(Technical Design Document)

God of Kung Fu

Action-centric 2D Platformer game

TABLE OF CONTENTS

Table of Contents

Game Introduction

Title: *God of Kung Fu*

Genre: 2D Action-Platformer / Martial Arts Adventure

Inspiration: Classic martial arts beat 'em ups, Metroidvania-style progression, and narrative-driven combat games.

Premise

In *God of Kung Fu*, you play as a young martial arts disciple whose master, Guang, falls victim to the Shadow Lord, a malevolent force corrupting the greatest warriors of the land. Before dying, Master Guang reveals that his three siblings: Huo Feng (Master of Speed), Shui Lian (Master of the Soaring Leap), and Lei Shan (Master of Chi) have been enslaved by the Shadow Lord's dark influence.

Your journey takes you across treacherous landscapes, where you must battle corrupted warriors, free them from the Shadow Lord's grip, and inherit their techniques. Only by mastering their skills can you enter the Cave of Reflection, where you must confront your own shadow—an ultimate test of skill and spirit.

The story explores themes of self-mastery, redemption, and the balance between light and darkness, culminating in a final battle against the Shadow Lord himself.

Gameplay Features

1. **Combat System**
    - **Melee & Ranged Attacks:** Fluid kung fu strikes, combos, and special moves.
    - **Animation-Driven Feedback:** Attacks have weight, with hit-stop effects and knockback.
    - **Boss Battles:** Each corrupted master has unique attack patterns, requiring mastery of newly learned skills.
2. **Movement & Platforming**
    - **Precision Jumps & Wall Interactions:** Navigate cliffs, traps, and collapsing terrain.

- **Skill-Based Mobility:**
    - **Speed Boost (Huo Feng's Technique):** Dash through enemies and evade attacks.
    - **High Jump (Shui Lian's Technique):** Scale tall structures and avoid ground hazards.
    - **Chi Blast (Lei Shan's Technique):** A ranged attack that also interacts with the environment (e.g., breaking barriers).

3. Progression & Abilities

- **Unlock New Techniques:** Defeating bosses grants their signature moves, opening up previously inaccessible areas (Metroidvania-style progression).
- **Health & Energy Management:**
    - **Health Pickups:** Restore vitality mid-combat.
    - **Chi Regeneration:** Energy recharges slowly - use ranged attacks wisely.

4. Narrative & Dialogue System

- **Story-Driven Quests:** Master Guang and defeated bosses provide lore and guidance.

5. UI & Feedback

- **Health Bars & Damage Popups:** Clear visual feedback for attacks and healing.
- **Minimalist HUD:** Keeps focus on action while providing essential info.

Tools and Technologies Used

Game Engine, Plugins, and Development Tools

- **Game Engine:** Unity (2021+), chosen for its robust 2D support, asset pipeline, and extensibility.
- **Official Plugins/Features:**
    - **Unity Input System:** Modern input handling for keyboard and gamepad.

  - **TextMeshPro:** High-quality text rendering for UI and dialogue.
  - **Unity UI:** Flexible UI system for menus, health bars, and popups.
  - **Universal Render Pipeline (URP):** Enhanced graphics and performance.

- **Other Tools Used:**
  - **Google ImageFx:** Dialogue Image Generation.
  - **Visual Studio Code:** Scripting and code management.

Scripting Language Used for Development

- **Language:** C#
- **Reasons:** Unity's primary scripting language, offering object-oriented design, strong typing, and extensive documentation. C# enables efficient event-driven programming, integration with Unity's MonoBehaviour lifecycle, and easy management of game logic, UI, and physics.

Game Development

Game Features

1. **Player Inputs:** Movement (left/right), jump, small jump, run, melee attack, ranged attack, interact, pause, menu navigation.
2. **Player Features:** Health management, ability unlocks (jump, ranged attack), animation states (idle, run, attack, hit), knockback on damage, healing via pickups.
3. **Controls:** Keyboard and gamepad support, mapped via Unity Input System.
4. **Enemy AI:** Patrol logic, attack triggers, damage reception, death states, cooldowns for attacks.
5. **UI Features:** Health bars (player and boss), damage/heal popups, pause menu, dialogue system, main menu.

6. **Audio:** Background music, sound effects for attacks, pickups, and UI actions.
7. **Progression:** Scene-based ability unlocks, boss battles, and level transitions.

Scripts, Variables, and Methods Developed

AnimationStrings.cs

Purpose:

Centralizes string constants for animation parameter names, reducing typos and improving maintainability.

Key Concepts:

- Static string variables for animator parameters (e.g., "isGrounded", "attack").
- Used by character scripts to trigger or check animation states.

Attack.cs

Purpose:

Handles attack logic for characters, including hit detection and damage application.

Key Variables:

- References to attack zones or colliders.
- Damage values.

Key Methods:

- Methods to trigger attacks, check for collisions, and apply damage to targets.

BossDeathHandler.cs

Purpose:

Manages events and effects that occur when a boss character is defeated.

Key Variables:

- Reference to the boss object and related UI.

Key Methods:

- Handles boss death animation, disables boss controls, and triggers victory events.

BossHealthBar.cs

Purpose:

Displays and updates the boss's health bar UI.

Key Variables:

- Reference to the boss's health component.
- UI slider or bar.

Key Methods:

- Listens for health changes and updates the UI accordingly.

Damageable.cs

Purpose:

Provides health management for any object that can take damage or be healed.

Key Variables:

- Current and maximum health.
- Flags for invincibility or alive status.

Key Methods:

- Methods to apply damage, heal, check for death, and trigger related events.

