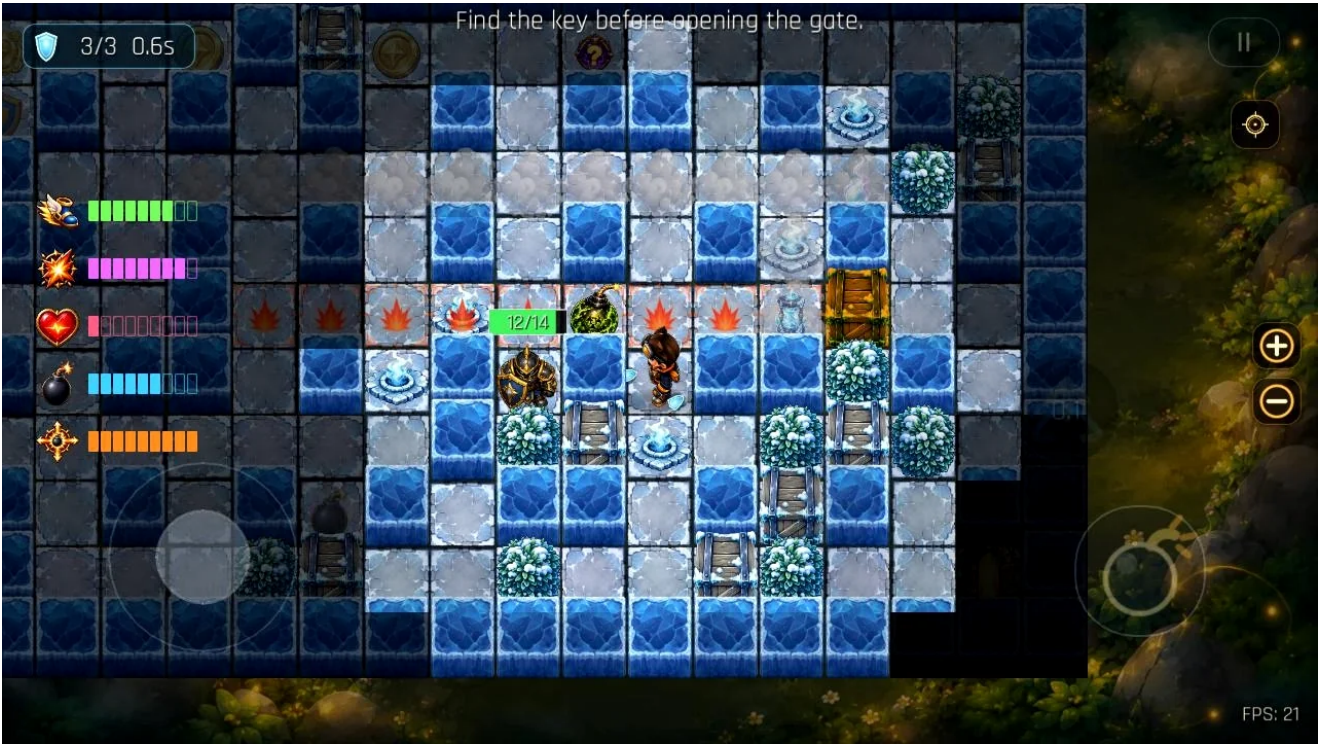


BlastFuse: Bomb Arena - Game Design Case Study

Dang Tuan Anh (DTA Craft) - 2026-07-24

This is the most recent project, and the main design sample being submitted here.



Real gameplay screenshot, Frost world: current HUD layout with segmented stat bars (speed, blast, HP, bomb, damage), a shield timer, and the frost tile set

1. Project Overview

Genre	Offline single-player bomb-placement action game with campaign progression, boss fights, and light meta-progression
Platform	Android (shipped, signed builds), Windows Desktop (export target present); no iOS build yet
Engine	Godot 4.7, Mobile feature profile, GL Compatibility renderer, Jolt Physics
Role	Solo designer/developer - game design, gameplay/systems implementation, UI/UX, level design, balancing, testing, and build/release preparation, all handled directly
Project status	Most recent project, currently in active development; 3 incrementing signed Android App Bundles shipped (v47–v49), the flagship title on the studio site
One-sentence pitch	An offline mobile arena game where players place bombs to clear enemies and obstacles across a 60-level campaign, fight data-driven bosses, and grow a small equipment/ability loadout between runs

