Game Design Document (GDD)

[Working title]

## Table of contents

1. [Introduction](#_orjphw9foo5j)
   1. [Scope of the document](#_qkjo1wye2f7y)
   2. [Elevator pitch](#_agpucdberhxb)
2. [Game overview](#_dkwebavy3ce)
   1. [Game concept](#_6an3rbmlxrow)
   2. [Audience](#_wkp9bunr3dvr)
   3. [Genre](#_e1383bjfc05z)
   4. [Setting](#_wozggf5e2tgd)
   5. [Game structure](#_g2nvtbqiae1w)
   6. [Player](#_2d0g43qomr1h)
   7. [Game flow summary](#_86gcu1fxftyz)
   8. [Look & Feel](#_4dsgsqs8nnjg)
3. [Gameplay](#_m7ronjwofttr)
   1. [Objectives](#_lwjj4rwtxzqu)
   2. [Progression](#_w2sz5kyiu2n5)
      1. [Challenge structure](#_ga8x0klpxsk3)
   3. [Play flow](#_3w9ccow83ivt)
   4. [Difficulty](#_wo1bgeeqvlqe)
4. [Mechanics](#_9j0d9v6zcl2)
   1. [Rules](#_iun9evf3d82h)
   2. [Game universe](#_ixispjkzdmjq)
   3. [Physics](#_z5o213dbfk80)
   4. [Economy](#_pkfphbxfezsp)
   5. [Character movement](#_9vju0aimo0g1)
   6. [Player interaction](#_f6yfihxkqrke)
      1. [Game menus](#_p4kklzi0wjao)
      2. [Saving](#_j6cmxnfydse8)
      3. [Game options](#_wbt3g51bxnlw)
   7. [Assets](#_3adx767bv6ye)
5. [Graphics and audio](#_lbaxwgq732o5)
   1. [Visual system](#_h9q1yxrzhyyq)
      1. [Player camera](#_4bv6v7k3rxb8)
      2. [Landscape](#_bhee23kblrrx)
   2. [Interface](#_54bpfjn2f06d)
   3. [Audio system](#_wo4jsux84lab)
      1. [Game music](#_sm3fy5iuxr7t)
      2. [Audio look & feel](#_c5ml65dxr45)
6. [Story and narrative](#_gxu7q4zfyumy)
   1. [Backstory](#_myz19qnriy61)
   2. [Main plot](#_gcyhfgb4ayrg)
      1. [Plot progression](#_995d9yvq0gtk)
   3. [Cutscenes](#_vaj95ksnsx6e)
7. [Characters](#_uif99gj0lmks)
   1. [Main characters](#_51ggerdw4yeb)
      1. [Backstory](#_r6hfv9ak03j)
      2. [Personality](#_e5qcg3hwmhr9)
      3. [Appearance](#_ragck7uynlpk)
      4. [Abilities](#_x80n1t8xanxs)
      5. [Relationships](#_bwfh0mb06gkg)
   2. [Supporting characters](#_rwg8g2xsmcda)
   3. [Enemies](#_o03mkkxww2v3)
8. [Game world](#_u9jmxqmgxbqs)
   1. [Look & Feel of the world](#_1rpcqba8eidm)
   2. [Locations](#_trwhbbxwhyy7)
      1. [Connection to the plot](#_e0rx5d6jlh8i)
   3. [Levels](#_p5dw1h21xnyg)
      1. [Tutorial level](#_pg8lln38kizu)
      2. [Main levels](#_w1blswmad7if)
      3. [Optional levels](#_g0qaq4cdno2r)

## 

## 

## 

## 

## 1. Introduction

### 1.1. Scope of the document

Just me.

### 1.2. Elevator pitch

By day, players forge social bonds in a Persona-style high-school simulator—unlocking and switching between multiple magical-girl avatars; by night, they descend into a shifting magical labyrinth beneath the city to tackle procedurally generated roguelike challenges, wielding each character’s unique abilities enhanced by the relationships they’ve built.

## 2. Game Overview

### 2.1. Game concept

**2.1.1 Overview:**

The city and its school are caught in a repeating one-year time loop. Players can spend each day in the city and school hub building social bonds to unlock and switch among multiple magical-girl avatars to play as in the roguelike section,or spend the day descending into a procedurally generated magical labyrinth beneath the city. Each successful run yields dungeon resources. Used in the hub to enhance abilities or unlock characters. The ultimate goal is to defeat the final evil at the bottom of the labyrinth before the year resets, thereby shattering the time loop.

**2.1.2 Player Feel:**

• **Community:** Players gradually form a living social circle. NPCs have reactive dialogue and schedules—commenting on the player’s actions, gifts, and successes—so relationships feel organic rather than static.

• **Risk-Reward/Cost Analysis:** Every action—choosing which NPC to spend time with, which gifts to give, which labyrinth paths to explore—carries trade-offs. Limited days, energy, and resources force the player to prioritize activities that best advance their bonds and empower their avatars for deeper, more challenging runs.

### 2.2. Audience

**Target Audience:**

**Primary:** Players aged approximately 15–30 who appreciate deep, reactive social-simulation systems (as in Persona-style relationship building) paired with challenging, action-oriented roguelike mechanics (as in Hades).

**Secondary:** Fans of magical-girl anime, modern urban settings, and school-based narratives who value meaningful trade-offs and emergent stories driven by player choice.

### 2.3. Genre

| **Hybrid Twin-Stick Shooter Roguelike / Social Simulation:**  – **Twin-Stick Shooter Roguelike:** Procedurally generated magical-labyrinth exploration, real-time twin-stick shooting combat a la Binding of Isaac, high risk-reward “Resolve” modifiers, and tactical resource choices in each run.  – **Persona-Style Social Sim:** Modern urban school and city hub, daily time-management, reactive NPC relationship building that unlocks new characters and abilities. |
| --- |

### 2.4. Setting

**Surface Hub:** A contemporary urban environment split between a modern city (streets, shops, landmarks) and a high-school campus within that city.

**Subterranean Labyrinth:** A shifting magical dungeon beneath the city that reconfigures each night.

### 2.5. World structure

**Hub World:**

• Split into two interconnected daily hubs—a modern city district (residential streets, shops, park) and a high-school campus (classrooms, courtyard, club rooms).

• Each sub-area hosts unique NPCs with reactive dialogue, interactive events, and opportunities to deepen social bonds.

**Roguelike (Labyrinth):**

• Procedurally generated multi-floor magical dungeon with branching paths and room-based encounters.

• Each floor culminates in a boss challenge; victory unlocks the next floor’s layout and yields dungeon resources. Used in the hub.

### 2.6. Player

**Single-player experience:**

• The player controls one magical-girl avatar per dungeon run, selected from a growing roster of characters. Additional avatars are unlocked by deepening social bonds during daytime hub activities. Characters can be swapped only in the hub between runs, and each run’s choice is locked in once the descent begins.

