Pre-Production Milestones (Weeks 1 - 6)

		Production	GD	Eng/Dev	Art	Audio	QA
Milestone 0 — Project Setup & Alignment							
Sprint 0.1	Week 1	Establish production workflow	Outline core loop and high-level mechanics.	Initialize Unity URP 2D project.	Gather reference boards for possible art directions.	Establish preliminary tone & references for ambience/combat.	Set up early bug-reporting framework (Notion/Jira templates).
		Finalize demo scope, constraints, and expectations.	Start Micro-GDD skeleton.	Configure project settings (resolution, camera, input system).	Begin visual theme exploration.		
		Create project folder structure & documentation space.		Connect repo + set branching strategy.			
		Define Definition of Done (for code, art, QA).					
		Align team on 10–15 minute demo objectives.					
Milestone 1 — Vision Lock & Core Design							
Sprint 1.1	Weeks 2–3	Lock demo scope with the design team.	Complete Micro-GDD:	Review design documents for feasibility.	Create preliminary silhouettes and style explorations.	Define SFX/Music needs list.	Start documenting early test cases for the future core loop.
		Prioritize features (must-have, nice-to-have, stretch).	Combat rules (melee + ranged)	Begin planning technical architecture.	Evaluate readability needs based on design.		
		Approve loop diagram and design pillars.	Resource system				
			Zombie AI behavior				
			Wave structure				
			Crafting + turrets				
			Base repair system				
			UI/UX flow				
			Create gameplay flowcharts (combat loop, crafting loop, survival loop).				
			Create economy tables (materials, crafting costs).				
			Define KPIs (TTK, wave duration, resource yield).				
Milestone 2 — Technical Blueprint & Whitebox							
Sprint 1.2	Weeks 3–4	Approve TDD for production readiness.	Adjust map layout for pacing.	Complete Technical Design Document (TDD):	Create placeholder sprites for blockout readability.		Test navigation, movement space, collision sanity.
			Validate core loops in whitebox.	Combat system architecture.	Collaborate on level readability.		
				AI state machine structure.			
				Crafting and inventory backend.			
				Shared ammo system.			
				Save/load architecture.			
				UI—code messaging/events.			
				Build initial whitebox level:			
				Player spawn			
				Base location			
				Resource clusters			
				Zombie spawners			
				Implement placeholder prefabs.			
Milestone 3 — Art Direction Lock							
Sprint 1.3	Weeks 4–6	Facilitate art direction discussions.	Evaluate gameplay readability with both styles.	Integrate tests into whitebox.	Produce style test sets (Pixel + Hand-Painted):		Report readability issues, contrast problems.
		Lock the chosen visual direction for demo.	Choose which style best supports the experience.	Validate performance and readability.	Player (idle + walk)		
					Zombie		
					Resource node (tree, rock, scrap)		
					Test animation readability and silhouette clarity.		
					Draft and finalize Art Style Guide:		
					Palette		
					Line rules		
					Animation FPS		
					Tile size rules		
					Effects style (hit flashes, muzzle flashes)		

Production Milestones (Weeks 7–24)	Production	GD	Eng/Dev	Art	Audio	QA
Milestone 4 — Core Systems Implementation (Weeks 7–16)						
Sprint 2.1 - Player Controller & Combat						
Weeks 7–9		Define TTK values.	Build movement controller (8-directional).	Provide placeholder animation frames for early testing.	Add placeholder SFX (swings, hits, firing, steps).	Test movement edges, hit detection, damage logic.
		Tune early combat feel.	Implement melee attack (hitboxes, timing, damage).			
		Validate melee vs ranged balance.	Implement ranged system (bullets, ammo, fire rate).			
			Add early hit-stop & stagger logic.			
			Add damageable interface & health system.			
Sprint 2.2 - Zombie AI & Wave System						
Weeks 9–11		Set early difficulty curves.	Build AI State Machine:	Create placeholder zombie animations.	Add zombie growls, attack sounds.	Test pathfinding, wave pacing, enemy behaviors.
		Tune zombie speed, HP, damage.	Idle/wander			
			Detect → chase			
			Attack			
			Implement pathfinding.			
			Build wave spawner with difficulty scaling.			
Sprint 2.3 - Resource + Inventory Systems						
Weeks 11–13		Set resource yields per node.	Implement resource nodes (tree/rock/scrap).	Add early node sprites.	Gather/harvest sounds.	Test node respawns, gathering, and inventory updates.
		Ensure pacing supports turret building.	Build harvesting interaction.			
			Build inventory backend with HUD binding.			
Sprint 2.4 - Base, Crafting, Turrets, Shared Ammo, Save/Load						
Weeks 13–16		Tune turret DPS, fire rate, ammo usage.	Implement base HP + repair interaction.	Provide early turret/base concepts.	Turret firing sounds.	Test full loop:
		Define crafting costs.	Implement crafting UI + logic.		Base repair sounds.	Gather → craft → place turret → defend → repair.
			Build turret placement grid.			
			Build turret targeting/shooting system.			
			Implement shared ammo between player & turrets.			
			Implement Save/Load for:			
			Inventory			
			Built turrets			
			Base HP			
Milestone 5 — Final Art Integration (Weeks 17–19)						
Sprint 3.1						
Weeks 17–19		Adjust layout for final art readability.	Implement animation controllers.	Final player animation set.		Test map edges, collision, visual bugs.
			Integrate new art into gameplay prefabs.	Final zombie animation set.		
				Final environment tileset.		
				Final resource nodes.		
				Base + turret final art.		
				Parallax backgrounds.		
				Polish VFX (hit flashes, muzzle flashes, death fx).		
Milestone 6 — UI/UX + Onboarding (Weeks 19–21)						
Sprint 3.2						
Weeks 19–21		Create onboarding flow:	Bind UI to gameplay events.	Create final HUD icons & layout.	UI sound effects.	Test UX clarity, flow, readability.
		"Press E to harvest."	Implement transitions & animations.	Design crafting interface.		
		"Craft your first turret."		Wave banner graphics.		
		"Repair your base."		Tooltip/tutorial popups.		
Milestone 7 — Balance, Feel, Optimization (Weeks 21–24)						
Sprint 3.3						
Weeks 21–24		Balance waves, resource yields, ammo scarcity.	Add object pooling.	Improve contrast, readability, VFX polish.	Final sound mix.	Full balance playtests.
		Balance turret efficiency.	Optimize physics & collisions.		Add dynamic layer for wave moments.	Bug sweeps & regression passes.
		Improve session pacing (10–15 min).	Reduce overdraw.			
			Improve frame pacing.			

Post-Production Milestones (Weeks 25–28)			Production	GD	Eng/Dev	Art	Audio	QA	*Marketing
Milestone 8 — Release Candidate (Weeks 25–26)									
Sprint 4.1									
Weeks 25–26			Approve Known Issues list.		Final bug fixes (P0/P1).			Full regression suite.	
			Schedule external release deadline.		Final build creation (RC1).			Validate all crafting, waves, saving/loading.	
					Stability improvements.			Validate all UI.	
Milestone 9 — External Demo, Trailer, Investor Package (Weeks 26–28)									
Sprint 4.2									
Weeks 26–28			Upload demo to Itch.io (unlisted) and Steam Draft.	Write demo feature summary.	Provide stable builds for capture.	Polish key art, logo, and trailer composition assets.	Mix audio for trailer cut.	Validate final build.	Capture gameplay footage.
			Organize deliverables for investor site.	Assist with pitch materials.				Ensure trailer footage matches actual gameplay.	Edit 60–90 sec trailer.
									Write store page descriptions.