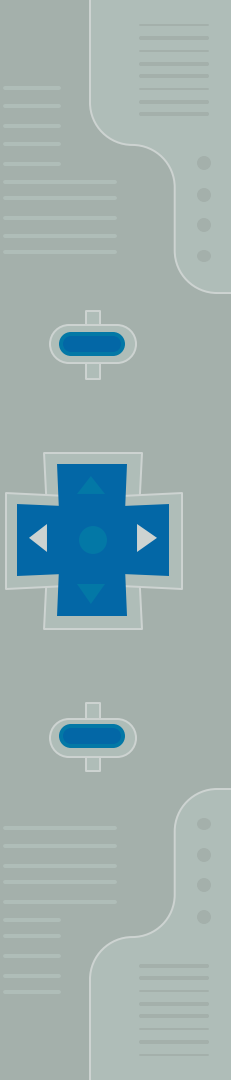


Green Sky Games

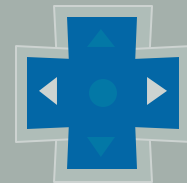


1 Bidhan / 2 Todd / 2+ Alex / 3 Mark / 4 Shan / 5 Hengyi / 6 Liz





GREEN SKY GAMES



01

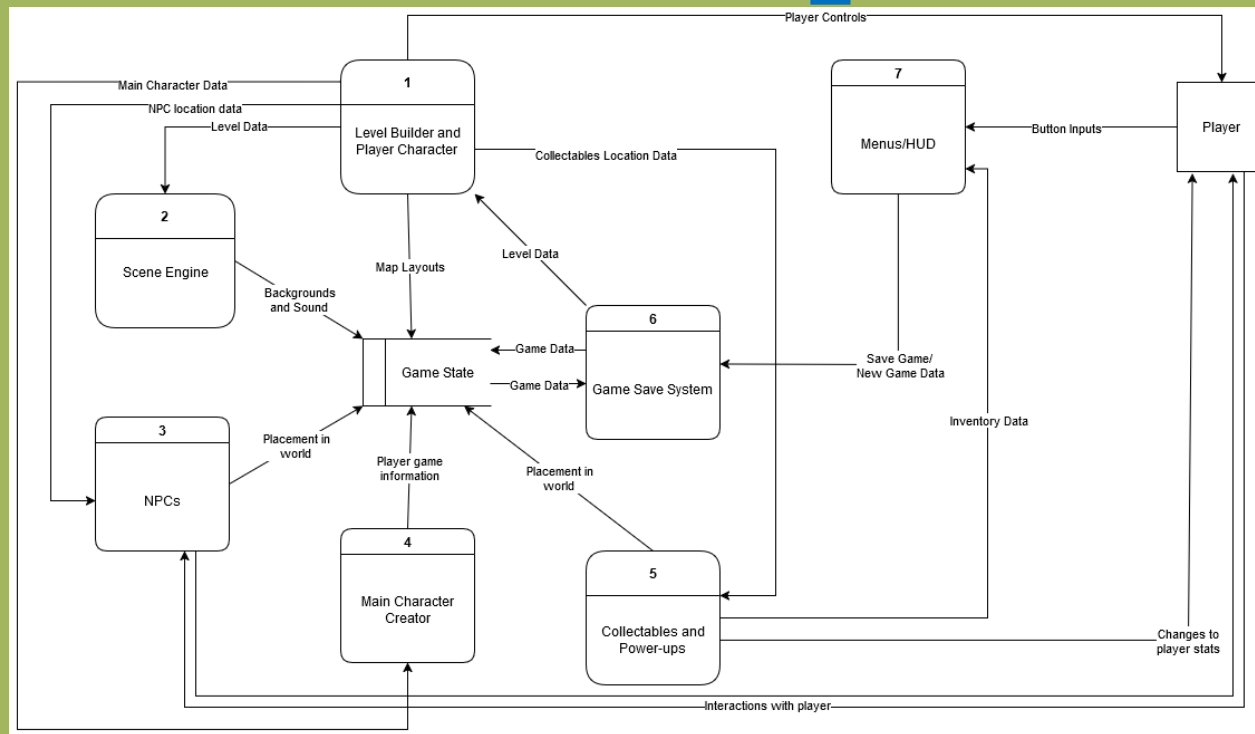
Post Mortem

Speaking: - Alex, TL2+



GREEN SKY GAMES

Data Flow Diagram



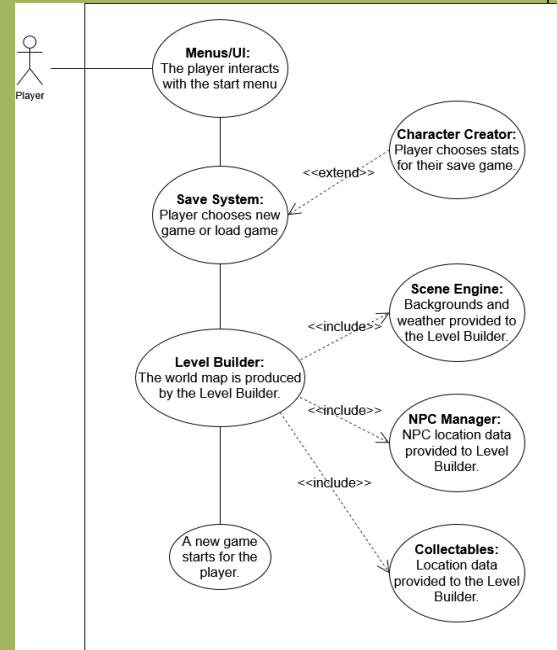
Speaking: - Alex, TL2+



GREEN SKY GAMES

Team Member Breakdown

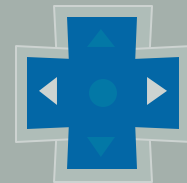
- TL1: Save System
- TL2: Storyline, dialogue, NPC responses
- TL2+: Level Design, Terrain Interaction
- TL3: Main Character Customization
- TL4: Power-ups/Collectibles, Inventory
- TL5: Scene Engine: Environment and Audio Systems
- TL6: Game menus/ screens, BC Mode toggle
- Data, stamp, and control coupling – low to medium



Speaking: - Alex, TL2+ and Shan, TL4



GREEN SKY GAMES



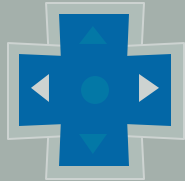
02

GRASP Breakdown

Speaking: - Alex, TL2+



GREEN SKY GAMES



TL1: Bidhan

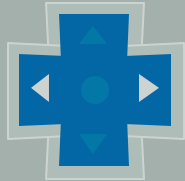
- Contributions: Save System
 - Includes preserving progress during gameplay.
 - Enabling respawn at saved location.
- Data stored: Player Position (X,Y,Z).
- Storage: PlayerPrefs
- High cohesion.

```
float x = PlayerPrefs.GetFloat("PlayerPosX");  
float y = PlayerPrefs.GetFloat("PlayerPosY");  