DetectionZone.cs

Purpose:

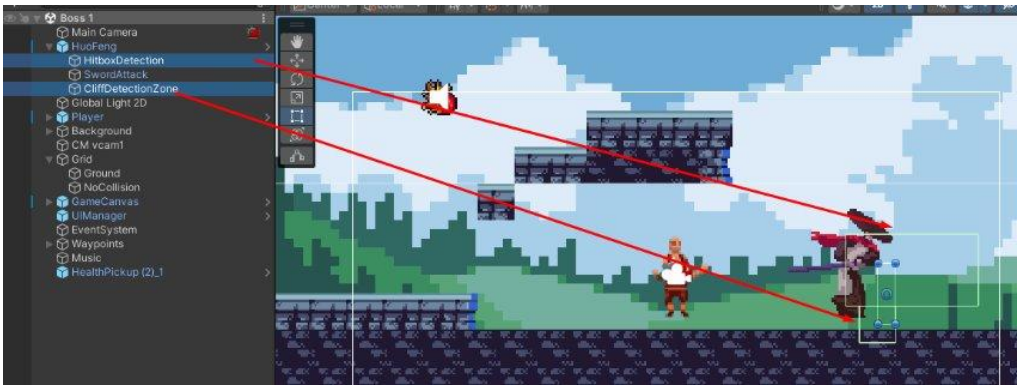
Detects when objects (e.g., player, enemies) enter or exit a defined area, often used for AI or triggers.

Key Variables:

- Collider references.
- List of detected objects.

Key Methods:

- Methods to handle OnTriggerEnter2D/OnTriggerExit2D events.



Dialogue/Dialogue.cs

Purpose:

Defines the structure for dialogue entries, including speaker, text, and possible choices.

Key Variables:

- Speaker name, dialogue text, options.

Dialogue/DialogueManager.cs

Purpose:

Controls the flow of dialogue sequences, including displaying text and handling player input.

Key Variables:

- Queue of dialogue entries.
- Reference to UI elements.

Key Methods:

- Methods to start, progress, and end dialogues.

Dialogue/DialogueTrigger.cs

Purpose:

Triggers dialogue sequences when the player interacts with certain objects or areas.

Key Variables:

- Reference to DialogueManager and dialogue data.

Key Methods:

- Method to initiate dialogue when triggered.

Events/CharacterEvents.cs

Purpose:

Defines and manages custom events related to character actions (e.g., health change, death).

Key Variables:

- UnityEvents for health, death, etc.

FlashTextAtStart.cs

Purpose:

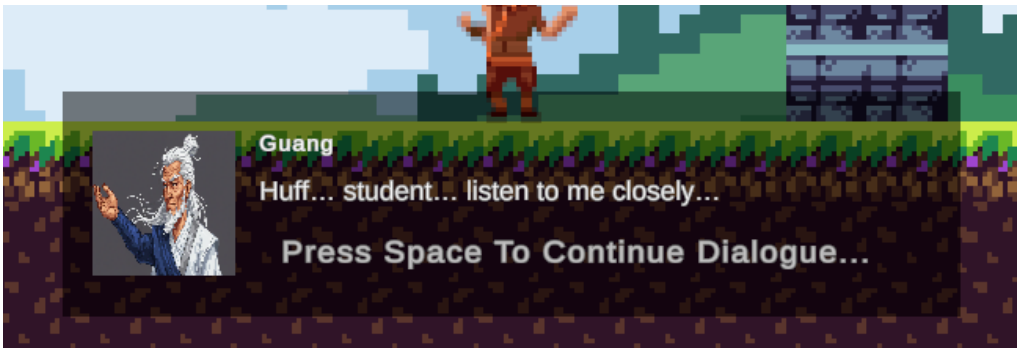
Displays flashing text at the start of a scene or level for player guidance.

Key Variables:

- Reference to UI text.
- Timing variables.

Key Methods:

- Coroutine to handle flashing effect.



FlyingEye.cs, Knight.cs, MasterGuang.cs, MovingObstacle.cs, HuoFeng.cs (Boss), ShuiLian.cs (Boss), LeiShan.cs(Boss), DarkSelf.cs (Boss) etc.

Purpose:

Enemy, obstacle and boss character scripts, each handling unique AI, movement, and attack patterns (Master Guang is idle character).

Key Variables:

- References to health, animator, and detection zones.

- AI state variables.

Key Methods:

- Methods for movement, attacking, taking damage, and state transitions.

HealthBar.cs, HealthText.cs

Purpose:

Display and update the player's health visually (bar or text).

Key Variables:

- Reference to player's health.
- UI elements.

Key Methods:

- Listens for health changes and updates UI.

HealthPickup.cs

Purpose:

Handles collectible health items that restore player health on pickup.

Key Variables:

- Amount of health restored.

Key Methods:

- OnTriggerEnter2D to detect player and apply healing.

MainMenuUI.cs

Purpose:

Manages main menu interactions, such as starting the game or quitting.

Key Variables:

- Main menu interface.

Key Methods:

- Methods for button actions such as start game, exit and switch levels.

MusicPlayer.cs

Purpose:

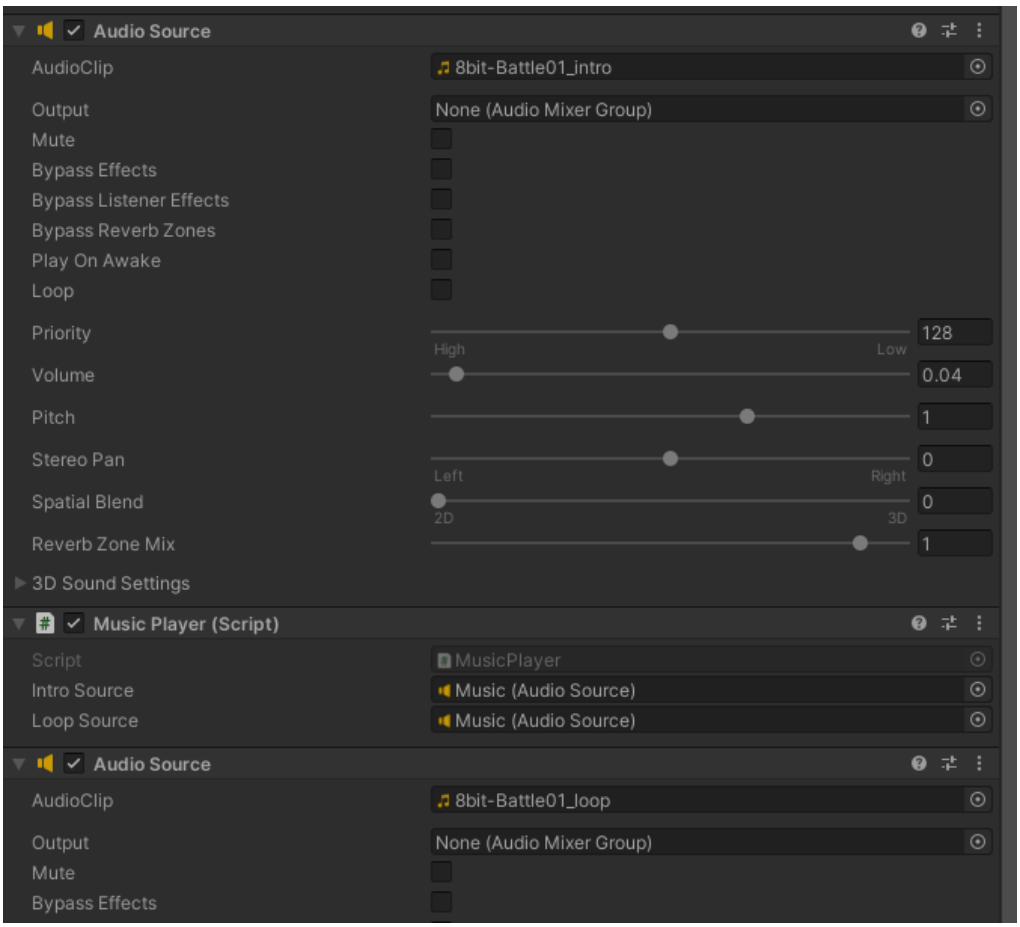
Handles background music playback and transitions between tracks.

Key Variables:

- AudioSource reference.
- List of music tracks.

Key Methods:

- Methods to play, stop, or switch music.



ParallaxEffect.cs

Purpose:

Creates a parallax scrolling effect for background layers to add depth.

Key Variables:

- Parallax speed.
- Reference to camera or player.

Key Methods:

- Updates background position based on player movement.

PauseManager.cs

Purpose:

Controls pausing and resuming the game, including showing/hiding pause menus.

Key Variables:

- Reference to pause menu UI.

Key Methods:

- Methods to pause, resume, and manage game state.

PlayerController.cs

Purpose:

Handles all player movement, input, abilities, and animation states.

Key Variables:

- Rigidbody2D, Animator, ability level, health.
- Input flags (jump, run, attack).

Key Methods:

- Methods for movement, jumping, attacking, taking damage, and ability unlocks.

Projectile.cs, ProjectileLauncher.cs

Purpose:

Manages projectile behavior (movement, collision, damage) and launching logic.

Key Variables:

- Speed, direction, damage.

Key Methods:

- Methods to launch, move, and handle collisions.

StateMachine/ (FadeRemoveBehaviour, PlayOneShotBehaviour, SetBoolBehaviour, SetFloatBehaviour)

Purpose:

Custom animation state machine behaviors for triggering effects or parameter changes during animations.

Key Methods:

- Methods to set animator parameters or trigger effects at specific animation states.

TouchingDirections.cs

Purpose:

Detects which directions the player or enemy is touching (ground, wall, etc.), used for movement and animation logic.

Key Variables:

- Flags for ground, wall, ceiling contact.

Key Methods:

- Methods to update contact states.

UIManager.cs

Purpose:

Central manager for UI elements, including popups, health bars, and menus.

Key Variables:

- References to UI prefabs and canvases.

Key Methods:

- Methods to display popups, update UI, and handle UI-related events.

Feature Implementation Methods

1. Player Movement, Abilities, and Animation

Procedures:

- The PlayerController script manages all player movement, input handling, and ability logic. It utilizes Unity's Rigidbody2D for physics-based movement and checks player input via the Unity Input System.
- Abilities like high jump or chi attack are unlocked based on the current scene. This is achieved by parsing the scene name and setting ability levels accordingly.
- Animation transitions are controlled by setting parameters in the Animator component, with parameter names stored in AnimationStrings.cs to prevent typos.

Scripting Components:

- PlayerController: Handles movement, input, and ability logic.
- TouchingDirections: Detects ground and wall contact to influence movement and animation states.
- Animator: Controls animation states based on parameters.
- AnimationStrings: Centralizes all animation parameter names for consistency.

Optimizations:

- Centralized animation parameter names reduce errors and simplify maintenance.
- Scene parsing for ability unlocks is automated, reducing manual configuration per scene.
- Physics and animation updates are decoupled to optimize processing, avoiding unnecessary calculations each frame.

2. Combat, Damage, and Health

Procedures:

- The Attack script manages attack logic, including collision detection with enemies via trigger zones.

- The Damageable component is attached to entities that can take damage or be healed, managing health, invincibility frames, and death logic.
- Health changes trigger UnityEvents, which are listened to by UI scripts (HealthBar, BossHealthBar, UIManager) to update visual feedback.

Scripting Components:

- Attack: Handles attack execution and hit detection.
- Damageable: Manages health, damage intake, healing, and death.
- HealthBar / BossHealthBar: Update the UI to reflect current health.
- CharacterEvents: Custom UnityEvents for health updates and death notifications.

Optimizations:

- UI elements update only when health changes, not every frame, reducing overhead.
- Damage popups are instantiated only when necessary, limiting UI clutter.
- Invincibility frames prevent multiple damage hits in rapid succession, optimizing damage handling.

3. Enemy AI and Boss Logic

Procedures:

- Enemy scripts (FlyingEye, Knight, HuoFeng, MasterGuang, MovingObstacle, ShuiLian) operate via state machines managing patrol, attack, and idle behaviors.
- Detection zones (DetectionZone) trigger state changes when the player enters or leaves specific areas.
- Bosses utilize BossDeathHandler to manage death sequences, UI updates, and scene transitions.

Scripting Components:

- Enemy AI scripts implementing state machines.

- DetectionZone: Uses Unity physics events for efficient area detection.
- BossDeathHandler: Manages boss-specific death events and transitions.