2. Target Player and Product Positioning

- Target audience: Casual mobile players who want short, self-contained sessions rather than live-service or PvP experiences - the game is built around "fast, readable rounds" in an offline-friendly package.
 - Player motivation: The loop rewards short-term mastery (clearing a level cleanly, beating a boss) and short-term collection (Gold, Bomb Parts, Energy Shards, equipment upgrades), which suits players motivated by steady, visible progress rather than competition or randomness. This is written into an explicit design guardrail in docs/Economy.md: the game's IAP is Gold-only, and it's deliberately kept away from anything that feels random, loot-box-y, gacha, or gambling-adjacent - a positioning choice against manipulative monetization patterns.
 - Comparable games: The core "place a bomb, wait for the fuse, clear a blast radius" mechanic is a genre descendant of classic bomb-arena games (Bomberman-style titles), with a structured campaign, boss-fight layer, and equipment/ability meta-progression layered on top.
 - Product differentiation: A few things that set this apart from a generic arena clone:
 - A data-driven boss-phase system (DataDrivenBossPhaseController.gd) supports ~22 distinct boss profiles (12 campaign bosses + 10 challenge-pack bosses) from shared phase/attack/telegraph data instead of one-off bespoke code per boss - this scales boss variety without scaling production cost the same way.
 - A replayable Boss Rush mode (data/modes/boss_rush.json) and a standalone Boss Fight Pack (data/tables/boss_fight_challenges.json, 10 challenges) let players revisit boss content outside the linear campaign.
 - A guardrailed economy (docs/Economy.md) that keeps core progression separate from IAP and from time-limited seasonal rewards, so paying or logging in daily can't outpace normal play.
-

3. Core Gameplay Loop

The loop, in one line:

Select Level → Play the Board (place bombs, dodge, chain blasts) → Clear Objective / Defeat Boss → Collect Rewards (Gold, Bomb Parts, Energy Shards, EXP) → Spend on Abilities & Equipment → Unlock Next Level, repeat

Step-by-step: 1. **Select Level** - Players pick a level from the World Select / Level Map screens. The campaign is split into three 20-level worlds (Earth 001–020, Frost 101–120, Space 201–220). 2. **Play the Board** - The core moment-to-moment gameplay: grid movement, bomb placement, fuse timing, blast-radius resolution, chain reactions, and enemy AI reacting to bomb danger. 3. **Clear Objective or Defeat Boss** - Each level has a defined objective (destroy a block count, detonate a bomb count) or a boss encounter with defined phases. 4. **Collect Rewards / Progress Meta-Loop** - Gold, Bomb Parts, and Energy Shards drop from enemies/bosses (8% and 90% drop chances respectively) and feed into ability upgrades, equipment Forge upgrades, and cosmetics. 5. The loop repeats with the next level unlocked, and periodically a **Return loop** (daily rewards/missions, seasonal milestones) and a **Boss Rush / Boss Fight Pack** side-loop give players a reason to revisit earlier content.

Session structure: The level-based structure (a discrete objective per level, tutorial-first onboarding, results screen after each attempt) is built around short, single-level sessions of a few minutes each, to fit mobile casual play patterns.



Real HUD crop from current gameplay: shield timer, then speed, blast, HP, bomb, and ability meters, each with an icon and a live bar - the same icons reused in the Forge and results screens

4. Core Mechanics

4.1 Bomb Placement & Blast Resolution

Aspect	Detail
Purpose	The central verb of the game - create hazards that clear obstacles/enemies but can also threaten the player.
Player Input	Place a bomb on the current grid tile via touch; some ability unlocks add kick and throw variants.
Game Rule	The bomb detonates after a fuse timer, resolving a blast pattern that can chain into other bombs.
Player Feedback	Dedicated VFX cues, a freeze-frame "hit stop" effect on impact, and a "Bomb Frenzy" effect for chained detonations.

Aspect	Detail
Failure / Edge Cases	Players can be caught in their own or an enemy's blast; dedicated revive/pause popups and a mobile failure-state test handle and verify this path.

