Category 5: Bug Explanation & Repair

GTB Score: Qwen 3.6 27B — 79 | Gemma 4 31B — 73 • Winner: Qwen 3.6 27B (+6 pts)

Evaluation Criteria: Evaluate: diagnosis quality, fix correctness, clarity of explanation

---

Python Game Loop Bugs (Prompts 401–430)

Prompt #401

Diagnose and fix this Python game loop: `while game_running: events = pygame.event.get() for event in events: if event.type == QUIT: game_running = False draw_frame() pygame.display.flip()` — the game window freezes after 30 seconds.

Prompt #402

Fix and explain: a Python entity update loop `for entity in entities: entity.update() if entity.dead: entities.remove(entity)` — occasionally skips entity removal or crashes with index error.

Prompt #403

Identify the bug in this delta time calculation: `dt = clock.tick(60) / 1000; player.x += speed * dt` — player moves at inconsistent speed when the game runs above 60 FPS.

Prompt #404

Diagnose: `def spawn_enemy(): enemy = Enemy(random.randint(0, WIDTH), 0); enemies.append(enemy)` called in a thread — causes random crashes in the main loop.

Prompt #405

Find and fix: a Python physics update `velocity.y += gravity * dt; position.y += velocity.y` — character sinks through platforms when FPS drops below 30.

Prompt #406

Explain the bug in this state machine: `if self.state == 'idle': self.state = 'moving'; elif self.state == 'moving': self.attack()` — enemy attacks every frame instead of once.

Prompt #407

Fix: `pygame.mixer.Sound('hit.wav')` called inside a collision check loop — causes audio stuttering and eventual crash after a few minutes.

Prompt #408

Diagnose: `score += 10 * combo_multiplier; combo_multiplier += 0.1` — multiplier grows without bound, causing score overflow after extended play.

Prompt #409

Find and explain: `if player.rect.colliderect(enemy.rect): player.health -= damage` — player takes damage every frame during overlap instead of once per collision.

Prompt #410

Fix this input bug: `keys = pygame.key.get_pressed(); if keys[K_SPACE]: player.jump()` — player can infinitely jump while holding space.

Prompt #411

Diagnose: `world_data = json.load(open('world.json'))` called in the update loop — game hitches every time a new area loads.

Prompt #412

Explain and fix: `def update(self, entities): for e in entities: if e != self: self.check_collision(e)` — O(n²) collision detection causing 1-second freeze with 200+ entities.

Prompt #413

Find the bug: `self.animation_frame += 1; if self.animation_frame > len(self.frames): self.animation_frame = 0` — occasional IndexError when accessing frames.

Prompt #414

Diagnose: a save system using `pickle.dump(game_state, file)` during gameplay — causes 2-second freeze every 5 minutes.

Prompt #415

Fix and explain: `while not game_over: update(); draw()` — CPU runs at 100% because there is no frame rate cap.

Prompt #416

Identify: `if bullet.y < 0: bullets.remove(bullet)` inside `for bullet in bullets` — bullets sometimes survive off-screen or loop crashes.

Prompt #417

Diagnose: `random.seed()` called inside `generate_level()` which is called every frame — level constantly regenerates instead of being stable.

Prompt #418

Find and fix: `player.velocity += input_vector * speed` without normalization — diagonal movement is 41% faster than cardinal movement.

Prompt #419

Explain: `pygame.draw.rect(screen, color, rect)` called before `screen.fill(background)` — sprites appear behind the background color.

Prompt #420

Fix: a GDScript autoload singleton that stores player state but resets on scene change — progress is lost when transitioning levels.

Prompt #421

Diagnose: `enemy.target = player` stored as a direct reference — when the player object is replaced on death, enemy still targets the old dead object.

Prompt #422

Fix and explain: `sound.play()` called in `_physics_process()` — footstep sound fires at physics tick rate (~60/sec) instead of once per footstep.

Prompt #423

Find the bug: a tilemap collision that works in editor but misses collisions in exported game — caused by `ResourceLoader.load()` path difference between environments.

