

High-Level Timeline

Total Duration: ~24–28 weeks (about 6–7 months)

- **Phase 0 – Setup & Alignment** (Week 1)
- **Phase 1 – Pre-Production** (Weeks 2–6)
- **Phase 2 – Core Systems Production** (Weeks 7–16)
- **Phase 3 – Content, Polish & Balancing** (Weeks 17–24)
- **Phase 4 – Packaging & External Readiness** (Weeks 25–28)

Each milestone below has:

- **Objective** – what it proves
 - **Owners** – who drives it
 - **Key Tasks** – what's actually happening
 - **Exit Criteria** – how you know you're done
-

◆ **Phase 0 — Project Setup & Alignment (Week 1)**

M0.1 – Project Kickoff & Foundations

Objective: Get everyone aligned on vision, tools, and communication so production can move without friction.

Owners: Producer (you), Designer, Senior Dev, Artist

Key Tasks:

- Confirm **demo scope** and constraints (10–15 min experience, one biome, one enemy type, one turret).

- Choose tools:
 - Unity 2D URP LTS version
 - Source control (GitHub, Plastic, etc.)
 - Task tracking (Jira/Trello/Notion)
 - Communication (Slack/Discord)
- Set up:
 - Unity project (URP, 2D template, input system)
 - Repo & branches (main / dev / feature branches)
- Create initial **project folder structure**:
 - Art/Characters, Art/Environment, Scripts/Systems,
Scripts/Gameplay, Scenes, Prefabs, Audio, UI
- Define **definition of done** for:
 - Features
 - Art assets
 - Bugs (P0–P3)

Exit Criteria:

- Unity project created and compiling.
- Repo initialized and everyone can pull/build.
- Task board created with initial Epics & Phase 1 tasks.
- One-page **demo scope statement** reviewed and agreed.

◆ Phase 1 — Pre-Production (Weeks 2–6)

M1.1 – Core Vision & Loop Lock (Weeks 2–3)

Objective: Lock the core loop and pillars so there's no “what is this game” ambiguity mid-development.

Owners: Designer (lead), Producer, Senior Dev, Artist

Key Tasks:

- Define **core gameplay loop**:

Fight zombies → Gather resources → Upgrade/repair base → Survive stronger waves

- Define **game pillars** (e.g. “tense but fair,” “resource-driven defense,” “cozy but dangerous nights”).
- Write **Micro-GDD (10–15 pages)**:
 - Player combat (melee + ranged)
 - Zombie behavior
 - Resource system
 - Base & turret system
 - Wave structure
 - UI & UX flows
- List **must-have vs. stretch** features for the demo.
- Create **flowcharts** for:
 - Wave progression
 - Resource → crafting → turret → defense loop

Exit Criteria:

- Micro-GDD completed and reviewed.

- One-page **loop diagram** everyone understands.
 - Clear “in-scope / out-of-scope” list for the demo.
-

M1.2 – Technical Blueprint & Whitebox (Weeks 3–4)

Objective: Nail architecture and a rough playable space before you add any real content.

Owners: Senior Dev (lead), Designer, Producer

Key Tasks:

- Create **Technical Design Document (TDD)**:
 - Systems: combat, AI, resources, crafting, base, save/load, UI.
 - Key interfaces (IDamageable, IResourceNode, etc.).
 - Event/messaging approach (UnityEvents, C# events, scriptable event system).
- Set up **core scenes**:
 - **Boot / MainMenu** (placeholder)
 - **DemoLevel** (whiteboxed layout)
- Whitebox:
 - Rough map (player spawn, base area, zombie spawn zones, resource clusters).
- Create **core prefabs (empty/placeholder)**:
 - Player
 - Zombie
 - Resource node
 - Base
 - Turret spot

- Decide on **resolution & camera** rules (pixel-perfect vs not, orthographic size).

Exit Criteria:

- TDD written and reviewed.
- You can:
 - Press Play
 - Move a placeholder player
 - Walk around a whiteboxed map (no real systems yet).

M1.3 – Art Style Test & Direction (Weeks 4–6)

Objective: Choose the art style and framework without blocking engineering.

Owners: Artist (lead), Producer, Designer

Key Tasks:

- Produce **test assets** in:
 - Pixel Art
 - Hand-painted
- For each style:
 - Player idle + walk
 - One zombie
 - Tree, rock, metal scrap pile
- Integrate into the whitebox scene for **readability tests**:
 - Is the player readable?