4.2 Grid Movement

Aspect	Detail
Purpose	Precise, readable positioning is the skill expression of the game - since bombs are placed on tiles, movement precision determines success.
Player Input	Directional/touch input, with lane-assist/recovery tolerance tuned for mobile play.
Game Rule	Cardinal grid-based movement rather than free-form movement, keeping bomb-tile alignment predictable.
Player Feedback	Regression tests cover movement direction and grid data correctness, since this mechanic is foundational.
Failure / Edge Cases	A touch-scroll guard prevents accidental input during scroll gestures on mobile - a concrete mobile-usability problem addressed directly.

4.3 Enemy AI (data-driven behavior primitives)

Aspect	Detail
Purpose	Deliver 15 visually distinct enemy types without needing 15 bespoke AI scripts, keeping content production scalable.
Player Input	Indirect - players react to enemy behavior rather than controlling it directly.
Game Rule	Each enemy is defined by a combination of 17 shared behavior primitives (<code>walk_chase</code> , <code>avoid_bomb_danger</code> , <code>bomb_seek_pounce</code> , <code>shield_front_block</code> , <code>phase_move</code> , <code>ambush_reveal</code> , and more), plus 3 personality profiles (cautious, aggressive, hunter).
Player Feedback	Enemy health bars, and a danger map the AI uses that also informs telegraphing.
Failure / Edge Cases	Escape-urgency handling makes enemies react differently when nearly cornered or at low HP - a fairness/readability refinement made after watching how the AI played in practice.



Ash Stalker



Bomb Slime



Bush Lurker



Clockwork Bomber



Crystal Beetle



Ember Sprite



Flying Hex Mask



Forest Trickster



Frost Totem



Fuse Bandit



Iron Brute



Mole Digger



Phase Wisp



Shield Knight



Void Caster

The 15 regular enemy types, each reading differently at a glance so players can tell threats apart mid-run

4.4 Boss Fights (data-driven phase system)

Aspect	Detail
Purpose	The game's biggest challenge/skill checks and its most memorable content beats.
Player Input	Same movement/bomb-placement verbs, but under boss-specific phase rules (attack windows, vulnerability windows).
Game Rule	<code>DrivenBossPhaseController.gd</code> reads phase data to sequence phases, telegraphs, summons, and teleports per boss; ~12 campaign bosses plus 10 non-campaign Boss Fight Pack bosses share this system, with one bespoke controller for a specific boss.
Player Feedback	Boss-specific visual scale (2.5x render scale), dedicated HP bars, and natural/forced telegraph states before attacks.
Failure / Edge Cases	A content validator enforces a strict 1:1 mapping between boss entries and phase data, so a boss can't ship without valid phase data.

4.5 Meta-Progression: Abilities, Equipment, and the Forge

Aspect	Detail
Purpose	Gives players a reason to keep playing beyond a single level - a light build-crafting layer between runs.
Player Input	Spend Gold/EXP on 15 abilities across 3 branches (demolition, mobility, awareness), and spend Gold + Bomb Parts + Energy Shards to upgrade 16 equipment items through Forge tiers T1→T2→T3.
Game Rule	Bomb Parts and Energy Shards are earned primarily from enemy/boss kills (8%/90% drop chance respectively) rather than purchased directly, keeping the Forge loop tied to play rather than payment.
Player Feedback	Dedicated Skills and Inventory screens with widgets visualizing the ability tree and equipment stat effects.
Failure / Edge Cases	Tests verify Forge resources and equipment state survive save/load and can't be corrupted or tampered with - one test specifically asserts that a tampered Energy Shard (<code>energy_shard</code>) quantity in a save file gets rejected.



Gold



HP



Bomb Part



Energy Shard



Magnet



Shield



Speed



Damage

A sample of real drop items: Gold and the two Forge currencies (Bomb Part, Energy Shard) alongside run pickups (Magnet, Shield, Speed, Damage)

4.6 Bomb Variants

Aspect	Detail
Purpose	The bomb itself comes in different elements - each one changes what the blast does, not just how it looks, adding build variety on top of the base placement/timing skill.
Player Input	Same placement input as the default bomb; the equipped variant determines the blast's behavior.

Aspect	Detail
Game Rule	12 elemental bomb types: Default, Fire, Frost, Thunder, Toxic, Nature, Water, Holy, Shadow, Void, Magma, and Clockwork.
Player Feedback	Each variant has its own icon and blast VFX, reused consistently between the HUD, Forge, and results screen.