Prompt #424

Diagnose: `if health <= 0: die()` inside an `_on_body_entered` signal — `die()` called multiple times on the same frame, causing double death logic.

Prompt #425

Explain and fix: a Python score that uses floating-point addition accumulation — score drift causes point-boundary triggers to fire incorrectly at high values.

Prompt #426

Fix: `self.health = MAX_HEALTH` in `__init__` but `MAX_HEALTH` is defined after the class — NameError on instantiation.

Prompt #427

Diagnose: `if player in sight_range: chase()` using Euclidean distance check on a tiled map — enemies chase through walls.

Prompt #428

Find and fix: a Python event system where `event_handlers = []` is defined at class level instead of instance level — all instances share the same handler list.

Prompt #429

Explain: a GDScript `yield(get_tree().create_timer(1.0), 'timeout')` inside an `if` block — if node is freed before timeout, causes null reference error.

Prompt #430

Diagnose and fix: `camera.position = player.position` updated every frame — camera jerks instead of following smoothly.

GDScript Specific Bugs (Prompts 431–460)

Prompt #431

Fix and explain: `$AnimationPlayer.play('walk')` called every frame in `_process()` — animation restarts from beginning every frame instead of playing continuously.

Prompt #432

Diagnose: a GDScript Area2D `_on_body_entered` that doesn't fire — caused by missing layer/mask configuration between colliding objects.

Prompt #433

Find and fix: `var score = 0` declared inside `_ready()` instead of at class level — score resets to 0 every time `_ready()` is called (scene reload).

Prompt #434

Explain: `set_process(true)` called in a node that is paused — `_process()` still runs because `set_process` bypasses the scene tree pause.

Prompt #435

Fix: a GDScript tween that moves a sprite to a target but overshoots — caused by `Tween.TRANS_BACK` easing applied without checking target is reachable.

Prompt #436

Diagnose: GDScript `export var speed = 100` modified in editor on instance A affects all other instances — caused by using a shared array/dictionary as a default export value.

Prompt #437

Find and explain: an enemy that doesn't stop chasing when the player dies — `player` reference not nulled on death, so `is_instance_valid()` still returns true briefly.

Prompt #438

Fix: `AudioStreamPlayer.play()` throws 'Cannot call method play on null' — stream not assigned in inspector and null check missing in script.

Prompt #439

Diagnose: GDScript `match` statement falling through to default when it shouldn't — value being matched is a float being compared to int literals.

Prompt #440

Explain and fix: a GDScript signal `emit_signal('on_death')` not received by connected node — signal name typo or node freed before signal reaches it.

Prompt #441

Find the bug: `for i in range(array.size()): array.remove(i)` — skips every other element because indices shift after each removal.

Prompt #442

Diagnose: GDScript `onready var label = $UI/HealthLabel` — NullReferenceError in exported project because node path differs from editor.

Prompt #443

Fix: a GDScript inventory system where removing an item from one character removes it from all characters — all character inventories reference the same Array object.

Prompt #444

Explain: `Input.is_action_pressed('jump')` registers in `_physics_process()` but the jump only triggers once even with button held — conflated with `is_action_just_pressed`.

Prompt #445

Find and fix: a GDScript timer that fires immediately on scene load instead of waiting — `Timer.autostart = true` left enabled in the Inspector.

Prompt #446

Diagnose: GDScript `lerp(current, target, 0.1)` called in `_process()` — movement speed varies with frame rate because delta is not factored in.

Prompt #447

Explain: `position.x = clamp(position.x, min_x, max_x)` used for camera boundary — camera stutters at boundary because clamp applied after movement.

Prompt #448

Fix: a GDScript save file written with `File.store_var()` but read incorrectly in a newer version — data corruption from variable type change between versions.

Prompt #449

Diagnose: a GDScript enemy that stutters between two positions — caused by two competing movement systems (NavigationAgent2D and direct position assignment) running simultaneously.