### 2.7. Core loop

The very basic actions the player takes when playing the game: Moving and shooting, running and jumping, reading and picking dialogue options, drawing and playing a card, etc.

**2.7 Core loop:**

The player alternates between two interlocking activity sets—Labyrinth Runs (roguelike) and City Hub (social sim):

**Roguelike (Labyrinth Runs):**

• Movement & Shooting – twin-stick navigation and fire

• Primary Ability – default magical-girl attack

• Secondary Ability – character-specific special attack

• Utility Action – dodge/dash or defensive skill

• Ultimate Ability – extremely powerful skill; consumes a limited number of charges that refresh on entering a new room

• Consumable Use – in-run items (healing, buffs, recalls)

• Dungeon Interaction – enter rooms, open chests, access NPC-boon rooms via relationship boons

**City Hub (Social Sim):**

• **Time Management:** Allocate limited daytime hours across activities

• **Conversation:** Talk with NPCs to increase bond levels

• **Gift Giving:** Spend NPC-preferred gifts to deepen relationships

• **Resource Allocation:** Spend dungeon-earned resources in the hub to unlock stuff

• **Activity Choice:** Choose which NPCs or events to prioritize each day, weighing risk versus reward across social and combat prep

### 2.8. Look & Feel

**Definitions:**

• **Look:** The game’s visual style—graphics, animations, color palettes, UI layout—acknowledging that look influences feel. (Use reference images from other games or media as needed.)

• **Feel:** The experiential side—playability, narrative/story elements, music, audio feedback, and overall emotional tone.

**Hub World:**

• **Visuals:** Soft, pastel color palettes; clean, uncluttered presentation; UI and environment elements emphasize clarity and comfort. Animations are smooth and deliberate, reinforcing a relaxed pace.

• **Audio/Story:** Gentle ambient music and subtle sound cues; narrative beats and NPC dialogue should feel warm and engaging.

• **Pacing:** Deliberate rhythms with clear downtime, encouraging exploration and community interaction. The contrast with the labyrinth builds anticipation for each run.

**Labyrinth):**

• **Visuals:** Darker, high-contrast palettes; dynamic environments; rapid visual transitions and sharper motion cues to heighten tension. Room layouts and glyphs remain instantly readable under pressure.

• **Audio/Feedback:** Impactful combat sounds and UI flashes; clear signals for damage, enemy actions, and resource pickups to support split-second decisions.

• **Pacing:** Fast-paced flow with minimal downtime, emphasizing urgency and risk.

**Core Feel Principles:**

• **Community:** Hub design makes NPCs and the world feel alive—reactive dialogue, story moments that change based on player actions.

• **Anticipation:** The peaceful hub versus intense labyrinth contrast drives excitement for dungeon runs.

• **Risk-Reward Dynamics:** Interface design highlights trade-offs—time remaining, resource counts, bond progression—so players can make informed choices at a glance.

## 3. Gameplay

### 3.1. Objectives

What is the main objective for the player? And what are the secondary objectives? Ex: The main objective is beating the final boss in the final level, the secondary objectives are fetching the hidden pieces in the earlier levels, discovering the story secrets through the map and beating the secret boss.

**3.1 Objectives**

• **Main Objective (Full Campaign):** Over the course of the one-year time loop, shatter the loop by defeating the final evil at the bottom of the magical labyrinth before the year resets.

• **Main Objective (Per Run):** Choose a magical-girl avatar in the hub and descend into the labyrinth to defeat that run’s boss and secure dungeon resources.

• **Secondary Objectives:**

– Build and deepen social bonds with NPCs to unlock new avatars and abilities.

– Gather dungeon resources to spend in the hub to unlock stuff.

– Balance time and resource allocation across social, exploration, and combat activities to prepare for deeper runs.

### 3.2. Progression

• **Dungeon Runs → Resources:** Every successful descent into the labyrinth yields dungeon resources.

• **Hub Spend:** Dungeon resources are spent in the hub to unlock things that help progress in the dungeon.

• **Soft Gating:** All progression uses resource-scaling rather than hard unlock gates. Choosing where and when to spend limited resources reinforces the core Risk-Reward/Cost Analysis theme.

• **Organic Difficulty Escalation:** Defeating bosses and hitting hidden milestones introduces tougher enemy variants, new floor layouts, and boss modifiers to the labyrinth without punishing less-skilled players.

• **Resolve System:** Before each run, players may enable curses (increasing dungeon difficulty) to boost resource rewards, offering an optional risk-for-reward lever.

• **Challenge Runs:** Fixed-scenario runs unlock over time, providing curated trials with preset conditions once earned.

• **Accessibility (Post-Launch):** Optional difficulty assists (e.g., “God Mode”) may be added after launch.

### **Core Progression Tracks**

1. **City Upgrades**
   * Players spend building materials to unlock or upgrade buildings and areas in the hub world.
   * Upgrades are soft-gated via rising resource costs, not locked behind mandatory boss kills.
   * City Reputation System – Complete hub and dungeon-related goals (e.g. “Reach Floor X Y times”) to unlock new content and perks, inspired by a Community Center-style system.
   * Seasonal Events – Unlocked by progression, offering unique storylines, activities, or limited-time resources.
   * Some ephemeral uses: spend building materials in-run to temporarily upgrade dungeon rooms (e.g. boost shop inventory).
2. **Base Upgrades**
   * Player Home Upgrades – Expand storage, unlock indoor features like crafting or cooking.
   * Base Expansion – Unlock new Base Facilities
3. **NPC Relationships**
   * New NPCs join the City/School as conditions are met; provide services or quests.
   * Players build relationships by giving **gifts**, which come in various categories (e.g. Sweet, Rustic, Fancy), with preferences differing per NPC.
   * Relationship level unlocks stronger mid-run boons (Hades-style). Boons are accessed through in-dungeon NPC rooms where players can spend gift resources.
   * Reaching Relationship levels with certain NPCs unlocks them as characters for future runs.
   * Relationship level is measured with “hearts” or similar, a Harvest Moon-style meter.
4. **Player Skill Tree**
   * Unlocked with **Player Upgrades** gained from dungeon runs.
   * Represents combat enhancements, passive bonuses, and core stat growth.
   * Ephemeral use: Can also be spent mid-run for one-time buffs or minor advantages.
5. **Pets**
   * Familiars – Companion creatures unlocked through the pet store that provide benefits in town and/or dungeon runs.
   * Pet Upgrades(Treats?) Can either be given to pets in hub world for permanent upgrades, or spent in dungeon for temporary upgrades.
6. **Dungeon Meta Progression**
   * Unlocks new dungeon content (floors, bosses, item pools) based on player achievements and milestones.
   * Progression here is not tied to currency, but to actions (e.g. “Reach Floor X 3 times,” “Defeat Boss Y”).
   * Avoids hard locks but introduces new variables organically over time.
7. **Other**
   * New Areas – Expand the City and surrounding map to access more content.