Default



Fire



Frost



Thunder



Toxic



Nature



Water



Holy



Shadow



Void



Magma

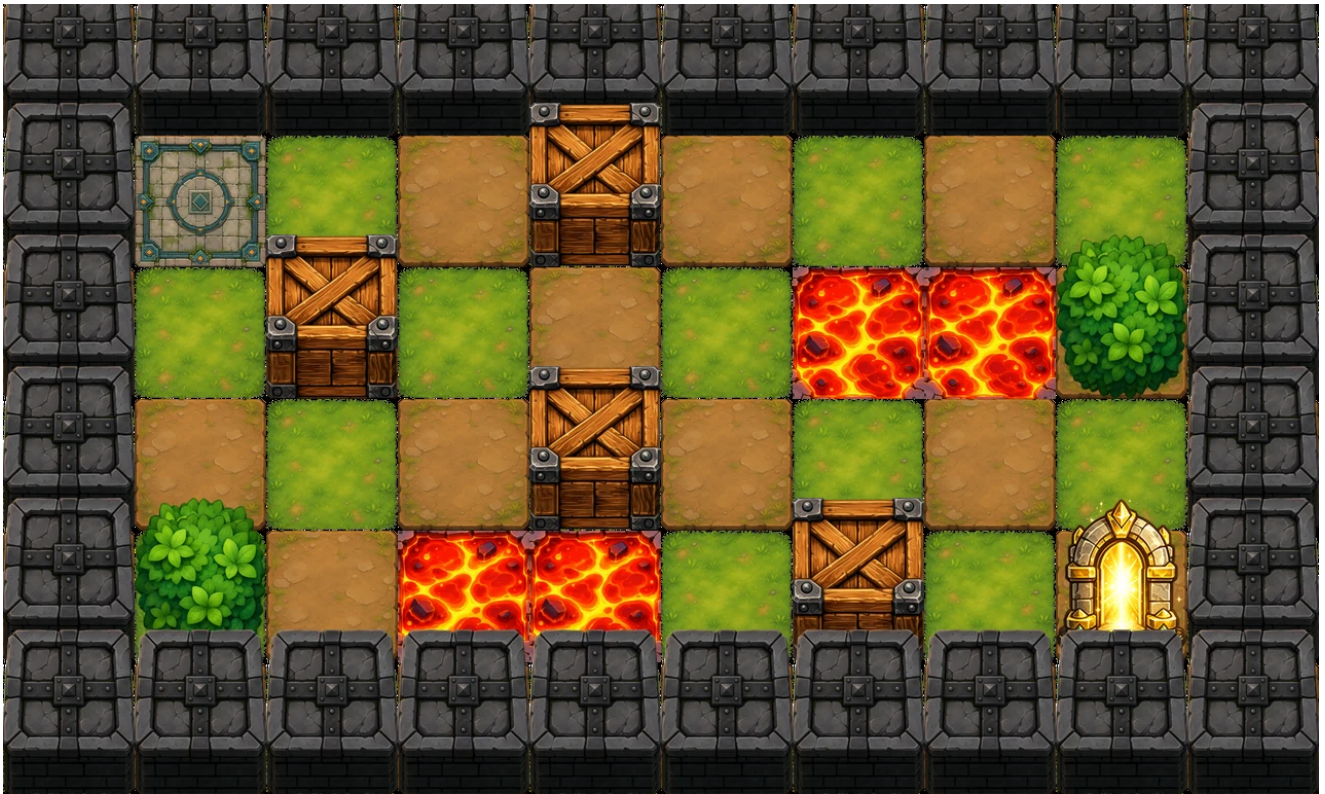