Optimizations:

- State machines minimize unnecessary checks by executing logic only relevant to the current state.
- Detection zones leverage Unity's physics event system, making triggers efficient and event-driven.

4. UI, Menus, and Popups

Procedures:

- The UIManager oversees all UI elements, including health bars, damage/heal popups, and menus.
- The PauseManager handles pausing, resuming, and toggling the pause menu.
- The MainMenuUI manages main menu interactions like starting or quitting the game.

Scripting Components:

- UIManager: Centralized UI control.
- PauseManager: Manages game pause/resume states.
- MainMenuUI: Handles main menu interactions.
- HealthText: Displays floating text for damage/heal feedback.

Optimizations:

- UI popups are created only when needed and destroyed after use to optimize memory.
- Menu transitions utilize coroutines for smooth animations.
- Platform-specific exit logic ensures proper behavior across different build targets.

5. Dialogue System

Procedures:

- DialogueManager manages dialogue flow, queuing dialogue entries and displaying them with a typing effect.
- DialogueTrigger initiates dialogue when the player interacts with objects or NPCs.
- Dialogue data structures (Dialogue) store dialogue content flexibly.

Scripting Components:

- DialogueManager: Controls dialogue display and flow.
- DialogueTrigger: Handles interaction-based dialogue initiation.
- Dialogue: Data structure for dialogue content.

Optimizations:

- Dialogue UI is activated only during conversations, reducing rendering overhead.
- Typing effects are implemented via coroutines, ensuring smooth and non-blocking text display.

6. Audio and Parallax

Procedures:

- MusicPlayer manages background music, switching tracks as needed.
- ParallaxEffect creates layered background movement by moving sprites at different speeds relative to the camera.

Scripting Components:

- MusicPlayer: Controls background music playback and switching.
- ParallaxEffect: Implements parallax scrolling based on camera movement.

Optimizations:

- Audio sources are reused rather than recreated, conserving resources.
- Parallax calculations are only performed when the camera moves, reducing unnecessary computations.

7. Obstacles and traps

Procedures:

- MovingObstacles control obstacle movement along predefined paths or patterns which cause damage to player while being in contact.
- TrapTiles appear as part of the environment but cause player to fall when stepped on.

Scripting Components:

- MovingObstacles: Updates position based on movement logic and handles collision detection, damageable and others with the player.
- TrapTiles: Detect player movement and trigger trap behaviour such as disabling colliders or playing collapse animations.

Optimizations:

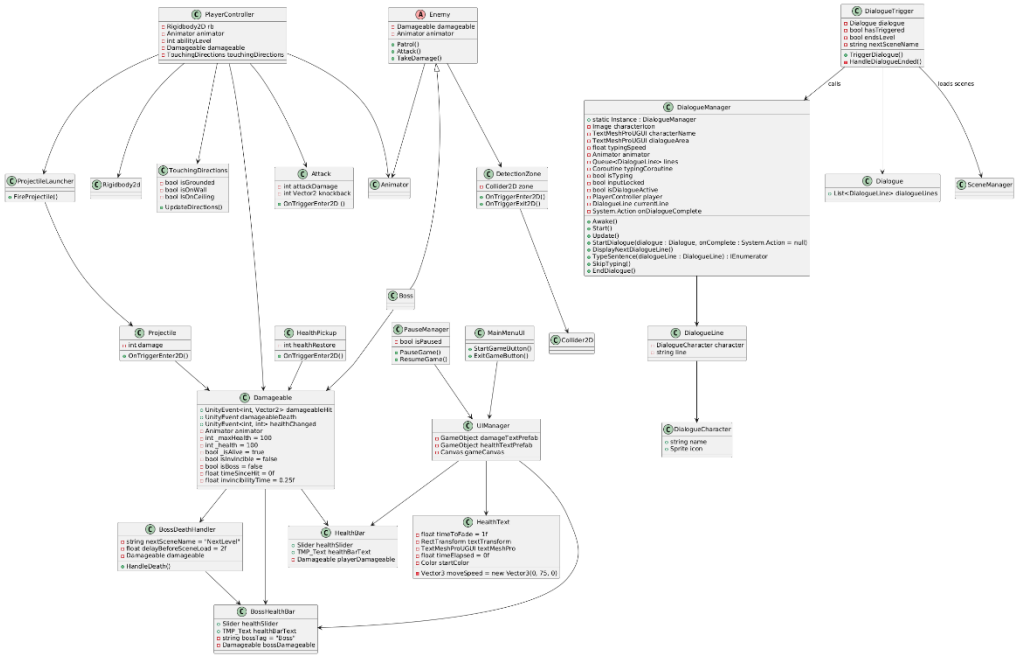
- Movement paths are calculated with simple math functions instead of complex physics.
- Trap triggers are event-based, activating only when the player interacts, reducing unnecessary update checks.

8. General Optimization Strategies

- **Event-driven architecture:** Utilizing UnityEvents or C# events to react to changes, avoiding continuous polling.
- **Component-based design:** Each script has a single responsibility, enhancing modularity and maintainability.
- **Pooling and instantiation:** UI elements and popups are instantiated only when necessary and reused or destroyed afterward to save memory.
- **Centralized constants:** Animation and input string names are stored in dedicated classes/files to prevent bugs and facilitate refactoring.

UML Class Diagrams of Classes and Objects

(Some variables, functions and objects were omitted for brevity)



Summarised Implementation of Graphics, UI, Audio, and Settings

- **Graphics:**
  - Sprites sourced from free online resources and made into animation clips.
  - Animation frame counts vary by character and action and thus are not fixed (idle: 4-6 frames, run: 6-8 frames, attack: 8-12 frames – just for example).
- **UI:**
  - TextMeshPro for crisp, scalable text.
  - Unity UI system for health bars, menus, and popups.
  - Canvas scaling ensures UI adapts to different resolutions (target: 1920x1080).
- **Audio:**
  - Music in .ogg/.wav format, optimized for looping and file size.
  - SFX for attacks, pickups, and UI actions
- **Settings:**
  - Resolution targets 1920x1080.
  - UI colors and fonts chosen for readability and thematic consistency.