Prompt #450

Find and explain: GDScript `get_node('../Player')` fails in an exported build — relative path breaks when node hierarchy changes between editor and runtime.

Prompt #451

Diagnose: an A* pathfinding implementation that finds paths correctly in open areas but produces unnecessarily long paths near walls — diagonal heuristic inconsistency.

Prompt #452

Fix and explain: a procedural dungeon generator that occasionally produces inaccessible rooms — connectivity check missing after room placement.

Prompt #453

Find the bug: a shuffle function using `array[i], array[random.randint(0, n-1)]` — produces non-uniform distribution (Fisher-Yates violation).

Prompt #454

Diagnose: a card game deal function — same hand is dealt every game session because `random.seed()` is never called (or called with fixed value in testing).

Prompt #455

Explain and fix: a binary search on a sorted loot table that returns wrong items — off-by-one error in the high boundary calculation.

Prompt #456

Find and fix: a wave function collapse implementation that occasionally enters an unsolvable state — contradiction propagation stops too early when a cell has zero valid states.

Prompt #457

Diagnose: a turn-based combat queue that processes the wrong unit when two units have the same initiative — missing stable sort or tie-breaking criteria.

Prompt #458

Fix: a fog of war calculation that leaves previously-seen tiles always visible even after moving away — dirty flag not reset when player moves.

Prompt #459

Explain: a collision detection function using `rect.colliderect()` that misses fast-moving projectile collisions — discrete collision detection without tunneling prevention.

Prompt #460

Find the bug: a respawn system that places the player at (0, 0) instead of the last checkpoint — checkpoint reference overwrites itself each time instead of saving position.

Logic & Algorithm Bugs (Prompts 461–480)

Prompt #461

Diagnose: a save/load system that corrupts save files when the game crashes mid-save — file written in place instead of to temp file with atomic rename.

Prompt #462

Fix and explain: an inventory sort that is inconsistent — items with equal primary sort key not stably sorted, causing visual flickering on repeated sorts.

Prompt #463

Find: a day/night cycle that advances twice as fast on certain days — modular arithmetic error in the hour calculation that skips midnight.

Prompt #464

Diagnose: a quest completion check that marks quests complete before all objectives are met — short-circuit evaluation exits early on `and` chain with wrong order.

Prompt #465

Explain and fix: a spawner that creates too many enemies — spawn count calculated before removing dead enemies from the tracked list.

Prompt #466

Find the bug: a crafting system that accepts incorrect recipes — ingredient check using `in` on a list (order-independent) instead of validating exact combination.

Prompt #467

Diagnose: a random event system that repeats the same 2–3 events even with 50 in the pool — `random.choice()` seeded with system time truncated to seconds, repeating seeds.

Prompt #468

Fix: a high-score table that never shows the player's score in the top 10 despite qualifying — comparison using string comparison instead of integer comparison.

Prompt #469

Explain: a state machine that gets stuck in a transition loop between two states — transition condition evaluates true in both directions simultaneously.

Prompt #470

Find and fix: a collision map generated from a 2D array where the bottom row is always marked as wall — off-by-one in the row loop range.

Prompt #471

Diagnose: a path smoothing algorithm that causes characters to clip through corners — string-pulling applied without line-of-sight verification between waypoints.

Prompt #472

Explain and fix: a turn timer that fires immediately when a turn starts — timer initialized before the turn-start delay rather than after.

Prompt #473

Find the bug: a physics simulation where objects at rest occasionally drift — floating point precision error accumulating in velocity that should reach zero.

Prompt #474

Diagnose: a procedural map that looks identical after seed = seed + 1 changes — seed passed to `random.seed()` but noise function uses its own internal state not reset by the seed.

Prompt #475

Fix: a memory system that stores player choices but the wrong choice is remembered — dictionary key collision between two similar choice IDs.

Prompt #476

Diagnose: a Python game that uses 4GB of RAM after 30 minutes — per-frame surface creation with `pygame.Surface()` not being freed.

