

Python Game Development II: Intermediate

Take your pygame-ce games from prototype to product: sprites at scale, smooth motion and particles, physics you can tune, scenes and save saves, tile-based platformers, generated worlds, UI and tools, NPC AI, juice and sound, two genre studies, a tower defense capstone, and a build anyone can download.

FORMAT**Instructor-led****DURATION****14 sessions · 42
hrs****SESSION****3 h incl. a break****LEVEL****Intermediate**

HOW THE COURSE RUNS

Fourteen **three-hour sessions**, one module each. Every session has the same shape: a warm-up that starts from last session's programs, two lessons (Session 13 is one capstone lesson), a break, and a wrap-up with an exit ticket. Each lesson runs **frame → live-code → guided build → check**: the instructor frames the idea and live-codes the lesson's examples, then the room builds the Practice Exercise together up to a named step.

The lessons hold more than a room can do: each module is **3 to 3.5 hours of content** for a learner working alone, against about 2.5 hours of lesson time in the room. So the room does the framing, the demos and a **guided build**, and the remaining exercise steps, the Going Further challenges and the learning journal **finish at home**. Each session's schedule below names what the room covers; plan on **1 to 2 hours of homework** a week.

Every lesson lives on the free course site, with the complete example solution, a quick quiz and a journal prompt, so a missed session can be caught up at home. The course ends with a **capstone**: a tower defense game built in Session 13, extended with a feature of your own, and shipped as a download after Session 14.

Weekly: 14 weeks, one session a week; the best fit for the homework load

Twice weekly: 7 weeks; ask for the at-home steps before the next session

Two terms: Sessions 1–7 (motion, physics, structure, platformers), then 8–14 (systems, genres, capstone, shipping)

MOTION & PHYSICS

Sessions 1–4

GAME STRUCTURE

5–7

GAME SYSTEMS

8–11

STUDIO

12–14

SESSION SCHEDULE

Times assume a 9:00 start. Shift the whole block to fit your schedule (an afternoon or evening run works the same way).
Lesson titles link to the lesson pages on the course site.

SESSION 1 Sprites at Scale

MOTION & PHYSICS

By end of session: organize hundreds of sprites with groups, draw layers and pixel-perfect masks, then keep a big scene fast with a spatial hash and object pools, measured rather than guessed.

Deliverable: Wave Defender (layers, masks, hit flash) and Swarm (spatial hash, spark pool)

PART 1 Module 1 · Sprites at Scale (I)

9:00–10:47

- | | | |
|-------|-----|--|
| 9:00 | | Welcome, orientation & pygame-ce check · Introductions, the course path, then everyone confirms pygame-ce 2.5+ and opens a window. |
| 9:20 | L01 | Groups, Layers & Masks · Live-code the rain, layers, callback and mask programs; guided build of Wave Defender TODOs 1–5. |
| 10:35 | | Break |

PART 2 Module 1 · Sprites at Scale (II)

10:47–12:00

- | | | |
|-------|-----|---|
| 10:47 | L02 | Spatial Hashing & Object Pools · Hash, timing, pool and cache demos; guided build of Swarm TODOs 1–3. |
| 11:50 | | Wrap-up & exit ticket · Two Swarm HUDs on screen, recap, homework, what to bring to Session 2. |

SESSION 2 Smooth Motion & Particles

MOTION & PHYSICS

By end of session: build the course's easing library with frame-rate-independent smoothing and angle and color blending, then drive a pooled particle system with gravity, drag, fades and additive glow.