Game Assets and Items

List of Assets

Graphical Assets References

- Character Portraits for Dialogue (Generated by Google ImageFX): https://labs.google/fx/tools/image-fx
- Art Assets:
    - RVROS Adventurer (Player Sprite): art/rvros-adventurer
    - Pixel Food (Health Pickups): https://henrysoftware.itch.io/pixel-food
    - RPG Icon Pack: https://kyrise.itch.io/kyrises-free-16x16-rpg-icon-pack
    - Monsters and Creatures (Enemies): https://luizmelo.itch.io/monsters-creatures-fantasy
    - Martial Hero 1 https://luizmelo.itch.io/martial-hero
    - Martial Hero 2 https://luizmelo.itch.io/martial-hero-2
    - Knight Sprite Sheet (Lei Shan and fireball): https://assetstore.unity.com/packages/2d/characters/knight-sprite-sheet-free-93897
    - Fantasy Knight (Shadow Knight): art/freeknight
    - Slimes: art/Slimes
    - ForestTilesets: art/freecutetileset
    - DarkForestTileset: https://atari-boy.itch.io/dark-forest-platform-pack
    - JungleTileset : https://jesse-m.itch.io/jungle-pack
    - CavesTileset: https://szadiart.itch.io/pixel-fantasy-caves

Audio Assets References

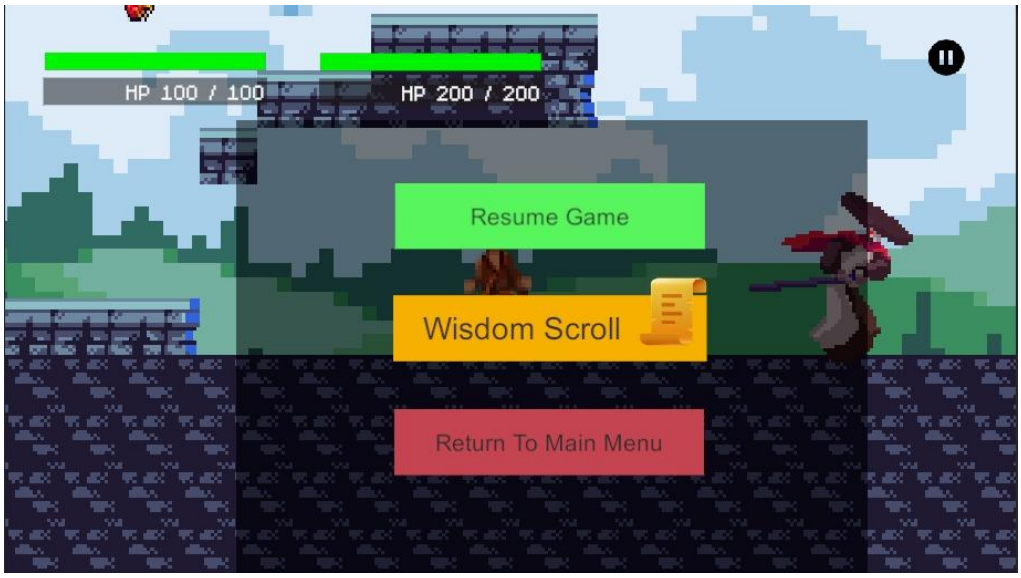
- Background Music:
    - audio/music/10 Battle1 (8bit style) - https://youfulca.itch.io/legendary-jrpg-battle-music-pack
- Sound Effects:

- Assets/Audio/SFX/RPG_Essentials_Free: https://leohpaz.itch.io/rpg-essentials-sfx-free

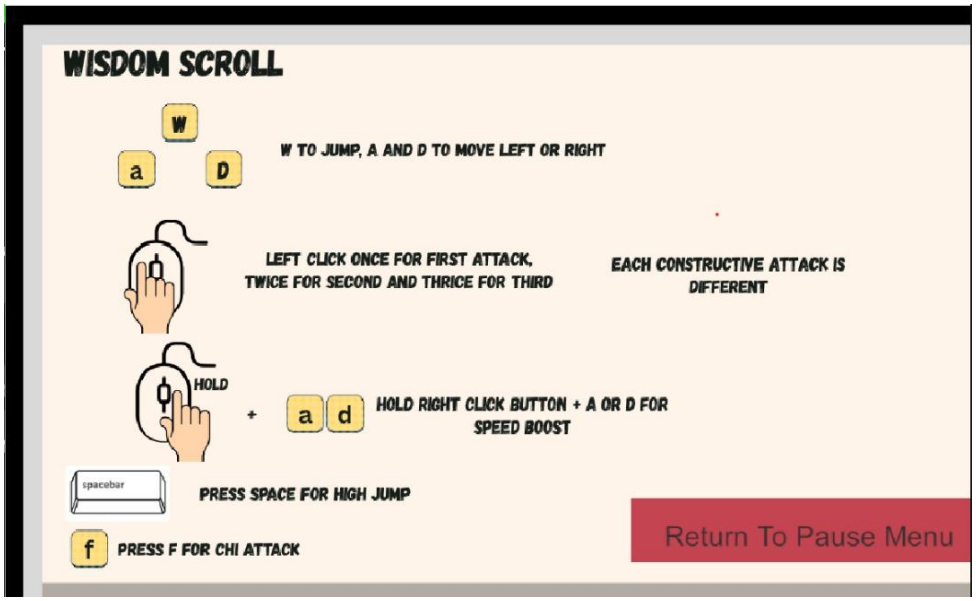
List of UI Items

- **Main Menu:** Start game, level button and exit game.

- **Pause Menu:** Resume, wisdom scroll, return to main menu.

