
Game Design Document: Frog's Adventure

---

1. Game Summary

1.1. What is the Game?

Frog's Adventure is a 2D platformer with pixel-art graphics. Players control a hero frog, running, jumping, and fighting enemies. As the game progresses, our frog gets stronger by learning new abilities from a Skill Tree.

1.2. Genre

2D Adventure & Platformer

1.3. Who is this Game For?

For anyone who loves fun, challenging, and old-school style platformer games.

---

2. Story and World

2.1. Story

The peaceful Lilypad Kingdom is attacked by the evil Dragon Lord, Drako. Drako kidnaps Princess Lily, the source of the kingdom's magic, and takes her to his castle. The kingdom's only hope is its bravest warrior, our frog, Frobert.

2.2. Characters

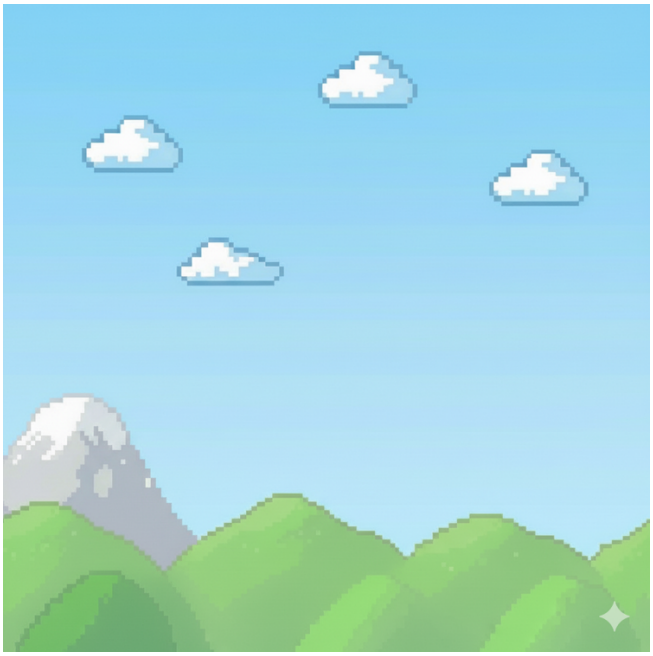
- **Frobert (The Hero):** Our brave frog who starts his adventure with only a simple jump.
- **Princess Lily (The one to be rescued):** The heart and soul of the kingdom.
- **Drako (The Villain):** An evil Dragon Lord who loves power. He is the final boss of the game.

2.3. Game Worlds

The game is split into 3 different worlds. Each world will introduce new mechanics, puzzles, and enemies.

World 1: The Whispering Forest

Focus: Teaches basic controls, Cherries, Falling Platforms, and the first Skill Tree unlocks.