- Are zombies visually distinct?
- Does the environment cause noise?
- Define **final art style direction**:
 - Fully commit to one style **or**
 - Use one style for the demo and keep dual-style as a full-game stretch.
- Create **Art Style Guide**:
 - Palette basics
 - Line thickness (if relevant)
 - Scale (pixels per meter)
 - Animation FPS and frame counts

Exit Criteria:

- One art style chosen for the demo.
 - Art Style Guide documented.
 - Test scene with chosen style running in engine.
-

◆ **Phase 2 — Core Systems Production (Weeks 7–16)**

This is where the game becomes “real,” even if ugly at first.

M2.1 – Player Controller & Combat Core (Weeks 7–9)

Objective: Make the player feel good to control and fight with before anything else.

Owners: Senior Dev (lead), Designer, Artist

Key Tasks:

- Implement:
 - Movement (4/8 direction)
 - Collision handling
 - Dash/roll (if included)
- Implement **melee combat**:
 - Attack arcs
 - Damage application
 - Simple hit-stop
- Implement **ranged combat**:
 - Bullet prefab
 - Fire rate, recoil timing (if any)
 - Ammo pool & reload rules
- Hook in **temporary VFX & SFX** for feel.
- Add basic **health & damage** system.

Exit Criteria:

- Player can:
 - Move, aim, and attack.
 - Kill placeholder zombies.
 - Internal playtest: team agrees “combat has potential” (even with temp art).
-

M2.2 – Zombie AI & Wave System (Weeks 9–11)

Objective: Establish enemy behavior and wave pressure that defines the game’s tension.

Owners: Senior Dev (lead), Designer

Key Tasks:

- Zombie state machine:
 - Idle/Wander
 - Detect player/base
 - Chase
 - Attack
- Implement **pathfinding** (NavMesh/2D or custom).
- Create **Spawner system**:
 - Configurable wave definitions (count, rate, spawn points).
 - Difficulty curve parameters (speed, HP, count).
- Add simple **telegraphing**:
 - Spawn warnings or audio cues.
- Integrate **basic balancing**:
 - Initial TTK, movement speeds, attack damage.

Exit Criteria:

- Player can fight waves of zombies until overwhelmed.
 - A full 5–10 minute survival session is possible with placeholder balance.
-

M2.3 – Resource Gathering & Inventory (Weeks 11–13)

Objective: Let players meaningfully gather materials and track them.

Owners: Senior Dev, Designer, Artist

Key Tasks:

- Implement **resource nodes**:
 - Wood (trees), stone (rocks), metal (scrap).
 - Node HP & respawn cooldown (if any).
- Implement **resource pickup & UI updates**.
- Build **inventory system**:
 - Simple numeric tracking for each material.
 - Event-driven updates to HUD.
- Add **feedback**:
 - Hit VFX on nodes.
 - Pickup SFX.
- Ensure resources tie into crafting (even if crafting is stubbed at this point).

Exit Criteria:

- Player can:
 - Walk to a node.
 - Harvest it.
 - See resources increase in UI.
 - Data persists correctly through a session.
-

M2.4 – Base, Repair & Turret Crafting (Weeks 13–16)

Objective: The central fantasy: defend, repair, and build turrets using collected resources.

Owners: Senior Dev (lead), Designer, Artist

Key Tasks:

- Implement **base entity**:
 - Base HP
 - Zombie targeting rules
 - Repair interaction (hold button or press)
- Implement **crafting UI**:
 - Turret crafting recipe (wood/stone/metal).
 - Simple craft confirmation.
- Implement **turret placement**:
 - Valid/invalid spots.
 - Turret prefab placed.
- Implement **turret AI**:
 - Targeting nearest zombie.
 - Firing rate, damage.
- Implement **shared ammo system**:
 - Single ammo pool used by player & turrets.
 - UI reflects ammo usage by both.
- Initial **balance pass**:
 - Cost of turret
 - Impact of turret (should feel powerful but ammo-hungry).

Exit Criteria:

- Full loop is playable:

Fight → Gather → Build turret → Defend → Repair

- Internal test: you can survive multiple waves by engaging with all systems.
-

◆ Phase 3 — Content, Polish & Balancing (Weeks 17–24)

M3.1 – Environment & Art Integration (Weeks 17–19)

Objective: Replace placeholder assets with final (or near-final) art for the demo.

Owners: Artist (lead), Senior Dev

Key Tasks:

- Build final **tilemap** and environment layout:
 - Farm area
 - Base location
 - Resource spawn areas
 - Paths zombies navigate
- Integrate final **character animations**:
 - Player full set
 - Zombie full set
- Replace placeholder props with final:
 - Trees, rocks, scrap
 - Base structure
 - Turrets

- Ensure consistent **lighting/contrast**:
 - Player & enemies always readable.

