

I am providing an image showing the visual style I want for my upgraded Pong game.
I already have a fully working Pong game built in:

- **Python 3.13.x**
- **arcade==3.3.3**
- A structured project directory
- Multiple modules (game window, sprites, scenes, etc.)
- Working menus and game logic

Your task is to upgrade ONLY the graphics and presentation layer while keeping all existing game logic fully intact.

✓ **Follow These Rules**

1. Keep my current code structure

Do **not** rewrite my game from scratch.

You must integrate upgrades into the following areas **ONLY**:

- Rendering system
- Sprite appearances
- Background / environment
- UI visual design
- Shaders (if applicable to arcade 3.3.3)
- Particle and glow effects
- Motion trails
- Menus and overlays

2. Preserve untouched:

- Game mechanics
- Scoring
- Player input
- Collision logic
- Physics behavior (unless visual-only changes are required)
- Directory layout
- Class names and file names (unless adding new modules)

✓ **Theme & Visual Style (From Attached Image)**

Rebuild the visual look to match:

- Retro-futuristic neon cyber-city
- Purple/blue/pink neon palette
- Glow effects
- Perspective grid floor
- Slight atmospheric depth haze
- Bright outlines around paddles and ball
- Fake 3D or shaded look

✓ Upgrades You Must Add

1. Neon City Background (based on attached image)

- Add a new background texture or layered drawing
- Add depth (foreground grid + distant skyline)

2. Pseudo-3D Paddle and Ball Graphics

- Gradient shading
- Soft glow
- Optional: light bloom using arcade shaders if compatible

3. Ball Motion Blur Trail

- Smooth fading trail that matches speed
- Neon color that blends with the grid/palette

4. Neon UI Overhaul

Upgrade all menus to match the neon-city aesthetic:

- Main Menu
- Pause Screen
- Game Over Screen
- Title text
- Buttons
- Score UI

All should match the palette and glowing style.

5. Minimal Code Invasion

All updates should be done by:

- Extending existing classes
- Overriding draw functions
- Adding helper modules
- Adding assets
- Adding shaders if beneficial

Do **not** break or rewrite the underlying game.

✓ **Output I Want From You**

1. **A step-by-step modification plan**
 - Specific files to modify
 - Where to insert new code
 - What new modules to add (if any)
2. **Upgraded code blocks** for only the parts that change
 - draw methods
 - sprite shading
 - background rendering
 - UI updates
 - motion blur system
 - shader setup
3. **Asset instructions**
 - How to use the provided image
 - Any necessary resizing or reformatting
 - Where to place assets in the project directory
4. **No rewriting the whole game**, unless I explicitly ask for a full overhaul later.

✓ **Attached Image**

Use the attached image **ONLY** for visual/style references—not for new game logic.

When ready, produce the step-by-step plan, then the updated code segments, and then the asset instructions.