LAST LIGHT – DEMO DEVELOPMENT DOCUMENT

Version 1.0
Prepared by Kato.8 Studios

---

1. Overview

1.1 Demo Purpose

This demo serves as a polished vertical slice showcasing the core experience of **Last Light**, a 2D survival shooter where players gather materials, defend a home base, and construct turrets to withstand escalating zombie waves. It is designed to demonstrate the game's identity, core mechanics, and production potential to investors, publishers, and collaborators.

1.2 Demo Experience Target

A **10–15 minute** gameplay loop featuring:

- Active zombie combat (melee + ranged)
- Resource gathering (wood, stone, metal)
- Base repair and turret construction
- Shared ammo economy between player and turrets
- One biome, one core enemy type
- Persistent base progression (save/load)

This demo is not the full game — it is a **tight representation of the gameplay fantasy and systems**.

1.3 Platform & Engine

- **Unity (2D URP)**
- **PC, Keyboard + Mouse**
- Pixel Art and Hand-Painted styles both supported for A/B testing

---

2. Demo Scope

2.1 Included Features

- Player movement + melee & ranged attacks
- Zombie pathfinding, attack behavior, wave escalation
- Harvestable material nodes
- Inventory UI
- Crafting UI (simple + intuitive)
- Base structure with repair function
- Craftable turret with basic targeting
- Shared ammo pool (player ↔ turret)
- Save/load persistence
- Small map with farm-like layout

2.2 Excluded (For Full Game Only)

- Multiple biomes

- Weapon variety
- NPCs or trading
- Advanced weather/lighting systems
- Full meta-progression
- Farming/crops
- Story elements
- Multiplayer or co-op

---

3. Development Phases

PHASE 1 — PRE-PRODUCTION (4–6 Weeks)

3.1 Vision & Requirements

Deliverables:

- Vertical Slice Goals
- High-Level GDD (10–15 pages)
- Feature Prioritization & Cuttable Scope
- Art Direction Test (Pixel vs Hand-Painted)
- Technical Blueprint (architecture, systems, coding standards)
- Basic whitebox map

Key Decisions:

- Lock player movement and weapon feel early
- Ensure art style test does not delay engineering
- Establish a prefab-driven, modular system for rapid iteration

---

3.2 Design Definition

Deliverables:

- Combat design spec (melee timing, ranged cadence, hitboxes)
- Zombie AI state machine
- Wave system (difficulty curve, spawn timing)
- Resource system (node health, drop table)
- Crafting requirements table
- Base upgrade flow
- UI wireframes (inventory, crafting, upgrade panels)

3.3 Technical Planning

Deliverables:

- Project folder structure
- Core prefab templates for:
    - Player
    - Zombie

    - Resource nodes
    - Turrets
    - UI Panels
- Save/Load architecture
- Input system configuration
- Performance constraints (pooling, atlas targets)

3.4 Art Style Tests

Deliverables:

- Player test sprite in both styles
- Zombie equivalent test
- Tree + stone + metal scrap samples
- Animation timing documentation
- Final selection: Pixel, Hand-Painted, or Dual-Support

PHASE 2 — PRODUCTION (12–16 Weeks)

---

4. Engineering Development

4.1 Player Systems

- Movement (8-direction or 4-direction depending on final art)
- Hitboxes & hurtboxes
- Machete attack sequence
- Gun firing system
- Ammo counter logic
- Damage interface API for all damageable objects

4.2 Combat & AI Systems

- Zombie state machine (Idle → Chase → Attack)
- Pathfinding
- Spawn manager + configurable waves
- Object pooling for AI + bullets
- Hit reactions, death behavior

4.3 Resource & Inventory Systems

- Resource spawning (trees, boulders, scrap piles)
- Gathering logic
- Inventory update events
- Drop table tuning
- UI sync between gameplay and HUD

4.4 Base & Crafting Systems

- Base structure with health
- Repair interaction
- Crafting menu logic
- Build placement validation
- Turret targeting + firing behavior
- Shared ammo system (player/turret)

4.5 Save & Load

Data saved:

- Player inventory
- Player health (if used)
- Base health
- Turret construction
- Current wave

Save system:

- JSON serialization
- Snapshot-based approach
- Load-on-start

4.6 UI Architecture

Elements:

- Inventory HUD
- Ammo display
- Health indicators
- Wave alerts
- Material counters
- Crafting/build menus

UI must remain:

- Readable
- Intuitive
- Non-intrusive

---

5. Art Production

5.1 Environment Assets

- Ground tiles
- Trees
- Rocks
- Metal scrap piles
- Farm perimeter props

- Base structure
- Parallax background layers

5.2 Character Assets

Player:

- Idle
- Walk
- Machete attack
- Shooting attack
- Hit reaction
- Death (optional)

Zombie:

- Idle
- Shamble
- Attack
- Hit reaction
- Death

5.3 Base & Crafting Assets

- Turret (body, head rotation frames)

- Repair tools
- Crafting icons
- Upgrade indicators

5.4 UI & VFX

- Resource icons
- Ammo icon
- Crafting menu frame
- Button hover/click states
- Hit FX
- Muzzle flash
- Zombie death FX

---

6. Audio Production

6.1 SFX List

- Footsteps
- Melee swings + impacts
- Gunshots
- Zombie growls, attacks, deaths
- Resources breaking

- Turret firing
- Crafting interactions
- UI input sounds

6.2 Music

- Ambient nighttime loop
- Wave escalation layer

6.3 Mixing

- Prioritize clarity during combat
- Duck ambient layers during high action

---

7. Game Design Tuning

7.1 Systems Tuned

- Player DPS
- Zombie HP & speed
- Turret fire rate
- Resource drop amounts
- Wave pacing
- Ammo scarcity

7.2 Iteration Framework

- Daily tuning tweaks during production
- Weekly internal playtests
- Bi-weekly balance reports
- KPI targets:
    - Time-to-kill (TTK) 1.2–1.7 seconds
    - Wave length 30–60 seconds
    - Average resource gain per minute

PHASE 3 — POLISH (3–4 Weeks)

8.1 Optimization

- Sprite atlas merging
- Pooling validation
- Reduce overdraw
- Remove unused assets

8.2 UX Improvements

- Clearer tutorial hints
- Better crafting readability

- Streamlined interaction prompts

8.3 Visual Enhancements

- Hit flashes
- Improved lighting/contrast
- Environmental polish
- Parallax refinement
- Screen shake tuning

8.4 QA Testing

- Bug fixing
- Edge case handling for save/load
- Wave consistency testing
- UI scaling tests

PHASE 4 — DEMO PACKAGE (2 Weeks)

9.1 Final Deliverables

- Windows playable demo (.exe)
- Itch.io private demo page

- Steam draft (hidden)

9.2 Press & Investor Material

- 60–90 second gameplay trailer
- Vertical slice feature summary
- Team overview
- High-level development roadmap

9.3 Post-Demo Adjustments

- Incorporate external playtest feedback
- Adjust onboarding clarity
- Final bug fixes

---

10. Schedule Overview

Phase	Duration
Pre-Production	4–6 weeks
Production	12–16 weeks
Polish	3–4 weeks
Packaging	2 weeks
Total	**~22–28 weeks**

11. Risks & Mitigation

Risk A — Combat may feel unsatisfying

Mitigation:

- Start combat prototyping immediately
- Add screen shake + hit stop early
- Frequent tuning playtests

Risk B — Art style delays

Mitigation:

- Only one style fully produced for demo
- Shared prefab structure ensures easy swap

Risk C — Scope creep in crafting system

Mitigation:

- Only one turret type in demo
- Only one base upgrade tier

Risk D — Save/Load instability

Mitigation:

- Build system early
- Test snapshots daily

Risk E — Enemy variety feels limited

Mitigation:

- Use pacing, density, and modifiers to create depth

---

12. Demo Deliverable Summary

Pre-Production Outputs

- GDD
- TDD
- Whitebox map
- Art style tests

Production Outputs

- Fully functional vertical slice
- Placeholder art replaced with final assets

Polish Outputs

- Performance pass
- QA pass
- Visual FX pass

Investor Outputs

- Final demo

- Trailer
- Pitch deck
- Financial roadmap
- Team plan