float z = PlayerPrefs.GetFloat("PlayerPosZ");  
Vector3 savedPosition = new Vector3(x, y, z);
```

Speaking: - Bidhan, TL1



GREEN SKY GAMES



TL2: Todd Carter

1. Dialogue system

- Dialogue box canvas with a Decorator pattern
- Branching dialogue with multiple choices
- Game events occurring from dialogue choices

2. NPCs

- Combat and hostility response
- Unique animations and sprites

3. Writing and storyline

- Dialogue for each NPC
- Created two different methods to reach victory

- High sequential and logical cohesion, though became worse towards wrap up



Speaking: - Todd, TL2

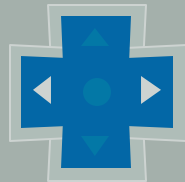
TL2:

```
//Quality of life functions for changing hostility.  
//Swap hostility  
public void switchHostility()  
{  
    this.isHostile = !isHostile;  
}  
//Turn off hostility  
public void offHostility()  
{  
    Debug.Log("Hostility off test.");  
    this.isHostile = false;  
}  
//Turn on hostility  
public void onHostility()  
{  
    Debug.Log("Hostility on test.");  
    this.isHostile = true;  
}  
//Set hostility  
public void setHostility(bool activate)  
{  
    this.isHostile = activate;  
}
```

Speaking: - Todd, TL2



GREEN SKY GAMES



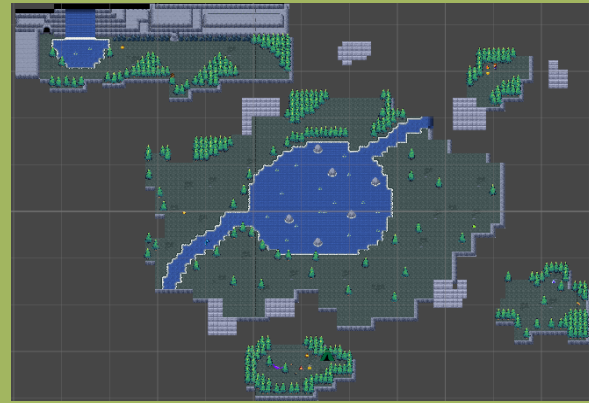


GREEN SKY GAMES

TL2+!