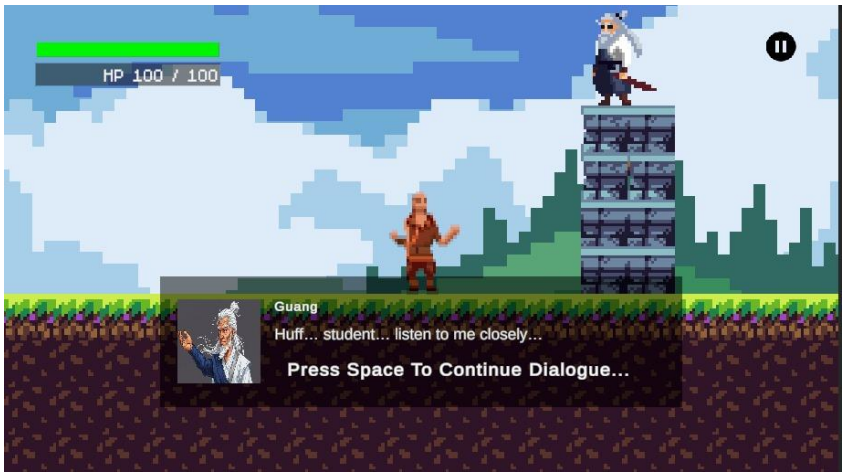


- **Widsom Scroll:** There are some moving, attack and skill tutorials

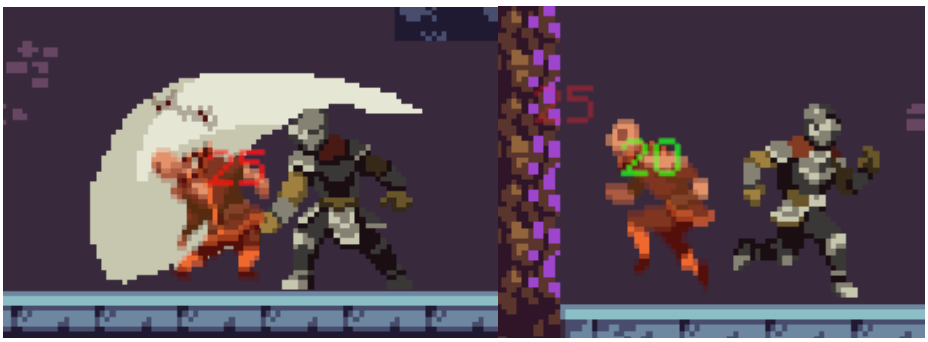


- **Health Bars:** Player and boss.

- **Dialogue Box:** Character name, portrait, text area.



- **Damage and Health Popups:** Floating text for feedback.

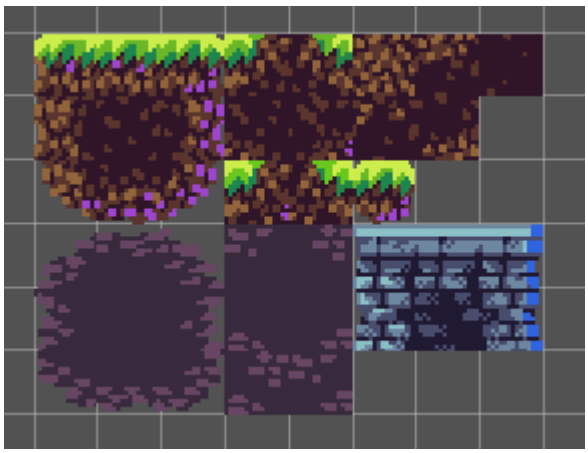


Game Level

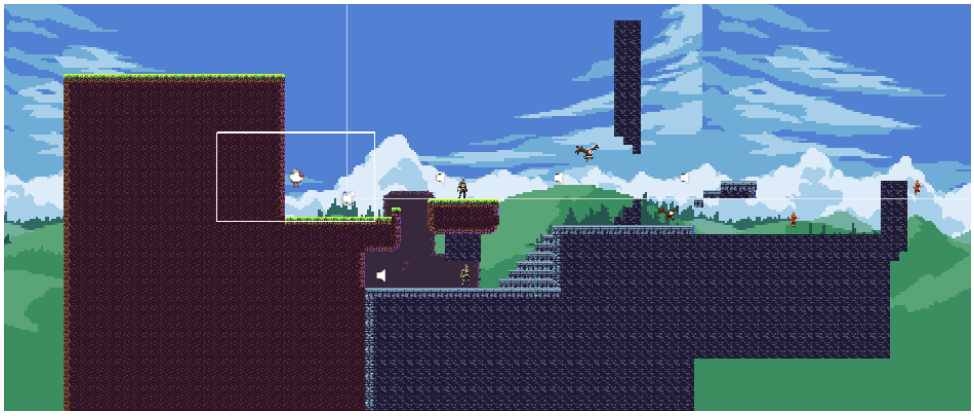
Level 1

- **Boss:** HuoFeng

- **Boss skill:** SpeedBoost
- **Player Current Skill:** Normal attack
- **Tileset:** ForestTileset



- **Tilemap:**



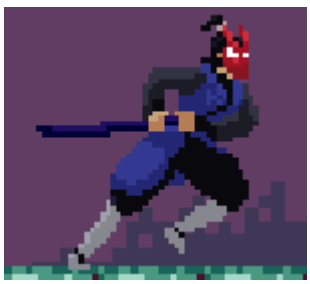
- **Enemy Types**: Knights, Flying Eye

- **Items:** Health Pickup (Fruit)

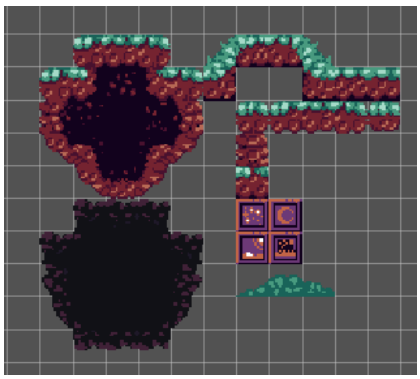
- **Description:** This level is the player's first challenge, featuring a few enemies and a boss named HuoFeng. HuoFeng possesses incredible agility and movement speed, and players must defeat him to learn his skill.