#### 3.2.1. Difficulty curve

Difficulty scales in two ways:

1. Organic Progression:

* Deeper dungeon floors naturally intensify (enemy density, complex mechanics).
* New layers/bosses unlock via both run achievements (e.g., defeat Boss X) AND hub milestones (e.g., upgrade Town Monument).
* Hints are diegetic: NPC dialogue, environmental changes, or Community Center tasks.

2. Player-Adjusted "Resolve" (Heat System):

* Before a run, players can enable curses (e.g., \*"Fragile” -20% HP\*).
* Each curse adds Resolve Points, scaling dungeon difficulty and resource/rare drop multipliers.
* Curses can be mixed/matched (e.g., add enemy armor + faster traps for higher rewards).

### 3.3. Play flow

Similar to the Core Loop section, here you’ll detail the expected flow of gameplay from the player’s perspective, not just the core loops of it. Mention if they are expected to do a couple of side missions before a main one, if they will gather collectibles to enhance their abilities at a specific point in the game, etc.

Each cycle represents one day in the one-year loop. Days consist of a Hub phase or a Labyrinth phase—chosen at dawn. Time in the Hub advances only when actions are taken.

**Hub Phase:**

• **Dawn Decision:** At the start of the day, the player may choose to remain in the hub or enter the labyrinth.

• **Hub Activities:** While in the hub, the player has a fixed set of daily action units. Each activity—conversation, gift giving, community task, or resource spending—consumes one unit. NPCs react to actions, and bond levels change NPC routines and dialogue. The day ends when action units are depleted or the player ends the day manually.

**Labyrinth Phase:**

• **Day-Long Commitment:** Entering the labyrinth at dawn consumes the entire day. Time inside does not affect the in-game date.

• **Dungeon Exploration:** Navigate rooms, defeat enemies, collect dungeon resources. No time constraints inside.

• **Exit Conditions:** The run ends—and the day concludes—when the player defeats the boss, uses an escape consumable, uses a recall room or loses all health

• **Bookmark Saves:** The player may create a bookmark save in any room that contains no active enemies. Loading a bookmark resumes the run.

**Decision Rhythm:**

Each dawn choice—hub or labyrinth—forces a Risk-Reward/Cost Analysis: invest in community and preparation or seek dungeon rewards at the cost of hub opportunities.

### 3.4. Difficulty

How will the difficulty of the game affect gameplay? How many different levels of difficulty will be implemented?

• **Organic Difficulty Scaling:** Achievement-based milestones unlock tougher enemy and room variants; no time-based gating ensures players control their own pacing.

• **Resolve System:** Before each run, players may volunteer for curses (e.g., -HP, faster traps) to increase dungeon difficulty and multiply resource rewards—emphasizing risk-reward decision-making.

• **Challenge Runs:** Fixed, curated scenarios unlock over progression, each with preset Resolve values, special win conditions, or limitations to test mastery.

• **Accessibility Options (Post-Launch):** Optional assists (e.g., “God Mode”) may be added after launch to broaden accessibility without altering core design.

• **Tone Separation:** The labyrinth remains deliberately intense and challenging, while the hub world retains a relaxed, low-stress atmosphere—reinforcing the dual mood of community-building versus high-stakes runs.

## 4. Mechanics

Most of the time, you will customize this section of the GDD to each of your games. For example, if your game has combat in it, you want to include a segment of “Combat” and one for “AI”, or if your game has a unique system for spawning, you’ll want to mention how it works.

### 4.1. Rules

The general rules of the game, what are the limits of the player’s actions.   
**4 Mechanics (Baseline Roguelike Prototype)**

#### **4.1 Roguelike Run Rules**

**4.1.1 Run Entry**

* Players may only enter the Labyrinth at dawn; beginning a run consumes the full in-game day.

**4.1.2 Run Initialization**

* Each avatar starts with unique base stats (health, mana/energy, movement speed, damage, etc.) and any designated starting items.
* Each Character recovers +5 mana on enemy kill by default
* Dungeon resources always begin at zero.

**4.1.3 Run Locking**

* Once inside, character choice and any pre-run curse/blessing selections are locked for the duration.

**4.1.4 Floor Progression**

* A fixed number of floors (initially five) must be cleared to face the final boss.

**4.1.5 Player Load-out & Item Handling**

* **Slots:** 1 Active, 1 Trinket, 1 Consumable.
* **Swapping:** Picking up a new Active/Trinket/Consumable replaces the equipped one and drops the old on the floor.
* **Dropping:** Trinkets and Consumables may be dropped without picking up a new one; Actives may not.
* **Use Rules:**
  + *Consumables* activate instantly (unless their own text says otherwise).
  + *Actives* fire only if their effect can fully resolve (e.g., “double ground resources” needs resources present).
  + Active items consume a fixed mana cost (defined per item).
  + Attempting to activate an item with insufficient mana fails and leaves the item on cooldown.

* **Inventory:** No backpack—equipped items = total inventory.
* **Pickup UX:** Equipables show name + small tagline and require a button-press to pick up; resources & hearts auto-collect on contact.
* **Floor-loot persistence:** Anything dropped or not picked up stays until you leave the floor.

**4.1.6 Actions & Cooldowns**

* Every avatar now four actions:

• Primary, Secondary and Utility – run on independent cooldown timers (and may also consume mana).

• Ultimate – has \*\*no cooldown\*\* but instead expends a fixed number of charges per room; charges reset when the player enters a new floor.

* Cooldown timers scale with any global time-scale modifiers (e.g., time-slow effects).
* Particularly strong Secondary/Utility actions may also consume a limited mana or other resource pool in addition to cooldown.

**4.1.7 Failure & Exits**

* **Abandon Run:** Quit mid-dungeon and return to hub; lose 80% of carried resources.
* **Escape Consumable:** Ends the run and retains **50%** of collected resources (baseline).
* **Recall Room:** Same as Escape Consumable—safe exit, 50% retention.
* **Defeat (HP = 0):** Ends run with no additional penalty; retain 20% of resources.
* **Defeat Final Boss:** Ends run; retain **100%** of collected resources.
* **Post-Boss Exit Portal (tentative):** Upon defeating a boss, a portal appears offering an optional safe exit back to town.
* No instant-death tile

**4.1.8 Health Recovery & Healing Mechanics**

* **Heart Pickups:** Scatter through loot tables.
* **Healing Rooms:**
  + Base 15% chance on Floor 1, –2 pp per subsequent floor down to a 5% floor minimum.
  + Guaranteed Floor 3 spawn if none have appeared in the first two floors

**4.1.9 Room Flow**

* **Combat Rooms:** Seal all exits until every enemy is defeated, then spawn a chest with an 80% chance.
* **Cleared Rooms:** Remain empty and fully back-trackable.
* **Floor Exits:** Strictly one-way.