Clockwork

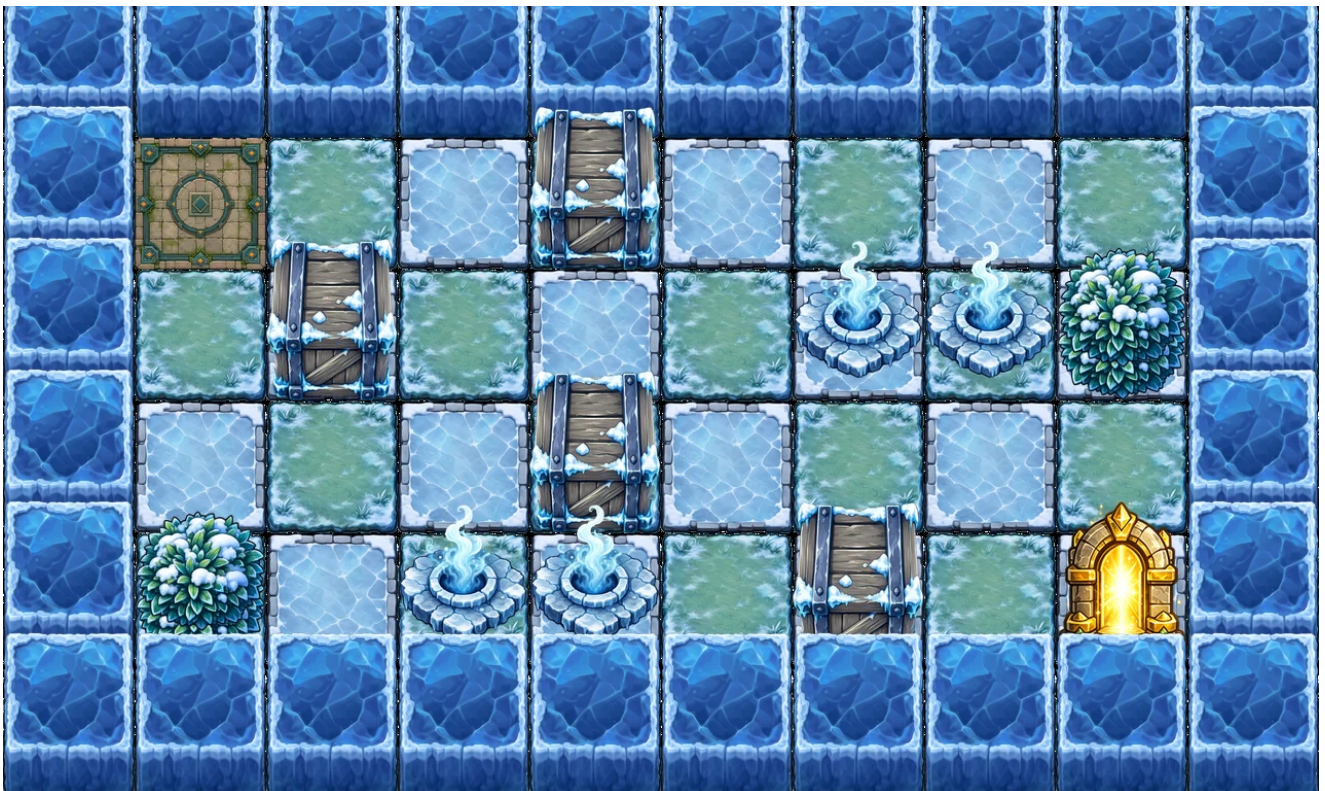
The 12 bomb elements - each changes blast behavior, not just its color