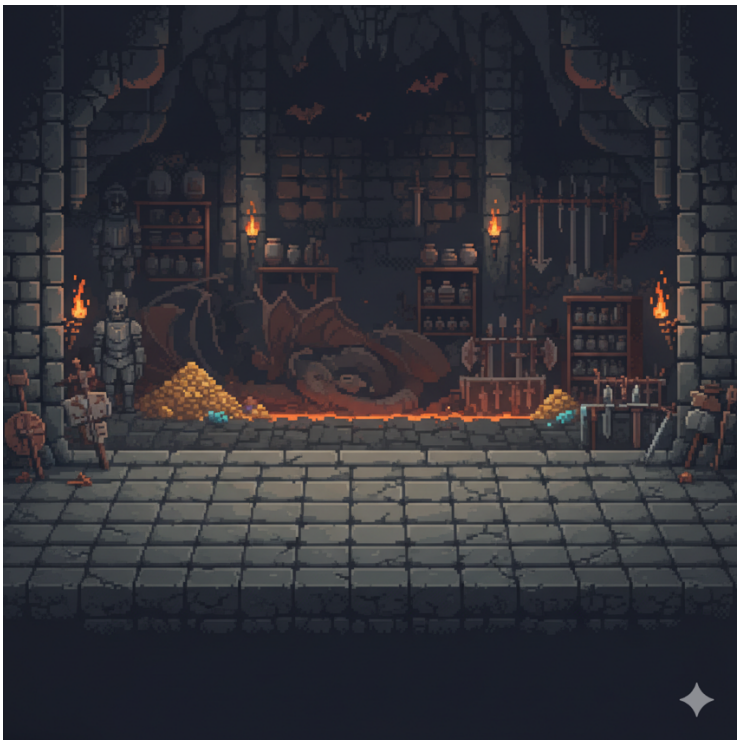


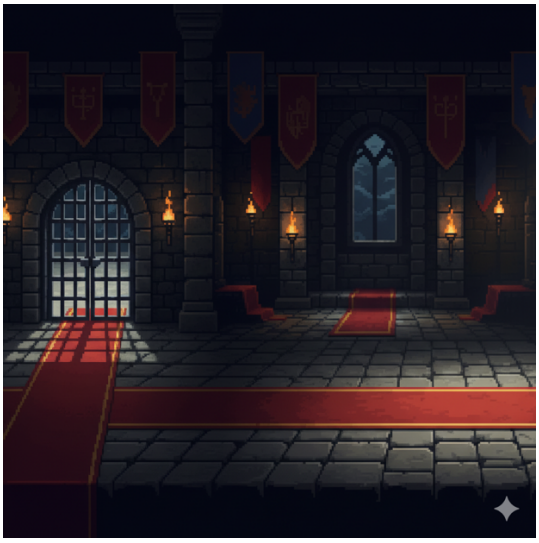
World 2: The Pink Lands

Focus: Introduces Fruit Power-Ups (like the Banana and Strawberry) and their related puzzles. Introduces Timed Platforms and Wind Lifts.

World 3: The Dragon's Castle

This world is designed to be the final, ultimate test of all the player's skills.





Example Levels:

- **"The Dark Labyrinth" (e.g., Level 13):** The first level inside the castle. The level is almost completely dark. The only light source is a small radius of light around Frobert. The player must navigate carefully to avoid traps and enemies hidden in the darkness.

- **"The Escape" (e.g., Level 14):** A "Time Limit" level. After the player defeats a mini-boss or hits a major switch, a timer starts (e.g., 90 seconds). The player must race to the exit as the castle starts to collapse or fill with lava. This is a high-intensity parkour sequence that combines all mechanics (falling platforms, timed jumps, etc.) under pressure.

- **"The Throne Room" (Level 15):** The final boss fight with Drako.

---

3. Gameplay

3.1. How to Play?

1. Start the level.
2. Get past obstacles and defeat enemies.
3. Collect **Cherries**, which are used as money.
4. Collect other **Fruits** to have special power-ups.
5. Finish the level.
6. Go to the **Skill Tree** and buy new abilities with your cherries.
7. Start the next level, now even stronger!

3.2. The Frog's Abilities

This section details the player's full potential. The current prototype (.cs files) includes a robust set of movement and combat mechanics. **In the final game, most "Advanced" and "Combat" abilities listed below will be locked by default and must be unlocked via the "Frog's Path" Skill Tree (see Section 4).**

- **Starting Ability:**
  - **Jump:** The player starts the game with only one heart (health) and can only jump.
- **To Be Learned from the Skill Tree:**
  - **Movement Abilities:** Wall Jump, Double Jump, Dash.

    - **Attack Abilities:** Finite Shuriken (Throwing Star), Infinite Shuriken.
    - **Other Abilities:** Extra Health.