**4.1.10 Damage, Defence & Status**

* **Health Model:** Discrete hearts (baseline 3 pips, base damage is half a pip).
* **I-frames:** 0.5 s on hit.
* **Status Effects:**
  + *Slow* –30% move/attack for ~1 s.
  + *Stun* ≤ 0.25 s (telegraphed).
* No universal dodge—defensive tools are character-specific.

**4.1.11 Enemy & Projectile Behaviour**

* Enemy projectiles never hurt other enemies unless specifically baked into the enemy behaviour.
* Soft-homing allowed only on elites/bosses; standard shots fly straight.
* Self-damage off by default (only abilities that explicitly state self-harm can hurt the caster).

**4.1.12 Environmental Hazards**

* Spikes, lava, traps deal defined damage per contact (TBD in balance pass).
* Hazards can re-hit once i-frames expire.

**4.1.13 Bosses & Routes**

* **Victory:** Defeat the current final boss
* **Alternate Routes:** Unlocked via achievements at branch points.
* **Boss Rewards:**
  + 1–2 full hearts.
  + One unique boss-pool item (suppressed if already owned).
  + Resource bundle scaled by Resolve multiplier.
* No automatic heart-container increases.

**4.1.14 Map Visibility & Discovery**

* **Mini-map:** Only shows visited rooms and their adjacent neighbors.
* **Fog-of-War:** Unvisited adjacent room types remain hidden on the mini-map until revealed by exploration or map-reveal items.

**4.1.15 Miscellaneous Run Features**

* **Save Points:** Bookmark-style saves in any room without active enemies; reload restores exact run state
* **Miniboss Rooms:** Only in dead-end branches or immediately before the final boss.
* **Difficulty Variants:** Enemy/trap variants scale by floor.
* **UI Indicators:** Health bar, mana bar, Ultimate-charge pips, minimap with floor name, skill cooldowns, dungeon-resource counter, bookmark icon (active vs. inactive).

##### **4.6 To-Add (Omitted for Prototype)**

*These systems are essential to the full game but are excluded from the baseline roguelike prototype to focus initial development on core dungeon functionality.*

* Social Bonding Mechanics (day-phase relationship systems)
* Hub Economy & Unlocks (spending resources to unlock dungeon aids)
* Resolve System (pre-run curses & blessings for risk-reward)
* Detailed NPC Boon Room Effects (boon definitions and impact)
* Player Interaction & UI Shell (menus, dialogue interfaces, hub HUD)
* Placeholder Assets (art, audio specifications)
* Exact Resource Drop Ranges (to be defined in 4.5)

### 4.2. Game universe

How the game universe works. Mention here the stuff that is done outside of the perception of the player, like restocking inventories of key NPCs.   
**4.2.1 Roguelike Background Systems**

| **Sub-system** | **Final specification** |
| --- | --- |
| **Floor Generation & Layout** | Procedural assembly from hand-authored room tiles.  • **Room count / floor:** Floor 1 ≈ 8 ± 2; +2 rooms per later floor to ≈ 20 ± 2.  • **Combat-room density:** ≈ 70 % on Floor 1, +3 pp per floor(capped at 85%)  • **Exits per room:** 2–4(Dead-ends have 1 exit), each exit shows a neutral room-type indicator; dead ends allowed.  • **Topology:** single linear critical path. 30–50 % of its rooms spawn 1–3-room branches. All special-purpose rooms (shop, treasure, secret, miniboss, boon, *lore room – placeholder*) appear only on those branches. |
| **Secret Rooms** | Base spawn chance 35 %/floor. After all enemies die, one floor tile pulses for 0.3 s; stepping on it within 10 s opens the entrance hidden in the wall, |
| **Treasure / Item Rooms** | Guaranteed each floor until the first final-boss floor; 20 %/floor thereafter.  • **Gate types** (equal weight, no repeat on consecutive floors): Key-Lock / Combat Gauntlet / Resource Offering / Trap Corridor / Wheel of Punishment / Debuff. |
| **Recall Rooms + Spawn State**  **Boss Room** | Spawn chance starts 10 % on Floor 1; +20 pp each floor without a recall room; hard-cap 90 %. Resets to 10 % whenever a recall room spawns (found or not).  Always generates on a dead end, contains the floor exit after boss is defeated. |
| **Enemy Progression & Room-Template Pools** | Six floor-tiers (1–3, 4–5, 6–7, 8–9, 10–11, 12). Each tier supplies Easy / Medium / Hard room templates (ratios TBD). No stat multipliers.  • **No-consecutive-reuse:** templates used on one floor are withheld from the next floor only. |
| **Miniboss Rooms + Tracker** | 50 % spawn chance/floor; if a miniboss spawned on the previous floor, next-floor chance = 25 %, then resets to 50 % after one floor with no miniboss spawn. One miniboss room per floor; four unique minibosses per tier; always in branch dead ends. |
| **NPC Boon Rooms + Pool** | Unlocks when any NPC hits 2 hearts.  • Spawn: 20 %/floor; guaranteed on Floor 3 if none spawned on 1–2; never on consecutive floors.  • In-room: offers **exactly three** random eligible NPCs (or all if <3) with ≥ 2 hearts; chosen NPC’s boon granted. |
| **Shop Upgrade Slots** | Each purchase adds +1 item slot (max 3 → 5 slots) for the current run. Upgrades can appear in every shop on every floor in the prototype. |
| **Treasure-Room Gate History** | Stores last-used gate type; that gate cannot appear on the next floor only (one-floor cooldown). |
| **Secret-Room Chance Modifier** | **Placeholder** — no dynamic modifiers in the prototype; fixed 35 % base chance. |
| **Resolve Resource Multiplier** | **Placeholder** — activates once the future Resolve system is implemented. |

### **4.2.1-A Treasure-Room Gate Definitions**

| **Gate** | **Prototype behaviour** |
| --- | --- |
| **Key-Lock** | Door opens when player spends **1 Key Crystal**. |
| **Combat Gauntlet** | Room seals; defeat all spawned enemies to unlock the treasure. |
| **Resource Offering** | Player sacrifices a fixed **percentage of random carried permanent resource** to open the treasure. |
| **Trap Corridor** | Treasure sits behind environmental hazards (spikes, darts, etc.). Reaching the chest without dying opens it — no resource cost. |
| **Wheel of Punishment** | Shows one low-tier item and a locked high-tier item. Player spin the wheel, pays the random cost (health, resources, or run-only debuff) the wheel lands on to unlock and take the higher tier item. When one item is taken, the other vanishes |
| **Debuff Gate** | Claiming the item immediately applies a specific run-only stat penalty. |