5. Level Design and Difficulty

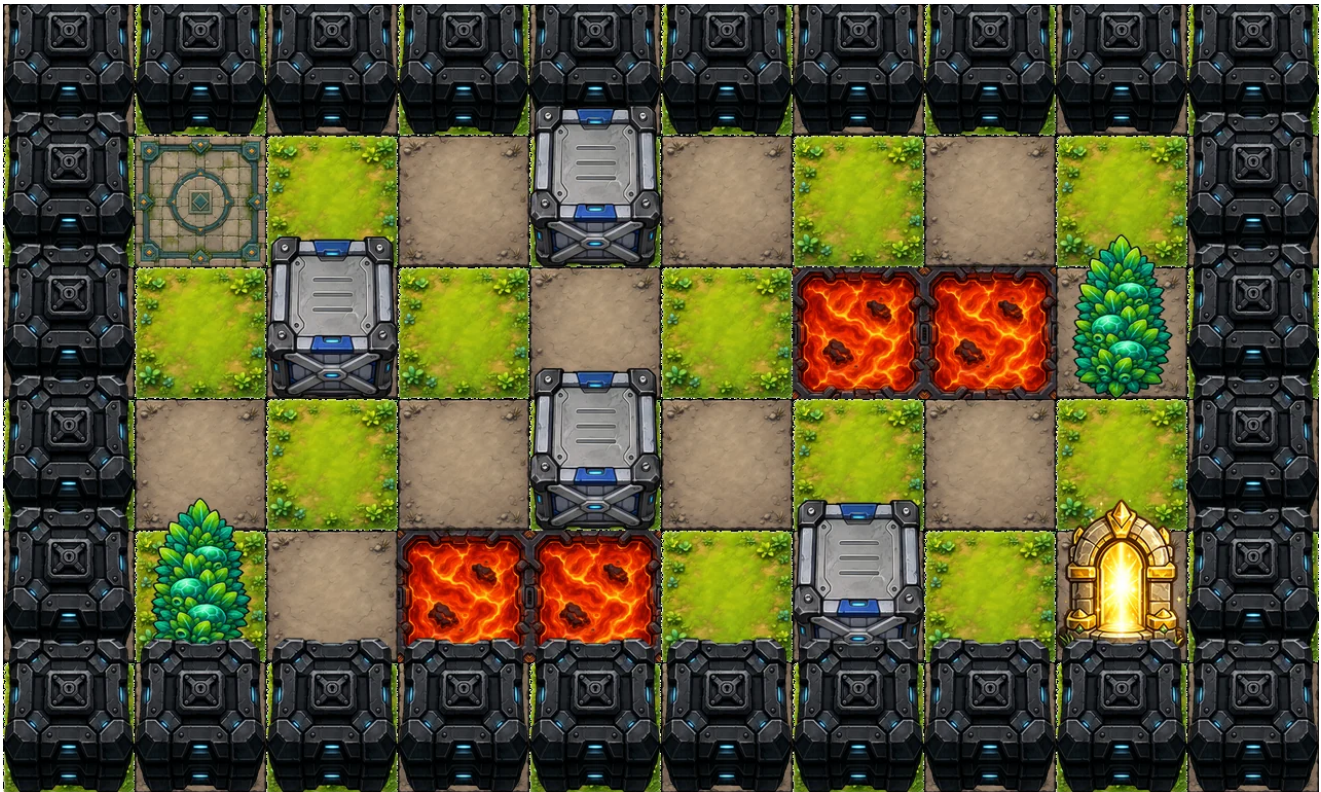
- Level flow: 60 levels across 3 worlds - Earth (001–020), Frost (101–120), Space (201–220) - each world gated behind the previous.
- Three world tile themes: the same grid layout and blocking logic is reskinned across all three campaign worlds - only the tile set changes, not the puzzle logic underneath it.



Earth world tile set: grass, wooden crates, lava hazards, and the lit exit gate



Frost world tile set: the identical layout reskinned with ice floor, frozen crates, and blue flame hazards



Space world tile set: the identical layout reskinned with metal floor, sci-fi crates, and alien plants

- Onboarding: A fully interactive tutorial board rather than static text screens, so players learn by doing on a real board.
- Mechanic introduction: Mechanics and enemy types are introduced gradually - early levels use reduced enemy HP and simpler objectives, ramping up as players progress.
- Difficulty progression: Objective target counts were explicitly tuned upward during the Frost-world expansion (raising destroy_blocks/detonate_bombs targets to 10/16/8) so the objective would be a real content phase rather than something clearable in passing. Early levels use HP overrides as a deliberate soft start.
- Escalation: Boss difficulty escalates through per-world capstone bosses (Earth: Phase Wisp, Void Caster, Clockwork Bomber, Molten Warden; Frost: Frost Sentinel, Glacier Warden, Frost Wisp, Frost Marauder; Space: Void Relay, Astral Foundry, Nova Wisp, Ash Marauder), plus a separate, harder Boss Fight Pack (10 bosses, "Crate King" through "Blast Core Titan") for players who want extra challenge outside the campaign.
- Retry structure: Pause/revive popups and a dedicated mobile failure-state test make retrying a designed flow, not just a hard game-over screen.
- Completion conditions: Level completion is gated on explicit objective data, not just an implicit "no enemies left" check.
- Regular enemy roster (15 types):

Enemy	Role	Movement	HP
Ash Stalker	Chaser	Ground	3
Bomb Slime	Bomber	Ground	1
Bush Lurker	Ambusher	Ground	3
Clockwork Bomber	Bomber	Ground	4
Crystal Beetle	Charger	Ground	5
Ember Sprite	Controller	Flying	4
Flying Hex Mask	Phaser	Flying	4