1. Level Design

- Individual map layouts for levels 1 through 5
- Level building with a Builder pattern
- Unique sprites and tile maps for different sections

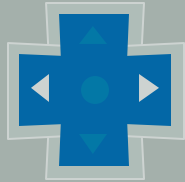


2. Terrain Objects

- Effects and responses for 5 different terrain object types
 - Various sprites and collision events
- Generally high cohesion through a mix of functional, sequential, and informational cohesion



Speaking: - Alex, TL2+



TL2+:

```
/* This code checks the tag of an object when it collides with a passageway and plays a sound upon impact if it is a player
 * It also chooses the next destination to transport the user to based on the current tag of the passageway and sends that to the getDestination function*/
@ Unity Message | 2 references
public virtual void OnTriggerEnter2D(Collider2D other)
{
    if(other.CompareTag("Player"))
    {
        string passageTag = this.tag;
        Debug.Log($"Player entered passage with tag: {passageTag}");
        LevelManager.Instance.tryLoadLevel(passageTag);
        AudioManager.instance.Play("DoorSound"); // Play the open door sound

        Vector2 newPosition = getDestination();
        if(newPosition != Vector2.zero)
        {
            // Log the tag to check the value
            Debug.Log("Current tag: " + gameObject.tag); // Print the tag of the current door

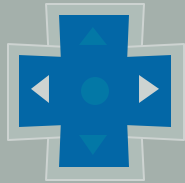
            Rigidbody2D rb = other.GetComponent<Rigidbody2D>();
            if (rb != null)
            {
                rb.linearVelocity = Vector2.zero;
                rb.constraints = RigidbodyConstraints2D.FreezeAll;
            }

            //other.transform.position = newPosition;
            StartCoroutine(teleportWithFade(other, newPosition, rb));
        }
    }
}
```

Speaking: - Alex, TL2+

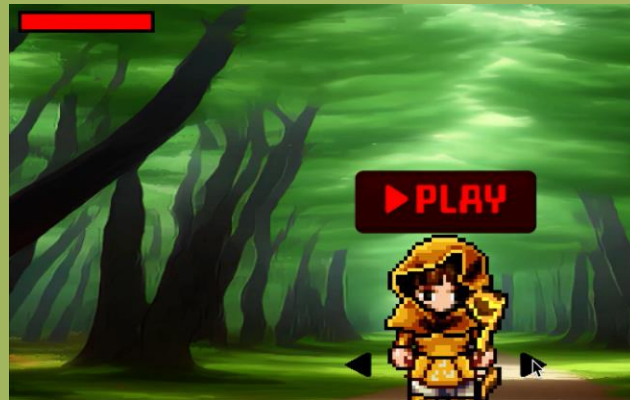
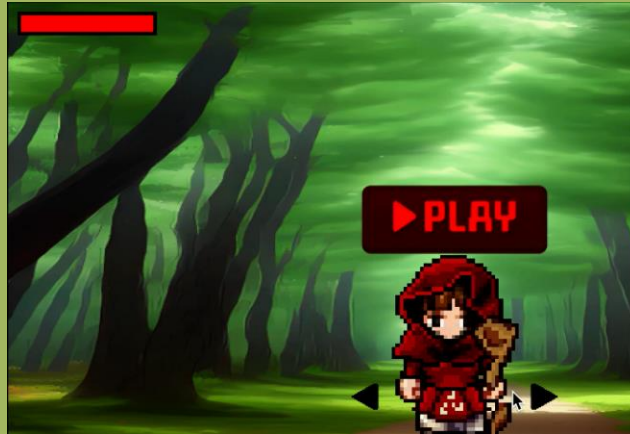


GREEN SKY GAMES



TL3:

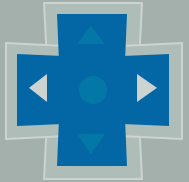
- Contributions:
 - Made different versions of Riding Hood (Red, Blue, Yellow, etc.)
 - Created a Character Selection Scene
 - Implemented animations to Riding Hood
 - Created a Health Bar for Riding Hood
- High cohesion



Speaking: - Mark, TL3



GREEN SKY GAMES



TL4:

1. Power-ups

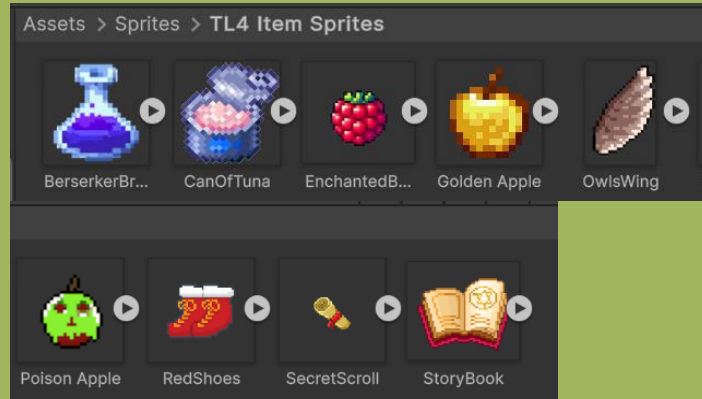
- Change player stats
- Choice prompt on collision
- Template Method pattern
- Unique sprites

2. Collectibles

- Automatically added to inventory
- Returns bool to The Cat if in inventory for different ending

3. Inventory

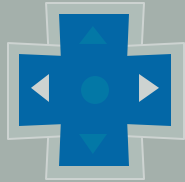
- Displays information on items
- Items can be added, removed, or used
- Slots increase with more clues



Speaking: - Shan, TL4



GREEN SKY GAMES



```

/* This function adds an item to the inventory using its name and sprite
 * It looks at the item slots I created in UI b/c I attached them in Inspector
 * Checks if there is an unoccupied slot and if not, put the item in
 * Uses isOccupied bool and updateInventoryUI() from ItemSlot script */
public bool addToInventory(string itemName, Sprite itemSprite, string itemDescription)
{
    Debug.Log($"maxInventory = " + maxInventorySize);
    for (int i = 0; i < maxInventorySize; i++)
    {
        if (itemSlot[i] != null && itemSlot[i].isOccupied == false)
        {
            itemSlot[i].updateInventoryUI(itemName, itemSprite, itemDescription);
            return true; // successfully added item to the inventory
        }
    }
    Debug.Log("Inventory is full or slot invalid!");
    return false;
}

```

➤ Functional and sequential cohesion-moderately high

```

/* This function attaches the item name and sprite to the slot so we can see it
 * If an item is added, it makes isOccupied true */
public void updateInventoryUI(string itemName, Sprite itemSprite, string itemDescription)
{
    if (itemImage == null)
    {
        Debug.LogWarning("Item image is missing or has been destroyed!");
        return;
    }
    this.itemName = itemName;
    this.itemSprite = itemSprite;
    this.itemDescription = itemDescription;
    isOccupied = true;

    itemImage.sprite = itemSprite;
}

```

Speaking: - Shan, TL4



GREEN SKY GAMES

TL5: Hengyi Tian

1. Environment System

- **Weather System:** 4 weather, including rain, Snow, Fog, and Dust
- **Transition Scenes:** 7 transition scenes, including Green Sky and Tornadoes
- Automatically change based on storyline and maps



2. Audio System

- **BGM System:** 8 environmental sounds, 9 background sounds
- **Dialogue System:** 11 dialogue sounds for NPCs
- **Sound Effects:** 8 sound effects
- Automatically change based on storyline and maps



