

Python Game Development I: Intro to Game Dev with pygame-ce

Your first games in Python. Over seven sessions you build a game loop, draw, read the keyboard, mouse and gamepad, move and aim with vectors, bring in images, sprites and sound, and finish with a capstone arcade game you design, playtest and put on the web for friends to play.

FORMAT

Instructor-led

DURATION

7 sessions · 21 hrs

SESSION

3 h incl. a break

LEVEL

Beginner

HOW THE COURSE RUNS

One module, one 3-hour session, two lessons, two small games. Every session builds a working program for each lesson: a bouncing ball, a sunset, a jumping square, a scrolling world, a predator chase, a turret, an asteroid field, a walking hero, a coin-collecting game, a sound board, an animated hero. The room runs each lesson's framing, the live-coding demos and the guided build; the rest of each exercise, the extra programs and the Going Further challenges finish at home on the free course site.

Sessions 6 and 7 are the capstone. You design Space Salvage (fly a salvage ship, grab crates, dodge asteroids) or your own twist on it, and build its core loop in Session 6. In Session 7 you polish it, watch a classmate playtest it in silence, put it in a web browser with pygbag, package it for friends, and play everyone's game in a closing showcase.

Every lab comes with its art and sounds (all free to use, CC0) and runs on Windows, macOS and Linux. No paid tools or accounts are needed.

Weekly: one session a week for seven weeks. Between sessions: the lesson exercises (about 1 to 2 hours) and your learning journal.

Twice a week: about four weeks, the same homework between sessions.

4-day intensive: two sessions a day (one on the last day). Exercises become evening practice.

GAME BASICS

Sessions 1–2

MOTION & MATH

Session 3

SPRITES & FEEDBACK

Sessions 4–5

CAPSTONE

Sessions 6–7

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 Your First Game Window

GAME BASICS

By end of session: install pygame-ce and prove it with a setup check, write an events → update → draw game loop, move things in pixels per second with delta time so they run the same at any frame rate, and draw shapes, colors, see-through Surfaces and text.

Deliverable: The Bouncing Ball Screensaver (the same speed at 30 or 144 FPS) and the Sunset Scene (Module 1)

PART 1 Module 1 · Your First Game Window (I)

9:00–10:47

9:00 **Welcome, orientation & pygame-ce install check** · Introductions, the 7-session path to a capstone game on the web, the journal. Everyone runs `setup_check.py` until it says You have...

9:25 **L01 The Game Loop** · Live-code the first window (`first_window_starter.py`), frames and `clock.tick()`, delta time; guided build of the Bouncing Ball...

10:35 *Break*

PART 2 Module 1 · Your First Game Window (II)

10:47–12:00

10:47 **L02 Drawing Shapes & Surfaces** · The grid, colors and `clamp_color()`, the five `pygame.draw` functions, SRCALPHA Surfaces, text; guided build of the Sunset Scene...

11:50 **Wrap-up & exit ticket** · Recap, dt exit ticket, homework (Going Further lists), what to bring to Session 2.

SESSION 2 Input & Screen Space

GAME BASICS

By end of session: tell one-shot events from held keys, move with a fair diagonal and jump once per press, read the mouse wheel and gamepads, and convert between screen, math-style and world coordinates with a camera that follows the player.

Deliverable: Move and Jump (keyboard, wheel, gamepad-ready) and Pan, Click, Convert, a world bigger than the window (Module 2)

PART 1 Module 2 · Input & Screen Space (I)

9:00–10:48

9:00 **Recap & lab setup** · Two-minute dt recap from the exit tickets; show a homework ball; copy the Module 2 starters; everyone runs...

9:12 **L03 Keyboard, Mouse & Gamepad** · Hold Space demo, keyboard events, held keys and the diagonal, Paint and Gamepad (run, not typed), guided build of Move and Jump...

10:36 *Break*

PART 2 Module 2 · Input & Screen Space (II)

10:48–12:00

10:48 **L04 Coordinate Systems** · Screen space up close, math-style coordinates, world vs. screen and the camera, the follow camera (run), guided build of Pan...

11:52 **Wrap-up & exit ticket** · Recap, camera-sign exit ticket, homework, what to bring to Session 3.

SESSION 3 Vectors & Angles

MOTION & MATH

By end of session: move anything in any direction at a fair speed with `pygame.Vector2`, tell in front from behind with a dot product, write your first classes, orbit with `cos` and `sin`, aim with `atan2` and turn the short way round.

Deliverable: Predator and Prey (a vector chase with a vision check) and the Turret Tracker (aim, fire, rotate) (Module 3)

PART 1 Module 3 · Vectors & Angles (I)

9:00–10:49

9:00 **Recap & lab setup** · Camera-sign recap from the exit tickets; copy starters; everyone runs `slow_mover_starter.py` (neither square moves: that's the...

