);Ð
            UpdateQuestLog();Ð
        }Ð
    }Ð
}Ð
```

- SkillTreeMenuController at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\Ui\SkillTreeMenuController.cs:
Code of file SkillTreeMenuController:

```
þÿusing UnityEngine;Ð
Ð
public class SkillTreeMenuController : MonoBehaviourÐ
{Ð
    public GameObject[] skillTrees;Ð
    public int currentSkillTree = 0;Ð
    private void Start()Ð
    {Ð
         skillTrees[currentSkillTree].SetActive(true);Ð
    }Ð
Ð
    public void SwitchSkillTree(int index)Ð
    {Ð
        if (index < 0 || index >= skillTrees.Length) return;Ð
Ð
        skillTrees[currentSkillTree].SetActive(false);Ð
        skillTrees[index].SetActive(true);Ð
        currentSkillTree = index;Ð
    }Ð
}Ð
```

- SpellBookUiController at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\Ui\SpellBookUiController.cs:
Code of file SpellBookUiController:

```
using System.Collections.Generic;Ð
using UnityEngine;Ð
using UnityEngine.UI;Ð
using TMPro;Ð
```

```
Ð
public class SpellBookUiController : MonoBehaviourÐ
{Ð
    public TMP_Text titleText;Ð
    public TMP_Text descriptionText;Ð
    public TMP_Text objectivesText;Ð
    public ScrollRect spellListScrollRect;Ð
    public GameObject spellListItemPrefab;Ð
    public Transform spellListContent;Ð
    Ð
    public AbilityController abilityController;Ð
  Ð
Ð
    private void Awake()Ð
    {Ð
       Ð
    }Ð
Ð
    private void Start()Ð
    {  Ð
        UpdateQuestList();Ð
    }Ð
Ð
    public void UpdateQuestList()Ð
    {Ð
        // Clear the quest list contentÐ
        foreach (Transform child in spellListContent)Ð
        {Ð
            Destroy(child.gameObject);Ð
        }Ð
Ð
        // Re-populate the quest list contentÐ
        foreach (Ability ability in abilityController.learnedAbilitys)Ð
        {Ð
            GameObject spellListItem = Instantiate(spellListItemPrefab,
spellListContent);Ð
            spellListItem.gameObject.SetActive(true);Ð
            spellListItem.GetComponentInChildren<TMP_Text>().text = ability.name;Ð
            spellListItem.GetComponent<Button>().onClick.AddListener(() =>
ShowAbilityInformation(ability));Ð
            spellListItem.GetComponent<UiAbilitySlot>().ability = ability;Ð
            Ð
        }Ð
Ð
        // Reset the scroll position of the quest listÐ
        spellListScrollRect.verticalNormalizedPosition = 1f;Ð
    }Ð
Ð
    public void ShowAbilityInformation(Ability ability)Ð
    {Ð
        // Set the quest information text fields to the current quest's dataÐ
        titleText.text = ability.abilityName;Ð
        descriptionText.text = ability.abilityDescription;Ð
    string info = "";Ð
Ð
    info += "- Base Damage: " + ability.baseDamage + "\n";Ð
    info += "- Strength Scaling: " + ability.strengthScaling + "\n";Ð
    info += "- Intelligence Scaling: " + ability.intelligenceScaling + "\n";Ð
    info += "- Cooldown: " + ability.cooldown + "\n";Ð
    objectivesText.text = info;Ð
Ð
        /*foreach(QuestObjective objective in ability.)Ð
        {Ð
            objectivesString += $"-({objective.GetObjectiveProgress()})\n";Ð
        }   */Ð
        Ð
    }Ð
}Ð
```

- ToolTipUiController at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\Ui\ToolTipUiController.cs:
Code of file ToolTipUiController:

```
þÿusing UnityEngine;Ð
using UnityEngine.UI;Ð
Ð
public class ToolTipUiController : MonoBehaviourÐ
{Ð
    public Text SkillName;Ð
    public Text SkillDescription;Ð
    public Text AlreadySkilled;Ð
    public Text SkillpointCost;Ð
    public Text AttrbuteReq;Ð
    public Image SkillIcon;Ð
Ð
    private void Start()Ð
    {Ð
    }Ð
Ð
    private void Awake()Ð
    {Ð
```