Level 2

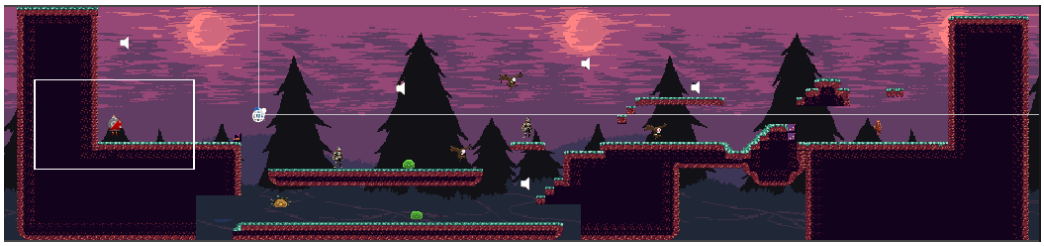
- **Boss:** ShuiLian



- **Boss skill:** Speed Boost, High Jump
- **Player Current Skill:** Normal attack and SpeedBoost
- **Tileset:** DarkForestTileset



- **Tilemap:**

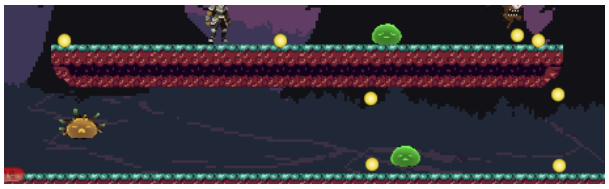


- **Enemy Types**: Knights, Flying Eye and Moving Obstacles (Slime)




- **Items:** Health Pickup (Fruit)

- **Special Mechanics**: Moving Obstacles

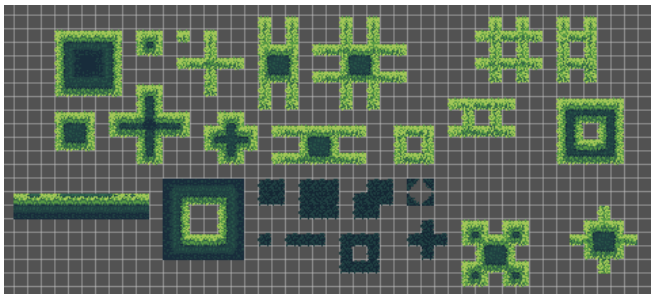


- **Description:**
  - New added moving obstacle performs damageable to player which get in touch with it. Player should make good use of speed boost skill that gained from level 1 to avoid the damage of obstacles.
  - Player gained high jump skill after defect ShuiLian boss by a lesson tutorial.

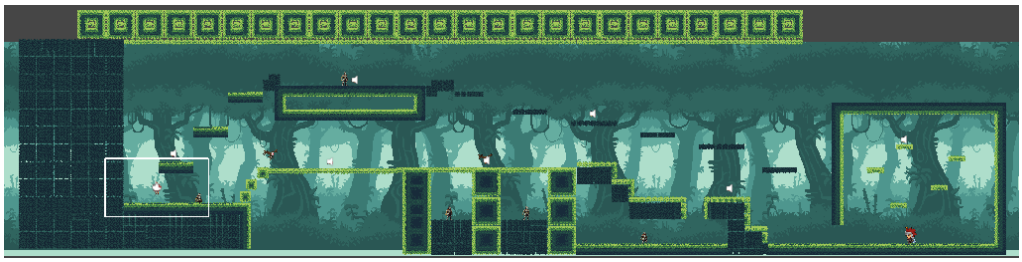
Level 3

- **Boss:** LeiShan

- **Boss skill:** Speed Boost, High Jump, Chi Blast
- **Player Current Skill:** Normal Attack, Speed Boost and High Jump
- **Tileset:**



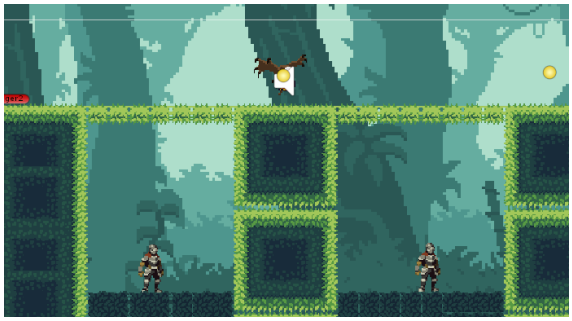
- **Tilemap:**



- **Enemy Types**: Knights, Flying Eye

- **Items:** Health Pickup

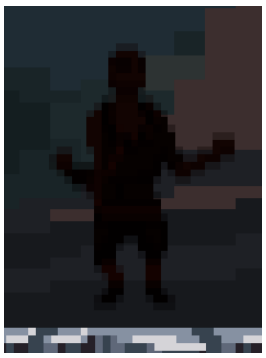
- **Special Mechanics**: Tilemap Traps



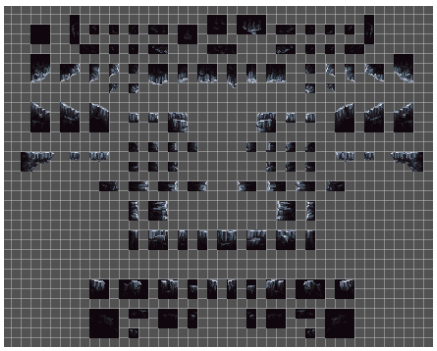
- **Description:** New special mechanics tilemap traps will appears as look as a part of the tilemap. It will cause player to fall while stepped onto it.