Exit Criteria:

- Entire demo playable with final art (or 95% final).
 - No obvious placeholder sprites in core loop.
-

M3.2 – UI/UX Polish & Onboarding (Weeks 19–21)

Objective: Make the game understandable quickly without hand-holding.

Owners: Designer (lead), Artist, Senior Dev

Key Tasks:

- Finalize **HUD layout**:
 - Health
 - Ammo
 - Resources
 - Wave indicator
- Implement **tutorial/onboarding**:
 - Tooltips
 - Simple prompts (“Press E to harvest”, “Spend materials to build turrets”).
- Clean up **menus**:
 - Pause menu
 - Basic settings (volume, resolution).

- Ensure **input hints** match actual bindings.

Exit Criteria:

- New player can understand:
 - How to move, attack, harvest, craft, and defend
 - Without external explanation.
 - Internal “fresh eyes” test passes (one person unfamiliar can play with minimal confusion).
-

M3.3 – Balancing, Feel & Performance (Weeks 21–24)

Objective: Tighten combat, difficulty, and performance to investor-ready quality.

Owners: Designer (lead), Senior Dev, Audio, Artist

Key Tasks:

- **Balancing:**
 - Adjust zombie speeds, HP, damage.
 - Tune wave durations, pacing, and breaks.
 - Tune turret power and ammo consumption.
- **Game feel:**
 - Screen shake tuning.
 - Hit stop timing.
 - Sound layering for impact.
- **Performance:**
 - Add pooling to bullets/zombies if not already.

- Optimize tilemaps and sprite atlases.
- Test on low-spec PC.

Exit Criteria:

- Target session (10–15 minutes) feels tense, not frustrating.
 - Stable FPS on target hardware.
 - No game-breaking bugs or obvious stutters.
-

◆ **Phase 4 — Packaging & External Readiness (Weeks 25–28)**

M4.1 – Internal Release Candidate (Weeks 25–26)

Objective: Lock a build candidate stable enough for external eyes.

Owners: Producer, Senior Dev, Designer

Key Tasks:

- Create **RC build** (Release Candidate).
- Full **internal QA pass**:
 - Critical path tests.
 - Save/load tests.
 - Wave progression tests.
- Compile **Known Issues list**:
 - Clearly mark which are acceptable for demo.

Exit Criteria:

- RC build with:
 - No P0 blockers.
 - Only “acceptable” minor issues.
 - Known Issues doc created.
-

M4.2 – External Demo Package (Weeks 26–28)

Objective: Turn the build into a professional pitch package.

Owners: Producer (lead), Designer, Artist, Audio

Key Tasks:

- Build **Windows installer/zip** for demo.
- Set up **Itch.io private page** with:
 - Screenshots
 - Short description
 - Controls section
- (Optional) Set up **Steam draft page** (hidden).
- Record **60–90 sec gameplay trailer**:
 - Intro (farm night vibe)
 - Combat with zombies
 - Gathering & crafting
 - Base defense with turret firing
- Prepare **supporting docs**:
 - One-page demo overview (for investors/publishers).

- “What’s next after the demo” roadmap.
- Estimated budget/timeline for full game (you already started this).

Exit Criteria:

- Playable external demo build uploaded.
 - Trailer complete.
 - Demo overview PDF + roadmap ready for deck inclusion.
-



Milestones Summary (Cheat Sheet)

1. **M0.1 – Setup & Alignment** (Week 1)
2. **M1.1 – Vision & Loop Lock** (Weeks 2–3)
3. **M1.2 – Tech Blueprint & Whitebox** (Weeks 3–4)
4. **M1.3 – Art Style Direction** (Weeks 4–6)
5. **M2.1 – Player & Combat Core** (Weeks 7–9)
6. **M2.2 – Zombie AI & Waves** (Weeks 9–11)
7. **M2.3 – Resources & Inventory** (Weeks 11–13)
8. **M2.4 – Base, Repair, Turrets & Ammo Sharing** (Weeks 13–16)
9. **M3.1 – Environment & Art Integration** (Weeks 17–19)
10. **M3.2 – UI/UX & Onboarding** (Weeks 19–21)
11. **M3.3 – Balancing, Feel & Performance** (Weeks 21–24)
12. **M4.1 – Internal Release Candidate** (Weeks 25–26)

13. M4.2 – External Demo Package (Weeks 26–28)