Currently Implemented Core Mechanics:

- **Core Movement (PlayerMovement.cs)**
    - **Horizontal Movement:** Standard running speed.
    - **Jumping:** A responsive base jump.
    - **Variable Jump Height:** The player can control jump height by releasing the jump key, allowing for short and long hops. (Coded in PlayerMovement.cs > Update()).
    - **Coyote Time:** A "game feel" feature. The player can still jump for a very short time after walking off a ledge, making platforming more forgiving. (Coded in PlayerMovement.cs > coyoteCounter).
    - **Sprite Flipping:** The character sprite correctly flips based on the direction of movement.
- **Advanced Movement (To be locked by Skill Tree)**
    - **Wall Jump:** The ability to slide down walls and jump off them. (Coded in PlayerMovement.cs > onWall() and Jump()).
    - **Double Jump:** The ability to perform a second jump in mid-air. (Coded as extraJumps in PlayerMovement.cs).
- **Combat (To be locked by Skill Tree)**
    - **Fireball Attack:** A ranged attack using a projectile. (Coded in PlayerAttack.cs).
    - **Attack Cooldown:** The player must wait a short time between attacks. (Coded in PlayerAttack.cs > cooldownTimer).
    - **Projectile Pooling:** The game efficiently reuses fireball objects instead of creating new ones, preventing lag. (Coded in PlayerAttack.cs > FindFireball()).
    - **Attack Stance:** The player can only attack when standing still on the ground, adding a "risk vs. reward" element. (Coded in PlayerMovement.cs > canAttack()).

3.3. Things to Collect

- **Cherry:** The currency used to buy new abilities.
- **Heart:** A heart found in levels that instantly restores one heart.
- **Pineapple:** A special item. There are 3 hidden pineapples in each world (9 in total).
- **Other fruits:** Fruits that temporarely allow player to have different kind of powers.

3.4. Temporary Power-Ups (Fruit Abilities)

In addition to the permanent Skill Tree, players can find special fruits within the levels. These grant a powerful, timed ability (e.g., 10 seconds) used to solve specific puzzles.

- **Banana (Shrink):** The player becomes tiny for 10 seconds, allowing them to access small tunnels and secret passages that are otherwise too small to enter.
- **Strawberry (Invisibility):** The player becomes invisible for 10 seconds. This allows them to pass by enemies without being seen or, more importantly, to bypass "security eye" traps that would normally trigger alarms or doors.
- **Kiwi (Magnet):** For the rest of the level, the player has a special magnet feature that attracts all the cherries to itself.
- **Heavy Watermelon:** For 10 seconds, the player becomes heavy. This allows them to slam down on "cracked" floors to break them or to activate heavy-duty pressure plates.

3.5. Parkour Elements & Puzzles

The levels will be filled with a mix of platforming challenges and puzzles that test the player's skills and new abilities.

- **Falling Platforms:** Platforms that look stable but fall a second after the player steps on them.
- **Timed Platforms:** Platforms that appear and disappear in a set pattern, requiring precise timing.
- **Moving Platforms:** Platforms that move up and down or left to right.
- **Wind/Air Lifts:** Specific zones that push the player upwards (like a fan) or sideways, creating complex parkour sections.
- **Switch & Lever Puzzles:** Players may need to hit a lever to open a door, which might be on a timer. This could be combined with a power-up (e.g., hit a switch, then use a Banana to shrink and get through the closing door just in time).
- **Time limited levels**
- **Completely Dark Levels**

3.6. Core Systems (Health, UI, and Level Flow)

These systems are already implemented and form the backbone of the game experience.

- **Health & Damage System (Health.cs)**
  - **Health Hearts:** The player has a starting health value. (This will be modified by the Skill Tree to start at 1).
  - **Invincibility Frames (iFrames):** After taking damage, the player flashes and is immune to further damage for a short duration. (Coded in Health.cs > Invulnerability() coroutine).
  - **Death & Respawn:** When health reaches zero, a death animation plays, and components are disabled. The player can be respawned. (Coded in Health.cs > TakeDamage() and Respawn()).
- **Checkpoint System (PlayerRespawn.cs)**
  - The player's progress is saved when they touch a "Checkpoint" trigger.
  - Upon death, the player respawns at the last activated checkpoint with full health.
- **UI & Game Flow (UIManager.cs)**
  - **Pause Menu:** The player can pause the game (Time.timeScale = 0) at any time using the 'Escape' key.
  - **Game Over Screen:** A "Game Over" screen is shown when the player dies without a checkpoint.
  - **Level Navigation:** The UI provides functions to Restart the level, return to the Main Menu, or Quit the game.
  - **Level Selection Menu (LevelSelectManager.cs , LevelButton.cs):** This screen allows the player to navigate between worlds and choose which level to play. It visually tracks which levels have been completed and which collectibles (e.g.,Cherries, Pineapples) have been found.