Prompt #477

Explain and fix: GDScript gradual FPS drop over time — `connect()` called in `_process()` attaching duplicate signal connections every frame.

Prompt #478

Find the bug: Python game that spikes to 100% CPU when the pause menu is open — update loop still running during pause instead of skipping.

Prompt #479

Diagnose: loading times increase with every new game session in the same run — large textures cached but never evicted from a dictionary that grows unbounded.

Prompt #480

Fix and explain: a particle system that causes FPS to drop from 60 to 5 — particles not pooled; new objects instantiated and destroyed per emission.

Performance, Memory & Integration Bugs (Prompts 481–500)

Prompt #481

Find: GDScript scene transitions cause 3-second freeze — large scene loaded synchronously on the main thread; should use `ResourceLoader.load_interactive()`.

Prompt #482

Diagnose: mobile game battery drain disproportionate to gameplay — `_process()` running physics calculations at 120Hz on a 60Hz screen.

Prompt #483

Explain and fix: a GDScript tile map that's slow to render large maps — each tile drawn individually with `draw_rect()` instead of using TileMap's built-in batched rendering.

Prompt #484

Find the bug: Python game allocates new path every frame for every enemy — A* recalculated every `_process()` call instead of on destination change only.

Prompt #485

Diagnose: a Python game that crashes with MemoryError during level load — entire level's entity data loaded into memory at once instead of streamed in chunks.

Prompt #486

Fix: GDScript animation causes jitter at high entity counts — `AnimationPlayer.play()` called with no check for currently playing animation, interrupting and restarting.

Prompt #487

Explain: Python game audio memory leak — `pygame.mixer.Sound()` objects created per sound event instead of pre-loaded and reused.

Prompt #488

Find and diagnose: a GDScript shader causing mobile GPU overheat — fragment shader doing expensive per-pixel lighting calculations that could be baked into a texture.

Prompt #489

Fix: Python save game takes 8 seconds — `json.dumps()` called on entire game state including full map data; map should be saved separately and referenced by seed.

Prompt #490

Diagnose: a GDScript multiplayer game that desyncs after 10 minutes — floating point physics accumulation differs between client and server due to delta time rounding.

Prompt #491

Diagnose: a GDScript input system that works in-editor but ignores certain keys in exported HTML5 build — browser intercepts key events (arrow keys, F keys) before Godot receives them.

Prompt #492

Fix and explain: a Python game networking bug — `socket.recv(1024)` assumes complete message arrives in one packet; large messages split across packets are corrupted.

Prompt #493

Find: a GDScript scene with a `_ready()` error 'Node not found: /root/GameManager' — autoload registered in the wrong order in Project Settings.

Prompt #494

Diagnose: a Python game that saves correctly on Windows but corrupts on Linux — file opened in text mode ('w') instead of binary mode ('wb'); newline translation differs by OS.

Prompt #495

Explain and fix: a GDScript achievement system that awards achievements multiple times — achievement checked in `_process()` without a 'has_been_awarded' flag.

Prompt #496

Find the bug: a Python multiplayer game where player positions are 1 frame behind — position sent before update instead of after update in the game loop.

Prompt #497

Diagnose: a GDScript UI that breaks on non-16:9 screen ratios — fixed pixel positions used instead of anchor-based layout; Container nodes not used.

Prompt #498

Fix: a Python game that loads mod files in alphabetical order, but mods have load-order dependencies — dependency resolution required before alphabetical sort.

Prompt #499

Diagnose: a GDScript game that works on desktop but crashes on mobile when loading a specific level — mobile GPU doesn't support a shader feature (e.g., SCREEN_TEXTURE) used in desktop shader.

Prompt #500

Find, explain, and fix: a complete GDScript combat system where: (a) damage sometimes isn't applied, (b) the death animation plays but the enemy doesn't disappear, and (c) the player occasionally attacks twice per button press. Identify all three bugs and provide a corrected version of the combat function.