### **4.2.1-B Room-Type Definitions**

| **Room type** | **Prototype definition** |
| --- | --- |
| **Combat Room** | Exits seal on entry. The generator chooses one of three variants:  1. **Standard Fight** — defeat all enemies to reopen exits and chance to spawn a reward chest  2. **Empty Variant** — no enemies; exits reopen immediately (small chance per combat-room roll).  3. **Trap Variant** — no enemies; instead the room contains environmental hazards that must be navigated to unlock the exits and claim the chest (small chance per combat-room roll). |
| **Treasure / Item Room** | Contains one equippable item gated by one of the six Treasure-Room Gates; gate type cannot repeat on consecutive floors. |
| **Secret Room** | Hidden 35 %/floor; discovery mechanic as described above. |
| **Shop** | NPC vendor sells two base-stock items (upgradeable during the run) for Gold; no stealing mechanic. |
| **Miniboss Room** | Single elite enemy; branch dead end; reward scales to miniboss tier. |
| **NPC Boon Room** | Offers three randomly selected eligible NPCs for boon selection (see Boon-Room Pool rules). |
| **Recall Room** | Safe exit that ends the run early and lets the player retain 50 % of carried resources. |
| **Healing Room** | 15 % base spawn on Floor 1 (-2 pp each later floor to a 5 % minimum); guaranteed if none spawned in the last three floors; restores health. |
| **Lore Room** *(placeholder)* | Reserved for narrative progression; excluded from prototype. |
| **Boss Room/ Floor Exit** | Boss of the Floor resides here, upon defeat, spawns a one-way door to the next floor. |

### **4.2.2 Social-Sim Background Systems *(placeholder)***

To be completed once daytime loop is fully scoped (NPC schedules, shop restocks, calendar events, etc.).

### 4.3. Physics

The overall physics of the world. Is it realistic? Low gravity? Destroyable environment?

#### **4.3 Movement & Physics**

**4.3 Physics**

**4.3.1 Movement Feel** Player movement is omnidirectional on a 2D plane with instant-on / instant-off response: pressing a direction immediately sets velocity to full move\_speed; releasing stops movement instantly. No inertia or momentum is simulated.

**4.3.2 Collision Response** On colliding with solid geometry, the perpendicular component of velocity is zeroed while the parallel component is preserved, producing a smooth slide-along-wall effect. Diagonal input naturally slows to the axis speed when sliding.

**4.3.3 Environmental Forces** Prototype stub: no active environmental forces (e.g., conveyors, wind) are present in the prototype.

**4.3.4 Movement-Linked Feedback** Animation playback rate scales linearly with the avatar’s current move\_speed (clamped to approximately 0.8× – 1.2× baseline). No hit-stop or time-slow occurs on hits or collisions, ensuring clear, readable action.

**4.3.5 Hitboxes & Hurtboxes**

* **Player:** Circle, radius 8 px, centred in the 64 × 64 sprite frame; serves as both hitbox and hurtbox.
* **Enemies:** Axis-aligned bounding box derived from the trimmed sprite frame, inset by 2 px on each side.

**4.3.6 Collision Layers & Masks**

* **Layers:**
  + Player – body & hurtbox
  + Enemy – body & hurtbox
  + PlayerProjectile – bullets/beams from player
  + EnemyProjectile – enemy shots
  + Environment – walls/obstacles
  + Hazard – spikes, traps
  + Pickup – items, power-ups
* **Solid Collisions (Blocks):**
  + Player ↔ Environment, Enemy
  + Enemy ↔ Environment, Enemy
  + PlayerProjectile ↔ Environment, Enemy
  + EnemyProjectile ↔ Environment, Player
  + Environment blocks all projectiles
* **Overlap Triggers:**
  + Player overlaps EnemyProjectile, Hazard, Pickup
  + PlayerProjectile overlaps Enemy
  + Hazard overlaps Player (damage trigger)
  + Pickup overlaps Player (pickup trigger)

**4.3.7 Physics Update Rate & Timestep**

* Fixed-step rate: 60 Hz (physics tick = 1⁄60 s)
* Substeps: 1 per rendered frame (no catch-up)
* Interpolation: ON — render positions are interpolated between physics ticks

**4.3.8 Projectile Kinematics & Motion Types**

* **Base Movement:** Kinematic bodies moving in straight lines at constant velocity.
* **Motion Types:**
  + **Straight:** Fixed direction & speed
  + **Variable Speed:** Speed ramps over lifetime (start\_speed → end\_speed via curve)
  + **Continuous Homing:** Adjusts toward target each tick at turn\_rate within acquisition\_range
  + **Initial-Targeted:** Locks direction once at spawn toward target, then travels straight
* **Bouncing Framework (item-driven):**
  + **Default:** Projectiles destroy on impact
  + **When enabled:** Each projectile has restitution and optional max\_bounces; on collision, the impacted velocity axis is inverted and multiplied by restitution until lifetime or bounce limit expires

### **4.4 Economy**

#### **4.4.1 Currency Types**

* **Run-Only Currencies (reset at run end)**

**• Gold (cap 100)**

**• Keys (cap 100)**

**• Health Hearts (start 3, cap 10)**

**• \*\*Mana Pickups (cap = Max Mana; Small = +25 mana, Large = +50 mana)\*\***

* **Permanent Resources** (no cap)  
  + **City Upgrade Materials**
  + **Base Upgrade Materials**
  + **Player Upgrade Materials**
  + **Pet Upgrade Materials**
  + **NPC Gifts**

#### **4.4.2 Resource Rules**

* Run-only resources observe the caps listed above.
* Permanent resources have no carry limit.
* **Run End Outcomes**
  + **Safe Exit:** keep 100 % of permanent resources gained in the run.
  + **Death / Abandon:** keep 20 % of permanent resources; lose 80 %.
* **Gold → Permanent Resource Conversion:** TBD (not used in prototype).

#### **4.4.3 Acquisition**

* Clearing a combat room has an **80 %** chance to spawn a chest.
* Baseline enemies drop nothing unless modified by equippable items or special traits.
* Chests may contain **pickups**, **permanent resources**, and/or **equippable item**.
* For slot-rolling, **Gold is treated as a Pickup**.
* Each slot will be rolled to spawn it’s reward, if all prior slots fail, the last slot is always a guaranteed success
* **Pickup vs. Resource taxonomy**
* • *Pickups* = Health, Gold, Keys, **Bombs (if added)**, Mana/Energy, other one-off consumables.
* • *Resources* = all five permanent resource categories (City Upgrade Materials, Base Upgrade Materials, Player, Pet, NPC).