```
        Cursor.visible = true;Đ
        Cursor.lockState = CursorLockMode.None;Đ
    }Đ
Đ
    public void UpdateUI(SkillNode node)Đ
    {Đ
        SkillName.text = node.skillName;Đ
        SkillDescription.text = node.skillDescription;Đ
        if (node.isUnlocked)Đ
        {Đ
            AlreadySkilled.gameObject.SetActive(true);Đ
        }Đ
        elseĐ
        {Đ
            AlreadySkilled.gameObject.SetActive(false);Đ
        }Đ
        SkillpointCost.text = "Cost: " + node.skillPointCost;Đ
Đ
        AttrbuteReq.text = "Requiment:";Đ
Đ
        for (int a = 0; a < node.mainStatRequirement.Count; a++)Đ
        {Đ
Đ
            AttrbuteReq.text += " " + node.mainStatRequirement[a] + ": " +
node.mainStatValue[a];Đ
        }Đ
Đ
        if (node.prerequisiteSkill != null)Đ
        {Đ
            AttrbuteReq.text += "Skill Requiment: " + node.prerequisiteSkill.skillName;Đ
        }Đ
        SkillIcon.sprite = node.icon;Đ
    }Đ
    public void UpdateUI(Ability ability)Đ
{Đ
    SkillName.text = ability.abilityName;Đ
    SkillDescription.text = ability.abilityDescription;Đ
    AlreadySkilled.gameObject.SetActive(false);Đ
    SkillpointCost.gameObject.SetActive(false);Đ
Đ
    AttrbuteReq.text = $"Base Damage: {ability.baseDamage}\nStrength Scaling:
{ability.strengthScaling}\nIntelligence Scaling: {ability.intelligenceScaling}";Đ
Đ
    SkillIcon.sprite = ability.icon;Đ
}Đ
```

```
Đ
public void UpdateUI(Item item)Đ
{Đ
    SkillName.text = item.itemName;Đ
    SkillDescription.text = item.description;Đ
    AlreadySkilled.gameObject.SetActive(false);Đ
    SkillpointCost.gameObject.SetActive(false);Đ
Đ
    if (item is EquipableItem equipableItem)Đ
    {Đ
        AttrbuteReq.text = $"Bonuses:\nStrength:
{equipableItem.strengthBonus}\nIntelligence:
{equipableItem.intelligenceBonus}\nDexterity:
{equipableItem.dexterityBonus}\nEndurance:
{equipableItem.enduranceBonus}\nWisdom: {equipableItem.wisdomBonus}";Đ
    }Đ
    elseĐ
    {Đ
        AttrbuteReq.text = "";Đ
    }Đ
Đ
    SkillIcon.sprite = item.icon;Đ
}Đ
Đ
Đ
}Đ
```

- UiAbilitySlot at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\Ui\UiAbilitySlot.cs:
Code of file UiAbilitySlot:

```
using UnityEngine.EventSystems;Đ
using UnityEngine;Đ
using UnityEngine.UI;Đ
internal class UiAbilitySlot : UiBaseDragAndDropFuncĐ
{Đ
   public Ability ability;Đ
   public Image icon;Đ
Đ
    private void Start()Đ
    {Đ
        if(ability!=null){Đ
            icon.sprite = ability.icon;Đ
        }Đ
    }Đ
Đ
```

```
   Ð
}Ð
```

- UiBaseDragAndDropFunc at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\Ui\UiBaseDragAndDropFunc.cs:
Code of file UiBaseDragAndDropFunc:

```
using UnityEngine;Ð
using UnityEngine.EventSystems;Ð
using UnityEngine.UI;Ð
public class UiBaseDragAndDropFunc : MonoBehaviour, IBeginDragHandler,
IDragHandler, IEndDragHandlerÐ
{Ð
    public RectTransform rectTransform;Ð
    private UIManager uiManager;Ð
    private GameObject dragObject;Ð
    public Vector3 originalPosition;Ð
    private GameObject dragedObject;Ð
Ð
    Ð
Ð
    private void Awake()Ð
    {Ð
        rectTransform = GetComponent<RectTransform>();Ð
        uiManager = FindObjectOfType<UIManager>();Ð
        originalPosition = rectTransform.localPosition;Ð
    }Ð
Ð
    public void OnBeginDrag(PointerEventData eventData)Ð
    {Ð
Ð
Ð
        dragedObject = eventData.pointerDrag;Ð
        dragObject = new GameObject("DragObject");Ð
        dragObject.transform.SetParent(uiManager.gameObject.transform);Ð
        dragObject.transform.SetSiblingIndex(uiManager.gameObject.transform.child
Count - 1);Ð
Ð
        RectTransform dragRectTransform =
dragObject.AddComponent<RectTransform>();Ð
        dragRectTransform.sizeDelta = rectTransform.sizeDelta;Ð
        dragRectTransform.position = eventData.position;Ð
Ð
Ð
Ð
        UiAbilitySlot uiAbilitySlot = dragedObject.GetComponent<UiAbilitySlot>();Ð
        UiItemSlot uiItemSlot = dragedObject.GetComponent<UiItemSlot>();Đ
            Image image = dragObject.AddComponent<Image>();Đ
        image.sprite = GetComponent<Image>().sprite;Đ
        image.raycastTarget = false;Đ
Đ
        if(uiAbilitySlot!=null)Đ
        {Đ
          image.sprite = uiAbilitySlot.icon.sprite;Đ
        }Đ
Đ
Đ
    }Đ
Đ
    public void OnDrag(PointerEventData eventData)Đ
    {Đ
        dragObject.GetComponent<RectTransform>().position = eventData.position;Đ
    }Đ
Đ
    public void OnEndDrag(PointerEventData eventData)Đ
    {Đ
Đ
Đ
        Destroy(dragObject);Đ
Đ
        if (eventData.pointerEnter != null)Đ
        {Đ
          UiHotKeySlot hotkeySlot =
eventData.pointerEnter.GetComponent<UiHotKeySlot>();Đ
          UiAbilitySlot uiAbilitySlot = dragedObject.GetComponent<UiAbilitySlot>();Đ
          UiItemSlot uiItemSlot = dragedObject.GetComponent<UiItemSlot>();Đ
Đ
Đ
          if (hotkeySlot != null)Đ
          {Đ
            if(uiAbilitySlot!=null)Đ
            {Đ
              hotkeySlot.ability = uiAbilitySlot.ability;Đ
              hotkeySlot.item = null;Đ
            }Đ
            else if(uiItemSlot!=null)Đ
            {Đ
              hotkeySlot.item = uiItemSlot.item;Đ
              hotkeySlot.ability = null;Đ
            }Đ
            elseĐ
```

```
                {Đ
                    hotkeySlot.item = null;Đ
                    hotkeySlot.ability = null;Đ
                }Đ
Đ
                hotkeySlot.updateinfo();Đ
            }Đ
        }Đ
Đ
        rectTransform.localPosition = originalPosition;Đ
    }Đ
}
```

- UiHotKeySlot at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\Ui\UiHotKeySlot.cs:
Code of file UiHotKeySlot:

```
using UnityEngine;Đ
using UnityEngine.EventSystems;Đ
using UnityEngine.UI;Đ
Đ
public class UiHotKeySlot : MonoBehaviourĐ
{Đ
    public Image icon;Đ
    public int hotkeyIndex;Đ
    private UIManager uiManager;Đ
    private RectTransform rectTransform;Đ
    public Vector3 originalPosition;Đ
    public Ability ability;Đ
    public Item item;Đ
Đ
    private HotkeyController hotkeyController;Đ
Đ
    private void Start()Đ
    {Đ
        hotkeyController = FindObjectOfType<HotkeyController>();Đ
        uiManager = FindObjectOfType<UIManager>();Đ
        rectTransform = GetComponent<RectTransform>();Đ
        icon = GetComponent<Image>();Đ
        originalPosition = rectTransform.localPosition;Đ
Đ
        Đ
        Đ
    }Đ
    public void updateinfo(){Đ
        if(ability!=null){Đ
            icon.sprite = ability.icon;Đ
            hotkeyController.hotkeys[hotkeyIndex].ability = ability;Đ
            hotkeyController.hotkeys[hotkeyIndex].item = null;Đ
        }Đ
        if(item!=null){Đ
            icon.sprite = item.icon;Đ
            hotkeyController.hotkeys[hotkeyIndex].item = item;Đ
            hotkeyController.hotkeys[hotkeyIndex].ability = null;Đ
        }Đ
Đ
Đ
    }Đ
    public  void OnPointerEnter(PointerEventData eventData)Đ
    {Đ
       uiManager.OpenToolTip(hotkeyController.hotkeys[hotkeyIndex],
rectTransform.position);Đ
    }Đ
Đ
    public  void OnPointerExit(PointerEventData eventData)Đ
    {Đ
        uiManager.CloseToolTip();Đ
    }Đ
Đ
Đ
}Đ
```

- UiItemSlot at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\Ui\UiItemSlot.cs:
Code of file UiItemSlot:

```
using UnityEngine;Đ
using UnityEngine.EventSystems;Đ
Đ
internal class UiItemSlot : ButtonWithToolTip, IRecieveDrop,IDragableĐ
{Đ
    public Item item;Đ
Đ
    public GameObject getDraggedObject()Đ
    {Đ
        throw new System.NotImplementedException();Đ
    }Đ
Đ
    public void OnBeginDrag(PointerEventData eventData)Đ
    {Đ
        throw new System.NotImplementedException();Đ
    }Đ
Đ
```

```
    public void OnDrag(PointerEventData eventData)Ð
    {Ð
        throw new System.NotImplementedException();Ð
    }Ð
Ð
    public void OnDrop(PointerEventData eventData)Ð
    {Ð
        throw new System.NotImplementedException();Ð
    }Ð
Ð
    public void OnEndDrag(PointerEventData eventData)Ð
    {Ð
        throw new System.NotImplementedException();Ð
    }Ð
}
```

- UIManager at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\Ui\UIManager.cs:

Code of file UIManager:

```
þÿ// UIManager.csÐ
using UnityEngine;Ð
public class UIManager : MonoBehaviourÐ
{Ð
    public GameObject pauseMenu;Ð
    public GameObject mainMenu;Ð
    public GameObject characterStatusUI;Ð
    public GameObject characterUi;Ð
    public GameObject tooltip;Ð
    public GameObject skillTreeMenu;Ð
Ð
    public GameObject questBookUi; Ð
    public GameObject questListQuickUi;Ð
    private ToolTipUiController toolTipController;Ð
    public PresentQuestUiController questUIPresenter;Ð
    public event eventUi onPlayerHealthManaChange;Ð
    public delegate void eventUi();Ð
Ð
public void updateQuestBook(){Ð
    questBookUi.GetComponent<QuestBookUIController>().UpdateQuestList();Ð
}Ð
public void showQuestBookUi(){Ð
    GameManager.Instance.ChangeGameState(GameManager.GameState.InMenu);Ð
    questBookUi.SetActive(true);Ð
}Ð
public void hideQuestBookUi(){Ð
    GameManager.Instance.ChangeGameState(GameManager.GameState.Playing);Ð
```

```
   questBookUi.SetActive(false);Đ
}Đ
public void showQuestListQuickUi(){Đ
   GameManager.Instance.ChangeGameState(GameManager.GameState.InMenu);Đ
   questListQuickUi.SetActive(true);Đ
}Đ
public void hideQuestListQuickUi(){Đ
   GameManager.Instance.ChangeGameState(GameManager.GameState.Playing);Đ
   questListQuickUi.SetActive(false);Đ
}Đ
Đ
public void showQuestUiPresenter(Quest quest){Đ
   GameManager.Instance.ChangeGameState(GameManager.GameState.InMenu);Đ
   questUIPresenter.showQuestInfo(quest,this);Đ
   questUIPresenter.gameObject.SetActive(true);Đ
}Đ
public void hideQuestUiPresenter(){Đ
   GameManager.Instance.ChangeGameState(GameManager.GameState.Playing);Đ
   questUIPresenter.gameObject.SetActive(false);Đ
}Đ
Đ
   private void Update()Đ
   {Đ
      // Check for user input to pause/unpause the gameĐ
      if (Input.GetKeyDown(KeyCode.Escape))Đ
      {Đ
         if (GameManager.Instance.currentState ==
GameManager.GameState.Playing)Đ
         {Đ
            PauseGame();Đ
         }Đ
         else if (GameManager.Instance.currentState ==
GameManager.GameState.Paused)Đ
         {Đ
            UnpauseGame();Đ
         }Đ
      }Đ
Đ
      // Check for user input to open/close the CharacterStatusUIĐ
      if (Input.GetKeyDown(KeyCode.C))Đ
      {Đ
         if (GameManager.Instance.currentState ==
GameManager.GameState.Playing)Đ
         {Đ
            OpenCharacterUi();Đ
            }Đ
            else if (GameManager.Instance.currentState ==
GameManager.GameState.InMenu)Đ
            {Đ
                CloseCharacterUi();Đ
            }Đ
        }Đ
        if(Input.GetKeyDown(KeyCode.V)){Đ
            if (GameManager.Instance.currentState ==
GameManager.GameState.Playing)Đ
            {Đ
                OpenSkillTreeMenu();Đ
            }Đ
            else if (GameManager.Instance.currentState ==
GameManager.GameState.InMenu)Đ
            {Đ
                CloseSkillTreeMenu();Đ
            }Đ
        }Đ
        if(Input.GetKeyDown(KeyCode.B)){Đ
            if (GameManager.Instance.currentState ==
GameManager.GameState.Playing)Đ
            {Đ
                OpenSkillTreeMenu();Đ
            }Đ
            else if (GameManager.Instance.currentState ==
GameManager.GameState.InMenu)Đ
            {Đ
                CloseSkillTreeMenu();Đ
            }Đ
        }Đ
    }Đ
    public void Awake()Đ
    {Đ
        toolTipController = tooltip.GetComponent<ToolTipUiController>();Đ
    }Đ
Đ
    public void PauseGame()Đ
    {Đ