9:12 **L05** **Vectors with `pygame.math.Vector2`** · `Vector2` arithmetic, the `=` copy trap, normalize and the zero-vector crash, subtract-normalize-scale, the dot product, the...

10:37 *Break*

PART 2 Module 3 · Vectors & Angles (II)

10:49–12:00

10:49 **L06** **Trigonometry for Games** · Degrees vs. radians, orbits, waves, `atan2`, rotating a picture, turning at a limited speed; guided build of Turret Tracker. Step 5...

11:52 **Wrap-up & exit ticket** · Recap, vector-and-angle exit ticket, homework, what to bring to Session 4.

SESSION 4 Images & Sprites

SPRITES & FEEDBACK

By end of session: load images safely, choose the right kind of transparency, transform from the original, organize game objects as `Sprite` subclasses in `Groups`, and slice sprite sheets into named frames with grid math and JSON.

Deliverable: Asteroid Field (`Sprite` classes and `Groups`) and Walk the Sheet (a hero that faces the way it walks) (Module 4)

PART 1 Module 4 · Images & Sprites (I)

9:00–10:44

9:00 **Recap & lab setup** · Vector/angle exit-ticket recap; copy both lab folders with assets/; everyone runs `asteroid_field_starter.py` (one rock drawn at...

9:12 **L07** **Images & Sprite Classes** · Safe loading with a placeholder, the three kinds of transparency, transforms and tinting, the bounded rotation cache, `Sprite`...

10:32 *Break*

PART 2 Module 4 · Images & Sprites (II)

10:44–12:00

10:44 **L08** **Sprite Sheets** · Make the test sheet, grid math with `%` and `//`, `subsurface()`, prepare frames once, JSON layout; guided build of Walk the Sheet. The...

11:52 **Wrap-up & exit ticket** · Recap, frame-number exit ticket, homework, what to bring to Session 5.

SESSION 5

Collisions & Sound

SPRITES & FEEDBACK

By end of session: detect rectangle and circle collisions, stop at walls one axis at a time so the player slides, give a short safety blink after a hit, start the mixer safely and hang sound effects and music on game events.

Deliverable: Coin Dash (collect, dodge, slide along walls) and a Sound Board with volume and mute (Module 5)

PART 1 Module 5 · Collisions & Sound (I)

9:00–10:52

- 9:00 Warm-up, lab setup & audio check · Session 4 recap (Groups, get_rect), copy the Module 5 starters, run sound_board_starter.py once to see whether each machine has a...
- 9:15 **L09** Collision Detection · Demos: rects by hand, FRect snail, circles, group helpers, Wall Slide (TAB compares UNDO and PER-AXIS). Guided build: Coin Dash...
- 10:40 Break

PART 2 Module 5 · Collisions & Sound (II)

10:52–12:00

- 10:52 **L10** Sound & Music · Demos: make_sounds.py, safe mixer start, the SPACE program, music, channels running out. Guided build: Sound Board steps 1–4...
- 11:52 Wrap-up & exit ticket · Recap, exit ticket, homework, preview Session 6 (animation and the capstone kickoff), what to bring.

SESSION 6

Animation & Capstone Kickoff

CAPSTONE

By end of session: time animation frames in seconds, switch idle, walk and attack states, write a one-page design doc, and build the core loop of your capstone one verb at a time: fly, collect, dodge, score.

Deliverable: The Animated Hero, a one-page design doc, and Space Salvage Part 1, a playable core loop (Module 6)

PART 1 Module 6 · Animation & Capstone Kickoff (I)

9:00–10:34

- 9:00 Warm-up & lab setup · Coin Dash with sound (Session 5 homework), exit-ticket recap, copy the Module 6 starters, make the space_salvage folder.
- 9:12 **L11** Sprite Animation · Demos: spinning coin, the leftover-time count (25 vs 30), the Animation class, choose_state(), flip once. Guided build: Animated...
- 10:22 Break

PART 2 Module 6 · Animation & Capstone Kickoff (II)

10:34–12:00

- 10:34 **L12** Capstone Part 1: Design & Core Loop · Design doc (10 min), starter tour, then To-dos 1–7 one verb at a time (move, spawn, collect, dodge, ramp). The three recorded...
- 11:52 Wrap-up & exit ticket · Recap, exit ticket, homework, preview Session 7, bring your space_salvage folder, install pygbag at home.

By end of session: add a title and game-over flow with a restart, animations and sound, run a silent playtest and change one thing, put the game in a web browser with pygbag, package a clean source zip, and show it to the room.