Deliverable: Easing Playground and a Fireworks Fountain, plus one particle recipe of your own

PART 1 Module 2 · Smooth Motion & Particles (I)

9:00–10:37

- | | | |
|-------|-----|--|
| 9:00 | | Warm-up & review · Swarm and Wave Defender from home: one HUD on screen, three quick questions, easing.py copied into place. |
| 9:15 | L03 | Interpolation & Easing · Lerp, curves, easing.py, tweens, damp(), lerp_angle; guided build of Easing Playground TODOs 1–4. |
| 10:25 | | Break |

PART 2 Module 2 · Smooth Motion & Particles (II)

10:37–12:00

- | | | |
|-------|-----|---|
| 10:37 | L04 | Particle Effects · Burst, colors, emitters, pool, glow; guided build of Fireworks Fountain TODOs 1–3. |
| 11:50 | | Wrap-up & exit ticket · Two fountains side by side, recap, homework, what to bring to Session 3. |

SESSION 3 Forces & Gravity

MOTION & PHYSICS

By end of session: move objects with forces, mass and semi-implicit Euler on a fixed timestep that behaves the same at any frame rate, and design platformer jumps from a height and a time.

Deliverable: Thruster Drift and Moon Jumper (with a world of your own design)

PART 1 Module 3 · Forces & Gravity (I)

9:00–10:42

9:00 **Warm-up & review** · Fireworks and recipes from home: one fountain on screen, three quick questions, the Module 3 starters copied.

9:15 **L05** **Velocity, Acceleration & Timesteps** · $F = ma$, semi-implicit Euler, exp drag, the accumulator; guided build of Thruster Drift TODOs 2–7.

10:30 *Break*

PART 2 Module 3 · Forces & Gravity (II)

10:42–12:00

10:42 **L06** **Gravity & Jumping** · Gravity, terminal velocity, designed jumps, short hops, landing; guided build of Moon Jumper TODOs 2–5.

11:50 **Wrap-up & exit ticket** · Two jumpers at different frame rates, recap, homework, what to bring to Session 4.

SESSION 4 Bounces & Impacts

MOTION & PHYSICS

By end of session: bounce with restitution, slide to a real stop with Coulomb friction, roll on slopes, and resolve circle collisions with impulses where mass decides who goes where.

Deliverable: Material Tester and Pool Break

PART 1 Module 4 · Bounces & Impacts (I)

9:00–10:45

9:00 **Warm-up & review** · Jumpers and worlds from home: one on screen, three quick questions, the Module 4 starters copied.

9:15 **L07** **Bounce & Friction** · Restitution, materials, friction, slopes, spin; guided build of Material Tester TODOs 2–6 and a prediction round.

10:33 *Break*

PART 2 Module 4 · Bounces & Impacts (II)

10:45–12:00

10:45 **L08** **Impulse Collisions (Circles)** · Momentum, the four steps, push apart, walls; guided build of Pool Break TODOs 2–5.

11:50 **Wrap-up & exit ticket** · The heavy-cue break on the projector, recap, homework, what to bring to Session 5.

SESSION 5

States, Scenes & Saves

GAME STRUCTURE

By end of session: replace piles of flags with Enum states, scene classes and a push/pop stack with fades, then save progress with dataclasses, validation, version migration and atomic writes.

Deliverable: Coin Dash (menu, play, pause and game-over scenes) and Save Slots

PART 1 Module 5 · States, Scenes & Saves (I)

9:00–10:37

9:00		Warm-up & review · Two Pool Break runs on the projector; "where does a menu go in this program?" sets up the session.
9:15	L09	Game States & Scenes · FSM + Enum demo, scenes, the stack, fades, the asset cache. Guided build: Coin Dash TODOs 2–4.
10:25		<i>Break</i>

PART 2 Module 5 · States, Scenes & Saves (II)

10:37–12:00

10:37	L10	Saving & Loading · JSON surprises, @dataclass round trip, atomic writes, validation, migrations. Guided build: Save Slots TODOs 2–5.
11:50		Wrap-up & exit ticket · Break a save file live and watch the game shrug it off; homework; bring list for Session 6.

SESSION 6

Tile Worlds

GAME STRUCTURE

By end of session: build levels from tile grids with culled drawing, per-axis collision and JSON level files, then follow the player with smooth, clamped cameras (deadzone, look-ahead, zoom).