- **Level Presentation (LevelNameDisplay.cs)**
  - At the start of each level, the scene name is elegantly displayed to the player with a fade-in and fade-out effect.

Level Structure

• **Room-Based Camera (CameraController.cs):** The camera smoothly follows the player *between* rooms, moving from one "room" (defined by a Transform) to the next.
• **Room Activation (Room.cs & Door.cs):** To save performance, enemies in a room are only activated when the player enters that room. This system is controlled by 'Door' triggers.

---

4. Progression System: "The Frog's Path"

The system that makes the player stronger is the **Skill Tree**.

4.1. How to Get a New Ability?

There are **two conditions** to unlock a new ability:

1. You must have finished a specific level (Example: "Complete Level 3").
2. You must have enough Cherries to buy it.

4.2. Sample Skill Tree

- **SURVIVAL:** 1st Extra Health (Requires: Level 2, 50 Cherries)
- **MOVEMENT:** Wall Jump (Requires: Level 5, 100 Cherries)
- **COMBAT:** Finite Shuriken (Throwing Star) (Requires: Level 3, 75 Cherries

---

5. Enemies and Bosses

5.1. Currently Implemented Enemies & Hazards

The project includes a wide variety of enemies and traps that are fully functional. These will be distributed across the 3 Worlds to create a smooth difficulty curve.

Static & Dynamic Traps

- **Arrow Trap (ArrowTrap.cs):** A trap that fires projectiles on a set timer, using an object pool for its arrows.
- **Firetrap (Firetrap.cs):** A trap that activates with a short delay when the player steps on it, creating a temporary damage zone.
- **Spikehead (Spikehead** A "smart" trap that uses Raycasts to check for the player and charges at them in one of four directions.

- **Spikes:** A trap that when the player touches, it will damage the player.

Enemy Types

- **Patrolling Enemy (Enemy_Sideways.cs, EnemyPatrol.cs):** Simple enemies that move back and forth between two set points. The EnemyPatrol version includes an "idle" state for more complex behavior.
- **Melee Enemy (Enemy.cs):** Uses a**.cs):**
- BoxCast to detect the player in a short range. When the player is in sight, it stops patrolling and attempts a melee attack on a cooldown.
- **Ranged Enemy (RangedEnemy.cs):** Similar to the Melee Enemy, but attacks from a distance by firing projectiles.

5.2. Final Boss: Dragon Lord Drako

The fight with Drako will have multiple stages.

---

6. Secret Reward: The "True Ending" Room

This is the ultimate reward for players who find all secrets in the game.

6.1. How It Works

- There is a locked door at the beginning of the game, marked with a Pineapple symbol.
- The door's lock opens only when the player has collected **all 9 Pineapples**.
- However, what's inside the room depends on one other condition: the player must have also **completed the game** (defeated Drako and rescued Princess Lily).

6.2. What's Inside the Room? (The True Ending)

- A lot of cherries as a reward.
- If the player enters the room *after* finishing the game, they will find Princess Lily waiting inside.
- When the player approaches her, it triggers a special cutscene:
    1. The Princess thanks Frobert for his bravery.
    2. She leans in and kisses him.
    3. A magical effect plays (a flash of light, a sound effect).
    4. Frobert (the frog) transforms into a prince.
- This reveals the "True Ending" of the game: our hero was a prince under a curse all along, and the kiss breaks the spell.
- The scene ends with the new Prince and Princess together, followed by a "Happily Ever After" message.