Enemy	Role	Movement	HP
Forest Trickster	Ambusher	Ground	3
Frost Totem	Controller	Ground	5
Fuse Bandit	Bomber	Ground	4
Iron Brute	Defender	Ground	5
Mole Digger	Digger	Ground	3
Phase Wisp	Phaser	Ground	4
Shield Knight	Defender	Ground	5
Void Caster	Caster	Ground	5

6. UI/UX and Player Experience

- Main screens: Boot Splash → Language Picker (first launch only) → Main Menu → World Select → Level Map → Gameplay → Results, plus Mode Hub, Village Hub, Boss Fight Select, Skills, Inventory, Store, Statistics, Save Slots, Settings, Credits, and a Debug Level Select.
- HUD information: A dedicated HUD panel with widgets for cooldowns, stats, and touch controls, rather than ad-hoc overlays.
- Input method: Touch-first, with input context switching so menu navigation and board control don't interfere with each other.
- Tutorial/FTUE: An interactive tutorial board rather than passive text/slideshow - the first lesson is the real mechanic, not a description of it.
- Readability decisions: A minimum font size is enforced across all UI specifically for mobile readability, checked by a dedicated test.
- Mobile usability decisions: A touch-scroll guard (tested directly) stops accidental taps/scrolls from triggering unintended UI actions - a real usability bug found and fixed during development.
- Localization: 6 shipped locales (English, Spanish, Portuguese-Brazil, French, German, Indonesian). Vietnamese was supported earlier on and later removed as a scope decision.

7. Balance and Progression

- Key variables: Movement/lane/diagonal-turn thresholds, fog-of-war radius, VFX timing, mobile low-VFX performance overrides, paid ability-reset Gold cost, enemy Bomb Part drop chance (8%), boss Bomb Part drop chance (90%).
- Reward structure:

Reward source	Approx. value
Daily missions	~47.4 Gold/day total
Daily login rewards (days 1–6)	315 Gold total
Seasonal milestone ladders	250–300 Gold
Annual one-time seasonal rewards	100–250 Gold

- Player stat ranges (starting value / floor / base cap):

Stat	Starting Value	Min	Max (base cap)
Speed	2.0	1.0	3.0

Stat	Starting Value	Min	Max (base cap)
Damage	1	1	3
HP	3	0	6
Bomb Capacity	1	1	3
Blast Length	1	1	3

- Upgrade/progression logic: Player stats start capped low and expand via Power Up pickups and progression spend, up to 3 upgrade levels per stat (Gold costs rise per level, e.g. 25/50/75 Gold for most stats) - a deliberate soft-cap-then-expand curve. Equipment stat bonuses (Runner's Charm on Speed, Blast Lens on Blast Length, Demolition Glove on Damage, and others) stack additively on top of these base caps rather than being clamped away, so a fully-gearred loadout can push a stat beyond its base max - there's no single fixed "potential" ceiling beyond what equipment and ability combinations can reach.
- Difficulty tuning: Per-level HP overrides and rising objective targets - a live-ops-style layer, separate from core balance.
- Seasonal and special events: Two layers of time-limited content run alongside the core game - four recurring seasons and three calendar-specific special events, each with its own drop item, drop-chance, and collection-reward ladder. Both layers are guardrailed the same way as the rest of the economy: they add variety and short-term goals, but they're never allowed to outpace core progression.

Event	Window	Drop Item	Drop Chance	Collect this many for a Gold reward	Other Effects While Active
Spring	Mar-May	Seasonal Nectar	4%	10 → 25 Gold 25 → 75 Gold 50 → 150 Gold	Blast Length pickups +50%, HP pickups +20%, wider fog-of-war (5 tiles), more Bush Lurkers/Forest Tricksters
Summer	Jun-Aug	Seasonal Ember	4%	10 → 25 Gold 25 → 75 Gold 50 → 150 Gold	Gold drops +40%, Speed pickups +30%, more Ember Sprites/Bomb Slimes
Autumn	Sep-Nov	Seasonal Leaf	4%	10 → 25 Gold 25 → 75 Gold 50 → 150 Gold	Damage pickups +40%, Bomb Capacity pickups +25%, more Bush Lurkers/Forest Tricksters
Winter	Dec-Feb	Seasonal Crystal	4%	10 → 25 Gold 25 → 75 Gold 50 → 150 Gold	HP pickups +50%, Bomb Capacity pickups +20%, more Frost Totems
Tet	Jan 28-Feb 20	Tet Coin	5%	5 → 40 Gold 15 → 90 Gold 30 → 170 Gold	Gold drops x2, Bomb Capacity pickups +50%
Halloween	Oct 25-Nov 1	Wisp Soul	5%	5 → 40 Gold 15 → 90 Gold 30 → 170 Gold	Damage pickups x2, Gold drops +50%, more Phase Wisps (x2) and Void Casters (+50%)