#### **4.4.4 Chest Types**

| **Chest** | **Slots** | **Slot Table** |
| --- | --- | --- |
| **Common** | 2 | 1 Pickup 2 Resource |
| **Uncommon** (gated) | 4 | 1 Trinket 2 Pickup 3 Resource 4 Pickup or Resource |
| **Rare** (gated) | 6 | 1 Equippable Item → later slots fail if earned 2 Trinket → slot 3 fails if earned 3 Consumable 4 Pickup 5 Resource 6 Resource or Pickup |
| **Glass** | 1 + (# rooms cleared before unlock) | Follows player; unlocks after floor boss; shatters on 2 hits. Slots resolve in order and always succeed:  1 Resource, 2 Pickup, 3 Pickup + Resource, 4 Pickup + Consumable, 5 Resource + Trinket, 6 Lower-Tier Equippable, 7 Higher-Tier Equippable |

#### **4.4.5 Chest Gates**

* **Locked:** 1 Key (Uncommon) / 2 Keys (Rare)
* **Resources:** pay % of a random permanent resource; Rare costs more
* **Gold:** pay flat Gold; Rare costs more
* **Trapped – Health:** take ½ Heart (UC) or 1 Heart (Rare)
* **Trapped – Mana:** pay 10 % (UC) or 20 % (Rare) of max Mana; if none, becomes Trapped-Health
* **Shell Game:** illusion chests; pick one; wrong pick spawns hazard (1 UC, 2 Rare)
* **Gauntlet – Wave:** defeat 1 (UC) or 2 (Rare) waves of enemies
* **Gauntlet – Timed:** survive 15 s (UC) or 30 s (Rare) of continuous spawns

#### **4.4.6 Primary Resource Sinks (In-Run)**

* **City Upgrade Materials:** purchase up to +3 shop item slots per shop each run (cost increases per purchase).
* **Base Upgrade Materials:** *TBD*.
* **Player Upgrade Materials:** grant a temporary stat buff to the player.
* **Pet Upgrade Materials:** grant a temporary buff to the familiar.
* **NPC Gifts:** upgrade the Boon Room’s available boon selection.
* A special **Resource Shop** may spawn, selling items priced in permanent resources.
* Certain Treasure-Room gates consume a percentage of resources to enter.

#### **4.4.7 Shop Scaling**

* Shop item prices remain constant throughout a run.
* Shop-slot upgrade cost (paid in City Upgrade Materials) rises each time it is purchased during that run.
* If the player holds an active **Resource Multiplier**, slot-upgrade costs are multiplied by the same factor.

### 4.5. Character movement

The range of movement that the player has within the game world.

*Shared Movement Model*

* + - *Every playable avatar inherits the movement model defined in §4.3 Physics: instant-response, omnidirectional traversal on a 2-D plane with slide-along-wall collision.*

*4.5.1 Basic Avatar — Dash*

* + - *• Distance: 2 × player collision-width*
    - *• Duration: 0.25 s*
    - *• Cooldown: 1.5 s between uses*
    - *• Invulnerability: player is untargetable for the dash’s full 0.25 s*
    - *• Direction: fires toward the current movement input; if none, uses the last non-zero input*

### 4.6. Player interaction

What can the player interact with?

### **4.6 Menus & UI Shell (Prototype)**

#### **4.6.1 Main Menu**

| **Element** | **Function** |
| --- | --- |
| **Start Run** | Loads directly into a new dungeon run using the current build. |
| **Options** | Opens the Options Menu (4.6.3). |
| **Quit Game** | Closes the application after confirmation (see 4.6.4). |

*Footnote:* A single‐line **Build Info / Credits** string (e.g., “Prototype v0.1.0 – © Studio Name”) is displayed at the lower-right corner.

#### **4.6.2 Pause Menu**

| **Element** | **Function** |
| --- | --- |
| **Resume** | Returns to gameplay. |
| **Restart Run** | Re-initialises the current seed from Floor 1 (confirmation required; see 4.6.4). |
| **Options** | Opens the Options Menu (4.6.3) without leaving the run. |
| **Quit to Desktop** | Exits the application after confirmation (see 4.6.4). |

#### **4.6.3 Options Menu (prototype scope)**

**Audio**

* Master Volume slider
* Music Volume slider
* SFX Volume slider

**Display**

* Fullscreen toggle
* Windowed resolution selector (restricted to OS-reported modes)
* V-Sync toggle

**Controls**

* Single-page key / button rebinding for **Move**, **Fire**, **Dash**
* Controller vibration toggle

*(No additional accessibility, graphics, or save-file settings are included at this stage.)*

#### **4.6.4 Confirmation Dialogs**

| **Trigger** | **Dialog Text** | **Choices** |
| --- | --- | --- |
| **Quit Game** (Main Menu) | “Quit game? Unsaved progress will be lost.” | **Yes** / **No** |
| **Quit to Desktop** (Pause) | “Quit game? Unsaved progress will be lost.” | **Yes** / **No** |
| **Restart Run** (Pause) | “Restart current run? All progress in this run will be lost.” | **Yes** / **No** |

*All menus are navigable by keyboard, mouse, and standard game-pad inputs. Menu ordering follows common UX conventions: primary action first, quit last; confirmation dialogs guard any irreversible action.*

#### 4.6.2. Saving

How will saving work with the game? Are there save points? Can the player save anywhere?   
  
**Suspend Saves** ● One bookmark at a time, only in a cleared room.  
 ● Loading deletes the file; you may create a new bookmark later.

#### 4.6.3. Game options

What options can the player change from the menus?

### 4.7. Assets

A list of the main assets that the game will use, split by type: “Player Model, Player Texture, Enemy Model, Terrain Material, Enemy Death Sound, etc.

| **Bucket** | **Prototype Placeholder Spec** | **Notes / Best-Practice Rationale** |
| --- | --- | --- |
| **Player Sprite Sheet(s)** | 64 × 64, 4-direction idle/walk | 1 sheet per playable avatar as more characters arrive |
| **Enemy Sprite Sheet(s)** | 64 × 64 baseline foes, shared atlas | High-frequency sprites stay batched together for performance |
| **Boss Sprite Sheet(s)** | One atlas per boss, up to 256 × 256 | Large/rare frames kept separate to prevent atlas bloat |
| **Tileset / Terrain** | 16 × 16 floor, wall, door, hazard tiles | Single tileset for vertical slice; hazards reference own sheet below |
| **Hazard Sprite Sheet** | Spikes, lava vents, dart traps | Shares “Hazard” collision layer; isolated for clarity in level editor |
| **Pickup Sprite Sheet** | Hearts, Gold, Keys, Mana, (Bomb TBD) | World-space art that drops from chests/rooms |
| **Equippable Item Icons + World Sprites** | 32 × 32 icons + dropped-item sprites for Actives, Trinkets, Consumables | Aligns with three equip slots defined in §4.1.5 |
| **Chest & Gate Art** | Chest variants (Common/Uncommon/Rare/Glass) + Gate types (Key-Lock, Resource, Trap, etc.) | Gates and chests use unique atlases so they can stream in only on eligible rooms |
| **VFX & Projectiles** | Simple geometric particles drawn in-engine | Colour-code later for damage types |
| **Minimap & Status-Effect Icons** | 16 × 16 glyph atlas | Covers explored room states and Resolve/buff pips |
| **UI & Core Icons** | Font + monochrome atlas (hearts, mana pips, key/coin HUD) | Re-use glyph sheet to reduce texture swaps |
| **SFX Library** | Hit, Dash, Pickup, Chest-open, Key-spend, Gate-unlock | One-shots ensure tactile feedback early |
| **Music Loops** | 1 hub theme, 1 dungeon theme (60–90 s, seamless) | Keeps scope tight while allowing mood tests |
| **Voice / VO** | *None for prototype* | — |
| **Build Info / Credits String** | “Prototype vX.Y.Z – © Studio Name” | Renders on Main Menu footer; negligible memory hit |