Deliverable: Coin Cave and Camera Modes

PART 1 Module 6 · Tile Worlds (I)

9:00–10:37

9:00		Warm-up & review · Coin Dash and Save Slots on the projector; the string-keys surprise as the bridge into Tile Maps.
9:15	L11	Tile Maps · Rows of ids, floor division, culling, per-axis collision, the 1 px probe, levels as JSON. Guided build: Coin Cave TODOs 1–4.
10:25		<i>Break</i>

PART 2 Module 6 · Tile Worlds (II)

10:37–12:00

10:37	L12	Cameras · World vs screen space, locked vs smooth ($1 - \exp(-k \cdot dt)$), deadzone, look-ahead, clamping, zoom.
11:50		Wrap-up & exit ticket · Four modes side by side on one volunteer's machine; homework; bring list for Session 7.

SESSION 7

Platformer Feel

GAME STRUCTURE

By end of session: write a character controller with acceleration, air control, coyote time, a jump buffer and a jump cut, then add depth with looping parallax layers.

Deliverable: Forgiving Jump and Parallax Walk

PART 1 Module 7 · Platformer Feel (I)

9:00–10:55

9:00 Warm-up & review · Coin Cave and Camera Modes on the projector; "which lines of Coin Cave would a great platformer keep?" as the bridge.

9:15 **L13** Character Controllers · Acceleration and friction with `approach()`, the skeleton, coyote time and the buffer, the jump cut, State and Animator.

10:43 Break

PART 2 Module 7 · Platformer Feel (II)

10:55–12:00

10:55 **L14** Parallax Scrolling · Scroll factors, tiling with modulo, back-to-front drawing, SRCALPHA haze.

11:50 Wrap-up & exit ticket · Both builds side by side; homework; bring list for Session 8.

SESSION 8

Chance & Generated Worlds

GAME SYSTEMS

By end of session: use seeded generators, weighted tables, shuffle bags and pity timers so luck feels fair and can be replayed, then grow islands and biomes from fractal value noise.

Deliverable: Loot Drops and Island Biomes

PART 1 Module 8 · Chance & Generated Worlds (I)

9:00–10:32

9:00 Warm-up & review · Two Forgiving Jump / Parallax Walk demos from Session 7, a three-question recall, the dice hook.

9:15 **L15** Randomness for Games · Seeds and own generators, weighted choices, shapes of chance, fair luck, spawn timers.

10:20 Break

PART 2 Module 8 · Chance & Generated Worlds (II)

10:32–12:00

10:32 **L16** Noise Terrain & Biomes · White vs value noise, 2D lattice, octaves, biomes, build once. Guided build: `island_biomes_starter.py` TODOs 2–6.

11:50 Wrap-up & exit ticket · Islands on screen by seed, recap, homework, preview Session 9.

SESSION 9 UI & Tools

GAME SYSTEMS

By end of session: anchor a HUD over a scrolling camera with a smooth health bar, minimap and dialog box, then build a level editor with brushes, undo and redo, validation and safe saves.

Deliverable: HUD Overlay and a Level Editor (with a teach, test, twist level)

PART 1 Module 9 · UI & Tools (I)

9:00–10:39

9:00 Warm-up & review · Seeds on screen from Session 8, recall, the "dashboard" hook.

9:15 **L17** UI & HUD · World vs screen space, anchors, text caching, a smooth health bar, a local-coordinate minimap, a dialog box.

10:27 Break

PART 2 Module 9 · UI & Tools (II)

10:39–12:00

10:39 **L18** Building a Level Editor · Pacing, grid data, stroke painting, snapshot undo/redo, validation, atomic save, constant-speed paths.

11:50 Wrap-up & exit ticket · One learner's level saved and reloaded live, recap, homework, preview Session 10.

SESSION 10 NPC AI

GAME SYSTEMS

By end of session: give a guard believable behavior with a state machine (patrol, chase, attack, search, investigate, flee), then move units through tile worlds with A* and path smoothing.

Deliverable: Guard NPC and Click to Move

PART 1 Module 10 · NPC AI (I)

9:00–10:32

9:00 Warm-up & review · A level from the Session 9 editor, recall, the "museum guard" hook.

9:15 **L19** NPC State Machines · Enum + state classes + a checking machine, perception, safe movement, dt timers, hysteresis, any-state flee, debug drawing.