GameManager.Instance.ChangeGameState(GameManager.GameState.Paused);Đ
        Time.timeScale = 0f;Đ
//      pauseMenu.SetActive(true);Đ
    }Đ
Đ
```

```
    public void UnpauseGame()Ð
    {Ð

GameManager.Instance.ChangeGameState(GameManager.GameState.Playing);Ð
        Time.timeScale = 1f;Ð
        pauseMenu.SetActive(false);Ð
    }Ð
Ð
    public void ShowMainMenu()Ð
    {Ð

GameManager.Instance.ChangeGameState(GameManager.GameState.InMenu);Ð
        mainMenu.SetActive(true);Ð
    }Ð
Ð
    public void HideMainMenu()Ð
    {Ð

GameManager.Instance.ChangeGameState(GameManager.GameState.Playing);Ð
        mainMenu.SetActive(false);Ð
    }Ð
    public void OpenCharacterStatusUI()Ð
    {Ð

GameManager.Instance.ChangeGameState(GameManager.GameState.InMenu);Ð
        characterStatusUI.SetActive(true);Ð
    }Ð
    public void OpenCharacterUi()Ð
    {Ð

GameManager.Instance.ChangeGameState(GameManager.GameState.InMenu);Ð
        characterUi.SetActive(true);Ð
    }Ð
    public void CloseCharacterUi()Ð
    {Ð

GameManager.Instance.ChangeGameState(GameManager.GameState.Playing);Ð
        characterUi.SetActive(false);Ð
    }Ð
    public void OpenSkillTreeMenu()Ð
    {Ð

GameManager.Instance.ChangeGameState(GameManager.GameState.InMenu);Ð
        skillTreeMenu.SetActive(true);Ð
    }Ð
    public void CloseSkillTreeMenu()Đ
    {Đ

GameManager.Instance.ChangeGameState(GameManager.GameState.Playing);Đ
        CloseToolTip();Đ
        skillTreeMenu.SetActive(false);Đ
    }Đ
    public void CloseCharacterStatusUI()Đ
    {Đ
        characterStatusUI.SetActive(false);Đ
    }Đ
    public void OpenToolTip(SkillNode node, Vector3 Positon)Đ
    {Đ
        tooltip.gameObject.SetActive(true);Đ
        tooltip.gameObject.GetComponent<RectTransform>().position = Positon;Đ
        toolTipController.UpdateUI(node);Đ
    }Đ
        public void OpenToolTip(Hotkey hotkey, Vector3 Positon)Đ
    {Đ
        tooltip.gameObject.SetActive(true);Đ
        tooltip.gameObject.GetComponent<RectTransform>().position = Positon;Đ
        if(hotkey.ability != null){Đ
            toolTipController.UpdateUI(hotkey.ability);Đ
        }Đ
        else if(hotkey.item != null){Đ
            toolTipController.UpdateUI(hotkey.item);Đ
        }Đ
    }Đ
    public void OpenToolTip(Ability ability, Vector3 Positon)Đ
    {Đ
        tooltip.gameObject.SetActive(true);Đ
        tooltip.gameObject.GetComponent<RectTransform>().position = Positon;Đ
        toolTipController.UpdateUI(ability);Đ
    }Đ
    public void OpenToolTip(Item item, Vector3 Positon)Đ
    {Đ
        tooltip.gameObject.SetActive(true);Đ
        tooltip.gameObject.GetComponent<RectTransform>().position = Positon;Đ
        toolTipController.UpdateUI(item);Đ
    }Đ
    public void CloseToolTip()Đ
    {Đ
        tooltip.gameObject.SetActive(false);Đ
    }Đ
}Đ
```

- WorldSpaceCanvasController at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\Ui\WorldSpaceCanvasController.cs:
Code of file WorldSpaceCanvasController:

```
þÿusing UnityEngine;Ð
Ð
public class WorldSpaceCanvasController : MonoBehaviourÐ
{Ð
    public static WorldSpaceCanvasController Instance;Ð
Ð
    public GameObject damageNumberPrefab;Ð
Ð
    private void Awake()Ð
    {Ð
        if (Instance == null)Ð
        {Ð
            Instance = this;Ð
        }Ð
        elseÐ
        {Ð
            Destroy(gameObject);Ð
        }Ð
    }Ð
Ð
    public void SpawnDamageNumber(float damage, Vector3 position)Ð
    {Ð
        if (damageNumberPrefab == null)Ð
        {Ð
            Debug.LogError("DamageNumberPrefab is not assigned in the
WorldSpaceCanvasController component.");Ð
            return;Ð
        }Ð
Ð
        GameObject damageNumberInstance = Instantiate(damageNumberPrefab,
position, Quaternion.identity, transform);Ð
        damageNumberInstance.gameObject.SetActive(true);Ð
        DamageNumberController damageNumberController =
damageNumberInstance.GetComponent<DamageNumberController>();Ð
Ð
Ð
        if (damageNumberController != null)Ð
        {Ð
            damageNumberController.SetDamageValue(damage);Ð
        }Ð
        elseÐ
        {Ð
            Debug.LogError("DamageNumberController component is missing on the
DamageNumberPrefab.");Ð
            Destroy(damageNumberInstance);Ð
        }Ð
    }Ð
}Ð
```

- UnitSpawnerController at C:\Users\Toastbrot\Downloads\STRATEGY 01.04.2022\My project\Assets\Scripts\WorldManagmentUnitSpawning\UnitSpawnerController.cs:

Code of file UnitSpawnerController:

```
using System.Collections;Ð
using System.Collections.Generic;Ð
using UnityEngine;Ð
Ð
class UnitSpawnerController : MonoBehaviourÐ
{Ð
    // VariablesÐ
    public GameObject unitPrefab; // Prefab of the unit to spawnÐ
    public float spawnRange; // Range at which to spawn the unitÐ
    public CharacterStats stats; // The stats for the spawned unitÐ
    public Ability[] abilities; // The abilities for the spawned unitÐ
    public AIState aiState; // The AI state for the spawned unitÐ
    public EquipManager equipManager; // The equip manager for the spawned unitÐ
    public AIController aiController; // The AI controller for the spawned unitÐ
Ð
    private Transform playerTransform; // Player transform to check distanceÐ
Ð
    void Start()Ð
    {Ð
        // Get the player transformÐ
        playerTransform = GameObject.FindGameObjectWithTag("Player").transform;Ð
    }Ð
Ð
    bool spawned = false;Ð
    void Update()Ð
    {Ð
        // Check if player is within spawn rangeÐ
        if(playerTransform == null){Ð
            return;Ð
        }Ð
        if(spawned){Ð
            return;Ð
        }Ð
        if(Vector3.Distance(transform.position, playerTransform.position) <=
spawnRange)Đ
        {Đ
            // Spawn the unitĐ
            GameObject unit = Instantiate(unitPrefab, transform.position,
Quaternion.identity);Đ
            unit.SetActive(true);Đ
            // Set the equip manager for the unitĐ
            unit.GetComponentInChildren<EquipManager>().SetEquipManager(equipM
anager);Đ
Đ
            // Set the AI controller for the unitĐ

unit.GetComponentInChildren<AIController>().SetAIController(aiController);Đ
            Đ
            // Set the stats for the unitĐ
            unit.GetComponentInChildren<CharacterStats>().SetStats(stats);Đ
Đ
            // Set the abilities for the unitĐ
            AbilityController abilityController =
unit.GetComponent<AbilityController>();Đ
            foreach(Ability ability in abilities)Đ
            {Đ
                abilityController.AddAbility(ability);Đ
            }Đ
Đ
            // Set the AI state for the unitĐ
            unit.GetComponentInChildren<AIController>().ChangeState(aiState);Đ
            Destroy(this.gameObject);Đ
        }Đ
    }Đ
}Đ
```