## 5. Graphics and audio

### 5.1. Visual system

An overall mention of how the visuals of the game will work, and if there’s a reason behind it. Is it 2D or 3D? Cell-shaded, minimalistic or realistic?

| **Aspect** | **Prototype decision** |
| --- | --- |
| Projection & style | 2-D, top-down pixel art (Godot) |
| Sprite & tile size | 64 × 64 px everywhere |
| Facing directions | 4 (up / down / left / right) |
| Rendering | Pixel-perfect integer scaling; game runs at 60 fps |
| Internal resolution | 1 024 × 576 px (16 × 9 tiles) |
| Lighting / post-FX | Flat pass + minimal additive VFX (no real-time lights) |
| Animation budget | Two-frame ping-pong loops at 12 fps (game logic at 60 fps) |
| Performance goal | Hold 60 fps on minimum-spec PC |
| Camera & UI details | Moved to 5 .1 .1 and 5 .2 respectively |

### **5 .1 .1 Player Camera**

* **Single-screen rooms** — instant snap; no smoothing.
* **Large rooms** — camera follows player in whole-pixel steps, clamped to room bounds; temporary 0 .75× zoom-out while inside.
* **Pixel-perfect lock** — camera origin rounded to integers every frame.

### **5 .1 .2 Environment**

* One utilitarian 64 × 64-tile map per room (walls, floor, props).
* Tiles static; no animation, decorations, or breakables yet.
* Walls use full-tile square colliders; no irregular shapes.

### **5 .1 .3 Character Sprites**

* 64 × 64 px, centre pivot (32, 32).
* Walk/idle: 2-frame loop @ 12 fps.
* Attack: 2 frames (“wind-up / wind-down”) @ 12 fps.
* Hurt & death handled with VFX (colour flash, etc.).

### **5 .1 .4 Projectiles & VFX**

* Bullet sprite: 16 × 16 px, centre pivot (8, 8), drawn **above** characters.
* Static image; sprite-based hit overlay on same frame.
* No rotation in prototype (all bullets symmetrical).

### **5 .1 .5 Colour Palette**

* **Hub** — soft, pale, muted pastels for a cozy vibe.
* **Dungeon** — dark indigo / charcoal / **dark purple** base with high-saturation neon accents (cyan, magenta, acid yellow, electric teal) for an uncanny, liminal feel.
* No hard palette-size limit; two area themes enforced via simple shader ramps.

### **5 .1 .6 UI / HUD**

* Moved to **Section 5 .2 Interface** for detailed treatment.

### **5 .1 .7 Screen-Shake & Camera FX — *stub***

* To be defined after prototype: integer-pixel shake amplitudes, accessibility toggle.

### **5 .1 .8 Shader & Post-Processing Rules — *stub***

* To be defined after prototype: palette-ramp shader IDs, additive glow material, post-FX order.

#### **To-Do 5 .1**

1. Final-pass outline / shading rules for sprites.
2. Decide exact icon dimensions once HUD art begins.
3. Flesh out stubs 5 .1 .7 and 5 .1 .8 during polish phase.

### 5.2. Interface

What will the user interface look like? How will the player interact with it? How will it affect gameplay?

The prototype’s interface balances constant situational awareness with a clean, low-clutter presentation at the native 1024 × 576 resolution.

#### **5.2 (HUD layout)**

* **Safe-zone** – A two-tile (128 px) wall band surrounds the play-field; no gameplay objects spawn inside it. All HUD elements sit within this margin.
* **Always-visible widgets**
  + **Top-left (vertical stack)** – Hearts → Gold → Keys
  + **Top-right** – Mini-map
  + **Bottom-centre (inside frame)** – Active-item label
  + **Bottom-centre (below frame)** – Skill bar: Trinket | S1 S2 S3 U | Consumable
* **Toggle-able widgets** – Passive-item strip (top-centre), mini skill-cooldown pips (under player sprite).
* **Icon size** – TBD (layout assumes “½-tile squares” until final dimensions are chosen).
* **Font** – TBD (bitmap or vector; programmer will select).
* **Navigation** – Hybrid mouse + keyboard/gamepad input across all HUD buttons.
* **Performance guard-rail** – HUD must never drop the global 60 fps target.

#### **5.2.1 Pause Menu**

| **Entry** | **Behaviour** | **Confirmation** |
| --- | --- | --- |
| **Resume** | Returns to play | — |
| **Restart Run** | Restarts current run | “Restart the run? Yes / No” |
| **Options** | Opens Options screen | — |
| **Quit to Desktop** | Exits application | “Quit to desktop? Yes / No” |

* Vertical list; hybrid navigation (↑↓ + Enter/A or mouse).
* Initial focus defaults to **Resume**.
* Game action and input remain paused while the menu or a confirmation dialog is open.

#### **5.2.2 Options Screen**

Left-side tab list: **Audio • Controls • Graphics • Debug**

**Audio**

* Master volume (slider 0–100 %)
* Music volume (slider)
* SFX volume (slider)
* Mute all (checkbox)

**Controls**

* Keyboard re-bind list (per-action)
* Gamepad re-bind list (active when pad detected)
* Controller vibration (On/Off)

**Graphics**

* Window mode (Windowed / Borderless / Fullscreen)
* V-Sync (On/Off)

**Debug** *(stub)*

* Empty container reserved for run-time variable toggles, FPS overlay, etc.
* Tab navigation: ←→ switches tabs, ↑↓ moves within a tab, Enter/A or mouse click activates controls.
* Apply / Back buttons at lower right; unsaved changes prompt “Save changes? Yes / No”.

### **To-Do 5.2**

* Choose final HUD icon dimensions and update layout accordingly.
* Select prototype font (bitmap or vector) and integrate into HUD.
* Populate **Debug** tab with variable-visibility toggles and FPS overlay.

### 5.3. Audio system

An overall mention of how the audio of the game will work, and if there’s a reason behind it. If your game has in-game voice chat, be sure to include it here.