Event	Window	Drop Item	Drop Chance	Collect this many for a Gold reward	Other Effects While Active
Christmas	Dec 24-26	(no collectible - pure drop boost)	-	-	HP pickups x2, Gold drops +50%

Reading the milestone column: e.g. for Spring, collecting 10 Seasonal Nectar grants 25 Gold, 25 grants 75 Gold, and 50 grants 150 Gold - each milestone is a separate one-time reward, not cumulative on top of the last.



Nectar (Spring)



Ember (Summer)



Leaf (Autumn)



Crystal (Winter)



Tet Coin (Tet)



Wisp Soul (Halloween)

The 6 real seasonal/event drop items: Nectar (Spring), Ember (Summer), Leaf (Autumn), Crystal (Winter), Tet Coin (Tet), and Wisp Soul (Halloween)

- Monetization model: IAP is Gold-only, sold in three fixed packs (Small: 250 Gold/\$0.99, Medium: 700 Gold/\$1.99, Large: 1600 Gold/\$3.99) - deliberately not random/loot-box/gacha. A guardrail keeps progression from ever being balanced around IAP, and event/daily rewards from ever exceeding core progression's power level.
- Balance tables: data/tables/progression.json, enemies.json, objectives.json, equipment_items.json, abilities.json, drop_tables.json, data/config/balance.json, and docs/Economy.md.

8. Testing and Iteration

The project has **24 dedicated test scripts** - on a solo project, automated tests act as a second pair of eyes. A few examples worth calling out:

- Save integrity tests - verify legacy unsigned saves migrate and get re-signed, tampered saves get rejected (one test tampers with an Energy Shard (energy_shard) quantity and asserts the load fails), and encrypted

backup export/import round-trips correctly. Save-data integrity is treated as a first-class design requirement, not an afterthought.

- Boss fight tests - load all 10 Boss Fight Pack challenges into a real board and assert boss spawn, HP bar creation, correct visual scale, and correct phase/telegraph state; a UI test asserts the select screen renders exactly 10 rows with correct reward metadata.
- Meta-progression tests - verify ability effects (quick_fuse, phase_slip, hazard_ward, etc.) and Forge resource drop/spend paths function correctly.
- Mobile-specific tests - usability and failure-state tests for issues anticipated and guarded against for a touch-based audience.
- Localization tests - verify the 6-locale string system.

Design changes made based on what iteration revealed:

- Objective target counts were raised during the Frost-world expansion because early targets were clearable in passing - a direct response to a difficulty problem found in play.
- Vietnamese localization was fully removed after being supported earlier - a scope-reduction call.
- Boss Rush mode shipped first, then was made replayable in a later pass once it was clear players wanted to revisit it.
- "Escape urgency" AI logic was added afterward to improve existing enemy behavior - a refinement made after watching the AI in action, not a day-one feature.

This project isn't treated as "build once, ship once." Balance values, AI behavior, and even entire feature scopes (like localization language count) get revisited and changed after shipping, backed by an automated regression suite that protects save integrity, gameplay mechanics, and UI.

9. Responsibilities

The entire game was made solo - no other contributors. The work covered:

- Game design: core loop design, boss-phase system design, ability/equipment/economy design, difficulty curve tuning across a 60-level campaign.
- Implementation: all gameplay systems, all UI, all autoloader/core systems - the entire codebase.
- UI/UX: full screen flow, HUD, tutorial, mobile-specific input handling.
- Level design: 60 campaign levels plus tutorial, debug, and mode-specific level data.
- Testing: all 24 test scripts covering gameplay, save integrity, UI, and localization.
- Balancing: all balance tables and the economy design document, iterated over multiple passes.
- Art direction: a mix of original and licensed/third-party art assets, directed to fit together visually.
- Production management: planning and sequencing feature work - building data-driven systems (like the boss-phase system) before the content that depends on them.
- Release preparation: export configuration, Android package/signing setup, version-code increments, and 3 signed .aab builds shipped.