Speaking: - Hengyi, TL5



GREEN SKY GAMES

- High Functional Cohesion contributes to single, clearly-defined tasks
- E.g. Sound Effect examples

```
public class AudioManager : MonoBehaviour
{
    public static AudioManager instance; // Static instance of the AudioManager class
    public AudioType[] AudioTypes; // Array to hold the audio types
    public List<AudioSource> allAudioSources = new List<AudioSource>(); // List to store all a
    public List<bool> audioSourceStates = new List<bool>(); // List to store the state, play o

    // Awake method is called when the object is initialized. Checking if the game is already
    // Ensure that the AudioManager is a singleton across scenes.

    private void Awake()
    {
        if (instance == null)
        {
            instance = this;
            DontDestroyOnLoad(gameObject); // Ensures AudioManager persists across scenes
            Debug.Log("AudioManager instance initialized.");
        }
        else
        {
            Destroy(gameObject); // Destroy duplicate AudioManager if one already exists
            return;
        }
    }
}
```

```
// Moves the door in one direction and then disables it
1 个引用
public void OneWay()
{
    // Play the passway sound when the door starts moving
    AudioManager.instance.Play("PassagewaySound");

    StartCoroutine(MoveAndDisable());
}
```

```
public class BackgroundMusic : MonoBehaviour
{
    public static BackgroundMusic instance; // Static instance of the Background class

    // Awake method is called when the object is initialized. Ensure the singleton pattern is appl
    // not destroyed on scene load. This method is also subscribes to scene change events.
    // Unity 3D 10 - 1-3-100

    private void Awake()
    {
        if (instance == null)
        {
            instance = this;
            DontDestroyOnLoad(gameObject); // Ensure Background object persists across new scenes
            Debug.Log("BackgroundMusic instance initialized.");
        }
        else
        {
            Destroy(gameObject);
            return;
        }

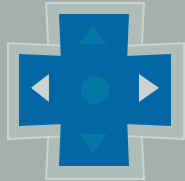
        SceneManager.sceneLoaded += OnSceneLoaded; // Listen for scene changes
    }
}
```

```
//Check if the player is in attack range by checking the player layer.
//If something is found, the hits array is > 0, and damage and knockback are dealt.
0 个引用
public void attack()
{
    Collider2D[] hits = Physics2D.OverlapCircleAll(AttackPoint.position, weaponRange, pl

    if(hits.Length > 0)
    {
        hits[0].GetComponent<PlayerHealth>().changeHealth(-damage);
        //hits[0].GetComponent<PlayerController>().Knockback(transform, knockbackForce,

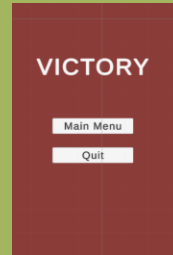
        // Play attack sound
        AudioManager.instance.Play("AttackSound");
    }
}
```

Speaking: - Hengyi, TL5



TL6: Liz

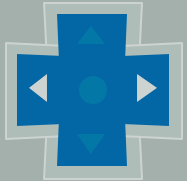
- Contributions:
 - Start Menu
 - Used the facade pattern
 - Pause Menu (Prefab)
 - Used the singleton pattern
 - Victory Screen
 - Game Over Screen
 - Credits Screen
 - BC Mode (Toggle and behavior)
- High Cohesion



Speaking: - Liz, TL6

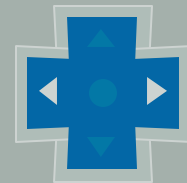


GREEN SKY GAMES





GREEN SKY GAMES



Project Improvements

Speaking: - Alex, TL2+



GREEN SKY GAMES

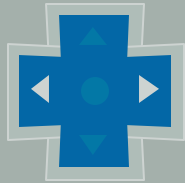
How to break down the project better

- Reduce NPC reliance on player
- Separate out player stats and movement elements
- Combine player creation with menus

Speaking: - Alex, TL2+



GREEN SKY GAMES



04

Course Improvements

Speaking: - Alex, TL2+



GREEN SKY GAMES

How to make the course better.

- Clearer expectations for project weekly deliverables
- Elaborate on required coding elements for the final game at the start of the course
 - Unity Testing, Dynamic/Static Binding, Patterns, Coding Standards, Demo, Copyright Implementation
- Modify order for software development specialists presentations to maximize understanding and usefulness

Speaking: - Alex, TL2+



GREEN SKY GAMES



05

Pair Programming

Speaking: - Alex, TL2+



GREEN SKY GAMES

Pair Programming: Game Prologue



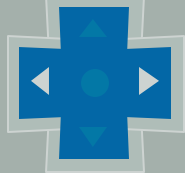
Action:

- Full screen wall of text that displays the start to the storyline (Star Wars style).

Dialogue:

- There will be text displayed on the screen for the player to read, as well as a voiceover for players who don't want to.

Speaking: - Alex, TL2+



THANKS!

Any Questions?



GREEN SKY GAMES