Deliverable: A polished, playtested Space Salvage with playtest notes, a web build, a source zip, and the closing showcase (Module 7, the course capstone)

PART 1 Module 7 · Finish & Share (I)

9:00–10:29

- | | | |
|-------|------------|--|
| 9:00 | | Warm-up & catch-up setup · Three scores and one question from each design doc; everyone moves to the L13 starter (the finished Part 1 with polish TODOs)... |
| 9:12 | L13 | Capstone Part 2: Polish & Playtest · Guided build: To-dos 1–6 (flame, puff, sound, title and restart, mute). Then a silent playtest in pairs (one tester each) and one... |
| 10:17 | | <i>Break</i> |

PART 2 Module 7 · Finish & Share (II)

10:29–12:00

- | | | |
|-------|------------|---|
| 10:29 | L14 | Share Your Game (pygbag + zip) · Game folder, three async changes, local web build at localhost:8000, itch.io draft (demo; learners with accounts follow), README... |
| 11:34 | | Showcase · Gallery walk: half the room plays, half hosts, then swap. Sticky-note feedback. |
| 11:52 | | Course wrap-up & exit ticket · Look back at Session 1, the last journal prompt, what's next (Game Dev II), feedback. |

COURSE DETAILS

■ Requirements

- ◆ Basic Python: variables, **if**, loops, functions, lists and dictionaries (the Introduction to Programming with Python course, or equivalent). Classes are not required: Session 3 (the Vectors lesson) introduces them when your game objects first need them.
- ◆ A Windows, macOS or Linux laptop you can install software on (a setup guide is sent before Session 1); tablets can't run the labs
- ◆ Python 3.10 or newer, pygame-ce 2.5 or newer, and VS Code with the Python extension: all free
- ◆ Internet access to install the software, and for Session 7's browser build (pygbag downloads its runtime the first time)
- ◆ Optional: a free itch.io account to publish your game on the web

■ What's Provided

- ◆ A setup guide with a one-line setup check to run before Session 1
- ◆ The full 14-lesson course workbook (yours to keep)
- ◆ Starter files for every lab, with all the art and sounds they use
- ◆ Capstone templates (design doc, playtest notes, README) and a catch-up starter so nobody falls behind between Sessions 6 and 7
- ◆ The free online course: every lesson with live examples, exercises, quizzes, and a learning journal

■ What to Bring

- ◆ Every session: your laptop and charger; your workbook; your learning journal
- ◆ S1: the setup guide done (the session opens with an install check)
- ◆ S2: a mouse with a scroll wheel and a gamepad, if you have them (both optional)
- ◆ S3: paper and a pencil for drawing arrows and angles
- ◆ S4: optional: one or two small PNG images with transparent backgrounds, and a note of where each came from
- ◆ S5 and S7: headphones or earbuds (your games start making sounds)
- ◆ S6: one idea for your own twist on a fly-collect-dodge game
- ◆ S7: your `space_salvage` folder, and `pygbag` installed at home after Session 6

■ Assessment

- ◆ Every session: the guided builds checked against each lesson's "You are done when" list
- ◆ Self-checks: a six-question quiz per lesson in the workbook, no grades
- ◆ Capstone: Space Salvage (or an equal-scope twist), playtested, shared and shown in the Session 7 showcase, submitted within a week and scored on the course rubric
- ◆ Completion: no rubric row below 2 of 3, a source zip that runs when unzipped, and playtest notes with one change (8 rows: movement & controls, collect/dodge & difficulty, code organization, game flow & feel, sound, design & playtest, web build, source package & showcase)
- ◆ Below the bar? Targeted feedback and a resubmission

What You'll Be Able to Do

Graduates leave with a dozen small games and one finished game anyone can play from a link.

- | | |
|---|--|
| ✓ Write a game loop that runs the same at any frame rate | ✓ Draw shapes, text, images and sprite-sheet frames |
| ✓ Handle keyboard, mouse and gamepad input fairly | ✓ Move, aim and rotate with vectors and trigonometry |
| ✓ Organize game objects as sprite classes in groups | ✓ Detect collisions and respond to walls, pickups and hits |
| ✓ Bring games to life with animation, sound effects and music | ✓ Design, playtest, polish and publish a game on the web |

© 2026 Ray de la Paz. Python Game Development I: Intro to Game Dev with pygame-ce. All rights reserved.

Technical trainer, full-stack developer, and course developer with 30+ years of experience · [Course home](#) · [Previous course: Introduction to Programming with Python](#) · [Next course: Python Game Development II](#)

v1.0 · 7 sessions · 21 hours