10:20 Break

PART 2 Module 10 · NPC AI (II)

10:32–12:00

10:32 **L20** A* Pathfinding · Maps as graphs, BFS, Dijkstra with heapq, A* and heuristics, the corner rule, following and smoothing.

11:50 Wrap-up & exit ticket · A replan on the projector, recap, homework, preview Session 11.

SESSION 11

Juice & Audio

GAME SYSTEMS

By end of session: make hits feel good with trauma-based screen shake, hit stop, flashes and tweens (with a reduce-motion option), then mix sound with volume buses, spatial panning and crossfades.

Deliverable: Juicy Hit and Spatial Sound

PART 1 Module 11 · Juice & Audio (I)

9:00–10:47

9:00		Warm-up & review · Guard and click-to-move demos from Session 10, recall, sound check, the pinball hook.
9:15	L21	Screen Shake, Tweens & Juice · Trauma shake, five modes, hit stop and flashes, tweens with a correct yoyo, reduce motion.
10:35		<i>Break</i>

PART 2 Module 11 · Juice & Audio (II)

10:47–12:00

10:47	L22	Audio Mixing & Spatial Sound · Buses, a channel pool, constant-power panning, <code>set_source_location</code> , a non-blocking crossfade.
11:50		Wrap-up & exit ticket · Juice off vs on, recap, homework, preview Session 12.

SESSION 12

Genre Studio

STUDIO

By end of session: study two genres end to end: match-3 matching, cascades and reshuffles and an always-solvable sliding puzzle, then RPG stats, XP, equipment, a quest that pays once, and saves.

Deliverable: Gem Swap and Village Quest

PART 1 Module 12 · Genre Studio (I)

9:00–10:37

9:00		Warm-up & review · Two Module 11 demos, the Enum-typo predict, both starters open and running.
9:15	L23	Puzzle Games · Matches, swaps, gravity, cascades; sliding-puzzle parity demo. Guided build: Gem Swap TODOs 1–4 (TODO 5 if time).
10:25		<i>Break</i>

PART 2 Module 12 · Genre Studio (II)

10:37–12:00

10:37	L24	RPG Systems · Stats, items, dialogue graph, quest, JSON with Enums. Guided build: Village Quest TODOs 1–3 plus equipping; TODO 4 previewed.
11:50		Wrap-up & exit ticket · Recap, homework (Gem Swap TODO 5 + a full game, Village Quest TODO 4), capstone preview, bring list.

SESSION 13

Capstone

STUDIO

By end of session: finish a complete tower defense game one system at a time: waves, waypoints, targeting and cooldowns, three kinds of shots, one scaled dt for pause and x2, juice and tests that run without a window.

Deliverable: Tower Defense (six waves, four towers, upgrades, CC0 art), finished at home

Also: the final assessment is launched: make the game your own and ship it after Session 14.

PART 1 Module 13 · Capstone (I)

9:00–10:52

9:00		Warm-up & setup · The "state vs. event" exit tickets from Session 12; starter + assets open and running.
9:10	L25	Tower Defense: the systems · Play the browser demo; data first; waypoints; targets and cooldowns; three kinds of shots; waves and phases; juice, art and tests.
10:00	L25	Build block 1 · TODO 1 spawner, TODO 2 waypoint walk, TODO 3 target + cooldown. The game becomes a game.
10:40		Break

PART 2 Module 13 · Capstone (II)

10:52–12:00

10:52	L25	Build block 2 · TODO 4 wave bonus once, TODOs 5–6 splash and frost, TODO 7 particles and shake.
11:47	L25	Playtest & balance · Swap seats and play a neighbor's game; change one balance number; one sentence each.
11:55		Wrap-up & exit ticket · Homework, the final assessment brief, bring list for Session 14 (the build session), exit ticket.

SESSION 14

Ship It

STUDIO

By end of session: freeze a game into a folder that runs without Python, find assets and save data correctly in a frozen build, zip a release, and plan a launch with version numbers, patch notes and live events.

