

Python Camp — Day 5

Finish & Show Off

David Pouliot

Southern Oregon University

Today

- Finish and polish your game
- Add the pieces that make it feel **done**
- **Gallery walk** — play everyone's games
- Save your code to take home

Polish

A start screen

Wait for a key press before the game begins.

```
1 waiting = True
2 while waiting:
3     screen.fill((0, 0, 0))
4     # draw "Press any key to start"
5     pygame.display.flip()
6     for event in pygame.event.get():
7         if event.type == pygame.KEYDOWN:
8             waiting = False
```

Show text on screen

Score, lives, "Game Over" — all use the same three steps.

```
1 font = pygame.font.SysFont(None, 48)    # once, before the loop
2
3 # ...each frame:
4 label = font.render(f"Score: {score}", True, (255, 255, 255))
5 screen.blit(label, (10, 10))
```

Polish checklist

- Start screen (press any key to begin)
- Game-over screen with the final score
- A lives system — 3 lives, then game over
- Track the high score across rounds
- Background music with `pygame.mixer.music`

Pick the ones that make **your** game more fun. Don't try all five.

Way ahead? Capstone challenges

Game already polished? **Boss challenge** — combine your week:

- Pull live data from a Day 2 API into your game — a weather background, or a real Pokémon as an enemy.
- Log each game's final score to a file, then chart your high scores with matplotlib (Day 3).
- Add a menu that lets the player pick which of your games to play.

Totally optional — a finished, fun game beats an unfinished ambitious one.

Gallery Walk

Gallery walk

- Get your game running on your screen and leave it up.
- Walk around and play everyone else's games.
- No presentations, no pressure — just play.

Have fun. You built these this week.

Take your code home

- Email the .py files to yourself, or
- Save your project folder to a USB drive.

Want to keep going? `docs.python.org`, the Pygame docs, and Advent of Code are all great next steps.

The rest of the Python world

Games were just the start. The **same language** you learned this week runs in all of these:

- **Web & servers** — Flask, Django, FastAPI: real websites
- **APIs** — pull in data, or build your own service
- **Automation** — scripts to rename files, scrape sites, send email
- **Data science** — pandas & NumPy on huge datasets
- **AI / ML** — scikit-learn, PyTorch, TensorFlow
- **Apps & bots** — desktop GUIs, Discord bots

One language, all of it. Pick whatever sounds fun — and keep building.