### **5.3 Overview & Standards**

**Audio Design Pillars**

* Clarity of feedback – every sound should instantly communicate its purpose.
* Mood support – music and ambience reinforce the game’s atmosphere without distracting.
* Non-fatiguing loops – tracks and ambiences remain pleasant over long sessions.

**Pipeline & Workflow**

* Author → DAW export → import as **48 kHz, 16-bit stereo OGG** → routed through Godot buses (Music, SFX, UI).
* File-name convention: Category\_Event\_v##.ogg (e.g., Player\_PrimaryAttack\_v01.ogg).
* Version-control: audio lives under /assets/audio/ with meta-files committed alongside sources.

**Engine & Toolchain Specs**

* Godot default 48 kHz mix rate.
* One reverb send per bus; no global compression until post-prototype.

**Licensing & Sourcing Policy**

* Prototype assets must be **CC0 / royalty-free**.
* CC BY tracks are permitted only if credited on the splash screen.

**Mix & Loudness Targets**

* Integrated loudness – 23 LUFS ± 2.
* True peak < –1 dBTP.

**Accessibility Baseline**

* Master volume, Music, and SFX sliders plus Mute-All option required by Beta.

### **5.3.1 Game Music (Prototype Scope)**

| **Track** | **Purpose** | **Notes** |
| --- | --- | --- |
| **Title / Main-menu** | Startup & navigation | Seamless 60–120 s loop. |
| **Dungeon / Combat** | Regular gameplay | Medium-intensity 60–120 s loop. |
| **Boss-room** | Boss encounters | High-intensity 60–120 s loop. |

*Static stereo mixes only; dynamic layering deferred to a later milestone.*

### **5.3.2 Sound Effects (Prototype Scope)**

#### **Player-Action SFX**

* Primary Attack
* Secondary / Alt Attack
* Utility Skill
* Ultimate Skill
* Taking Damage
* Player Death
* Resource Pickup (coins, keys, etc.)
* **Item Pickup** – single cue for all active/trinket/passive items
* Consumable Use
* Active-Item Use

#### **Enemy-Action SFX**

* Attack
* Taking Damage
* Death

#### **UI SFX**

* Hover / Focus
* Confirm / Click
* Back / Cancel

#### **Environment SFX**

* Door Open / Close
* Chest Open
* Trap / Hazard Activate
* Switch / Lever Toggle
* Room-Clear Stinger *(optional)*
* Ambient Loop *(optional)*
* Footsteps (player & enemies)

All SFX imported as 48 kHz OGG; routed to SFX bus.

### **To-Do 5.3**

1. Implement dynamic-layering system for music after prototype stability.
2. Add volume sliders and master mute to the Options menu before Beta.
3. Decide whether to promote the optional **Room-Clear Stinger** and **Ambient Loop** to mandatory status during Alpha polish.

## 6. Story and narrative

### 6.1. Backstory

What events of interest happened before the start of the game?

### 6.2. Main plot

What’s the main plot of the game? Just write the most important stuff here in a condensed form, remember that this is a game design document, not a web novel.

#### 6.2.1. Plot progression

How will the plot progress throughout the game?

### 6.3. Cutscenes

Don’t mention specific cutscenes (Just do it if they are extremely relevant to the game), only mention how you will use cutscenes in gameplay.

## 7. Characters

### 7.1. Main characters

Who are the main characters in the game? If you have more than one, then add a small description of all subpoints from this segment for each one of them.

#### 7.1.1. Backstory

#### 7.1.2. Personality

#### 7.1.3. Appearance

#### 7.1.4. Abilities

#### 7.1.5. Relationships

### 7.2. Supporting characters

Who are the main enemies? You don’t have to include all the previous subpoints for this one, just a brief description of them.

### 7.3. Enemies

Who are the supporting characters? As with the supporting characters, you don’t have to include all main character’s subpoints, except if the enemy description plays a huge part in the overall plot of the game.

## 8. Game world

### 8.1. Look & Feel of the world

Similar to 2.8, but in this case, you’re just talking about the game world, not the game in general.

### 8.2. Locations

What are the most important locations in the game and how will they be relevant to the game?

#### 8.2.1. Connection to the plot

Add this one for every mentioned location to tell how they will connect to the plot.

### 8.3. Levels

Just as each one of their names say, briefly describe the levels of the game (If there’s any).

#### 8.3.1. Tutorial levels

#### 8.3.2. Main levels

#### 8.3.3. Optional levels

# Currently Undocumented Systems/Mechanics/Designs etc.

The following is a list of things that will be in the game, but have not been reflected in the GDD yet, for reference:

**Relationship pacing rules**

* +1 affinity per daily conversation
* Up to **2 gifts per NPC per week**
* Chores/quests grant *extra* points outside those caps
* Flat daily cap throttles heart gains

**Chore-day structure**

* Overlapping daytime chores (e.g., “help the general store *or* the blacksmith on Wednesday”)
* Each chore consumes the rest of the in-town day but still allows a night-time dungeon run

**Event/quest visibility options**

* Toggleable HUD icons above NPCs / minimap markers for active mini-events
* Default “off” for full immersion; players can turn cues on in Settings

**Calendar reminder icon**

* Small HUD symbol that appears *one day in advance* for festivals, birthdays, etc.

**Weighted-RNG recall rooms**

* Chance increases each floor; *guaranteed* spawn after an undecided “safety floor” threshold
* Recall rooms contain a **single-use bed** (sleep in-dungeon) and a one-way exit portal

**Crop upkeep & automation**

* Crops wilt if unwatered while the player is on multi-day dives
* Late-game unlock: sprinklers / auto-watering to mitigate the risk

**Blessing & curse pre-run system**

* **Curses**: increase difficulty + resource multiplier
* **Blessings**: spend shared town currency/materials to add perks (e.g., dungeon time-slow)
* Currency choice still TBD, but it will *compete* with cozy-side spending

.

# To-Do

This section contains things that need to be revisited later.

**To-Do 4.2**

* 4.2-A Flesh out Lore Room content and integration.
* 4.2-B Finalise Easy / Medium / Hard template ratios and exact “empty” & “trap” Combat-Room probabilities.
* 4.2-C Design complete Resolve system and implement Resource Multiplier.
* 4.2-D Decide whether shops should stop spawning after a given floor in full balance pass.
* 4.2-E Create dynamic Secret-Room Chance modifiers if desired.
* 4.2-F Complete 4.2.2 Social-Sim Background Systems.

### **To-Do 4.4**

1. Define the **carry cap** for Mana/Energy Pickups.
2. Specify the **in-run use** for **Base Upgrade Materials**.
3. Set a **Gold → permanent resource conversion ratio** for the full game.
4. Define placeholder sinks (special in-run functions) for Health, and additional uses for Gold and Keys.

# For Later

This section contains things that need to be decided upon in future sections

Explore design variants for **Bombs** as a future Pickup type.