Deliverable: A frozen, zipped Star Catcher build and Release Day; then the capstone, shipped

PART 1 Module 14 · Ship It (I)

9:00–10:42

9:00		Warm-up: capstone show-and-tell · Four or five 2-minute demos of Tower Defense; PyInstaller check on every machine.
9:15	L26	Building Executables · Path bug demo, data folder, atomic save. Guided build: Star Catcher TODOs 1–3, then a freeze with build_game.py.
10:30		Break

PART 2 Module 14 · Ship It (II)

10:42–12:00

10:42	L27	Launch & Live Ops · itch.io vs. Steam, butler, store page, funnel, launch week, semver, live events. Guided build: Release Day TODOs 1–4.
11:45		Course close · The 14-session arc, course reflection, what's next (Game Dev III), exit ticket.

COURSE DETAILS

■ Requirements

- ◆ Completed **Python Game Development I** (or a small pygame-ce game of your own with sprites, collisions and sound) and introductory Python: functions, classes, lists and dictionaries. **Enum**, **dataclasses**, **copy**, **deepcopy**, **heapq** and **recursion** are taught where they're first needed.
- ◆ A laptop (Windows, macOS or Linux) you can install software on, with **Python 3.10+** and **pygame-ce 2.5+** (not the original pygame)
- ◆ **PyInstaller** for Session 14 (free; installed with pip) and about 200 MB of free disk for builds
- ◆ VS Code or another code editor, and about 1–2 hours a week for the at-home steps

■ What's Provided

- ◆ Free lesson pages for all 27 lessons, each with examples, interactive demos, a practice exercise, an example solution and a quick quiz
- ◆ Starter files for every practice exercise, and a **setup check** script to run before Session 1
- ◆ A participant workbook: key ideas, worked examples, the guided build, predict-then-run boxes, self-check quizzes and journal prompts
- ◆ The capstone starter with free (CC0) Kenney art and its tests, and a build script for Session 14
- ◆ Instructor demos and feedback on exit tickets, labs and the capstone

■ What to Bring

- ◆ **Every session:** your laptop and charger, the workbook, and your finished programs from the session before
- ◆ **S3, S4:** a calculator (your phone is fine)
- ◆ **S9, S10, S12, S13:** a mouse (a level editor, click-to-move, Gem Swap and tower placement)
- ◆ **S11:** headphones or earbuds
- ◆ **S14:** your Tower Defense game with its `assets/` folder, PyInstaller installed, and optionally a free itch.io account

■ Assessment

- ◆ **Self-checks:** six questions per lesson in the workbook (ten for the capstone lesson), plus each lesson's quick quiz (practice, not graded)
- ◆ **Practice exercises:** two per session (Session 13 is the capstone build), each with a clear "done when" list, most with automated tests the instructor can run
- ◆ **Exit tickets:** one short question at the end of every session
- ◆ **Capstone:** the Tower Defense game finished, extended with your own feature and a measured balance change, and shipped as a frozen build with a version number, patch notes, a README and credits; scored on a 5-row, 20-point rubric

What You'll Be Able to Do

By the end of the course you can take a pygame-ce game from prototype to a finished, shippable product:

- | | |
|--|---|
| ✓ Manage hundreds of sprites with layers, masks, spatial hashing and pools | ✓ Animate with easing and particles that look the same at any frame rate |
| ✓ Simulate forces, jumps, bounces, friction and collisions on a fixed timestep | ✓ Structure a game with scenes, and save progress that survives damage and old versions |
| ✓ Build tile-based platformers with cameras, tight controls and parallax | ✓ Generate fair loot and whole worlds from a seed, and build HUDs and level editors |
| ✓ Write NPC state machines and A* pathfinding, and add juice and spatial sound | ✓ Finish a complete game and ship it as a download with versions and patch notes |

© 2026 Ray de la Paz. Python Game Development II: Intermediate. All rights reserved.

Technical trainer, full-stack developer, and course developer with 30+ years of experience · [Course home](#)

v1.0 · 14 sessions · 42 hours