Level 4

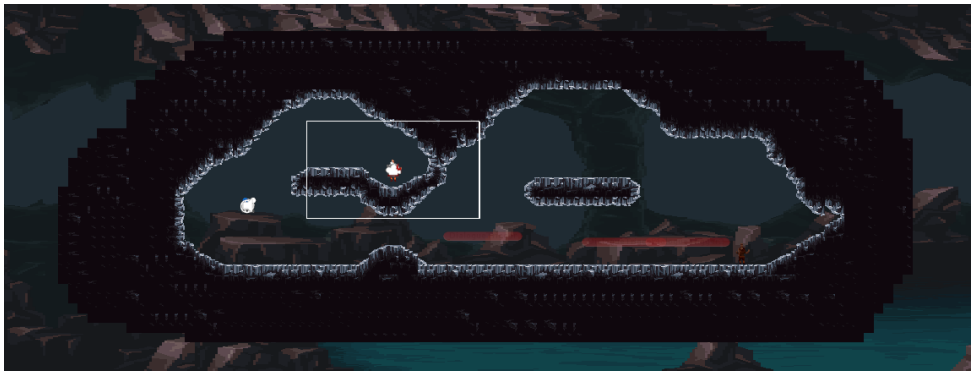
- **Boss:** DarkSelf



- **Boss skill:** SpeedBoost, HighJump, ChiBlast
- **Player Current Skill:** SpeedBoost, HighJump, ChiBlast
- **Tileset:** CavesTileSet



- **Tilemap:**



- **Items:** Health Pickup (Fruit)

- **Description:** In this cave, the player encounters a darkened version of themselves, born from all their fears and weaknesses. To escape alive, they must confront and overcome their own shadow.

Appendix

Appendix A - Core Story Dialogue

Story Opening Dialogue

Character	Dialogue
Master Guang:	Huff… student… listen to me closely…
Player:	Master Guang! You're hurt! Who did this?
Master Guang:	It was… the Shadow Lord. He's far stronger than I expected… I tried to stop him… but I have failed.
	When he emerged victorious, he attained balance. He became the Protector of Harmony and guided us all.
	Under his teachings, we four siblings found our paths.
	Huo Feng, the Master of Speed. His steps are like wind and fire. Shui Lian, the Master of the Soaring Leap. His strength lies in His calm and fluid movements. Lei Shan, the Master of Chi. His strikes echo like thunder, his energy unmatched.
	But now… they have all fallen. Twisted by the Shadow Lord's dark power. They serve him… as his generals.
Player:	Then we must save them! Together!
Master Guang:	No… I am already at death's door. My duty is to protect our village from the Shadow Lord's minions… I must go alone.
	Your task is greater. You must seek out my siblings. Defeat them in combat, free them from darkness… and learn their techniques.

	Only then will you be ready to enter the Cave of Reflection and face your shadow… as Tian Yun once did.
Player:	But Master… I'm not ready!
Master Guang:	You have more potential than you know. Trust yourself… trust the path.
	The road ahead is dangerous. The Shadow Lord's underlings will hunt you. You must master Kung Fu, or the world will fall.
	Go now! The world depends on you…

BOSS 1: Huo Feng – The Master of Speed (Corrupted)

Character	Dialogue
Huo Feng:	You dare approach me? Foolish child. I am the wind that burns!
Player:	Master Huo Feng! Snap out of it! The Shadow Lord is controlling you!
HuoFeng:	The Shadow Lord showed me truth. Speed is power, and I shall crush you with it! (After boss defeated)
	Urgh... what... what have I done? My mind was clouded... poisoned by shadows.
Player:	Master Guang sent me. He said only by learning from you can I grow stronger.
Master Guang:	Guang... always the wisest. Then listen well.

Huo Feng's Speed Skill Tutorial

Character	Dialogue
Huo Feng:	To move like the wind, you must learn to Speed Boost.

	Hold the right click button and move to Speed Boost.
	Use it to dodge attacks, outrun danger, or confuse your enemies. However, be careful of heights. Speed is your ally. Waste not a moment.

BOSS 2: Shui Lian – The Master of the Soaring Leap (Corrupted)

Character	Dialogue
Shui Lian:	You should have turned back, child. The shadows whisper, and I obey.
Player:	Shui Lian! You taught me calm. This is not you!
Shui Lian:	I am the lotus no longer. Only the drowning pull of darkness remains! (After boss defeated)
	...I remember... the wind... the waterfall... What have I become?
Player:	I need your help. The world needs your help.
Shui Lian:	Then let me pass on what remains.

Shui Lian's High Jump Tutorial

Character	Dialogue
Shui Lian:	The path ahead rises high. You must learn to rise with it.
	To High Jump, press space
	Use it to scale cliffs, avoid ground traps, and escape from chaos.
	Leap not with legs, but with spirit.

BOSS 3: Lei Shan – The Master of Chi (Corrupted)

Character	Dialogue

Lei Shan:	Another challenger? Good. I've grown hungry for destruction.
Player:	No! This isn't you. I need your wisdom, not your wrath!
Lei Shan:	Wisdom is a lie. Only power remains! (After boss defeated)
	The thunder fades... What madness overtook me?
Player:	You were controlled. I need to learn from you – to finish what Tian Yun began.
Lei Shan:	Then listen, for chi flows only through the worthy.

⚡ Lei Shan's Chi Blast Tutorial

Character	Dialogue
Lei Shan:	Chi is life. Chi is force.
	To release it, press F. That is your Chi Blast.
	It consumes your inner energy. Your energy bar recharges slowly as you move.
	Use it to strike enemies from afar, hit switches, or destroy barriers.
	But beware—use it wisely, or be left powerless.

🌑 Cave of Reflection (Final Level Opening Dialogue)

Character	Dialogue
Player:	This is it… the Cave of Reflection. Where Master Tian Yun achieved enlightenment. (He walks deeper…)
	It feels strange… like I'm being watched... (A dark figure steps out – identical to the player)
	W-What... That's... me? (More shadows appear—exact copies, moving in sync)

	This will be the most powerful opponent I will ever face.
	My shadow... all my fears… every weakness I tried to hide…
	I must defeat them all… or be lost in the darkness forever.

Story Ending Dialogue

Character	Dialogue
Master Guang:	Congratulations, student! You have faced great trials and overcome the darkness that once consumed my siblings. Through your strength, courage, and spirit, you have restored balance to our land.
	You have mastered the ancient techniques and proven yourself worthy of the title Protector of Harmony. But remember, the true battle is never over — keep your heart steady, your mind sharp
	The Shadow Lord is defeated for now, but evil always lurks in the shadows. Stand vigilant, for you are the light that guards this world.
Player:	Thank you, Master Guang. I will carry your teachings and protect our home.
Master Guang:	I believe in you, student. Now, go forth and live as the hero you were meant to be.