RTX 3060 AI Coding Setup — Supplemental Improvements

This document adds clarification, optimization, and advanced tips based on real-world usage of the RTX 3060 + Ollama + Continue CLI coding workflow. It is designed to stand alone — no need to refer back to the original guide.

---

1. Quick TL;DR Setup (Minimal Steps)

For experienced users who just want the GPU working:

```
1. Install RTX 3060 + power + boot
2. Install NVIDIA drivers (GeForce Experience recommended)
3. Install CUDA Toolkit 12.x
4. Install Ollama for Windows
5. Open PowerShell and run:
   ollama pull qwen2.5-coder:14b
6. Test GPU:
   nvidia-smi
7. Test model:
   ollama run qwen2.5-coder:14b
8. On Mac, set Continue config:
   apiBase: https://YOUR-WINDOWS-IP
```

This alone gets you a 14B coding model running at ~25 tokens/sec.

---

2. CUDA + Driver Compatibility Reminder

It's not necessary to install the *absolute latest* CUDA release.

Key rule:

> Your NVIDIA driver and CUDA version must be compatible, but CUDA does NOT need to match the version Ollama was built with.

When in doubt: - Install the latest NVIDIA driver - Install CUDA 12.2 or newer - Verify with:

```
nvcc --version
nvidia-smi
```

---

3. When NOT to Use 32B Models

32B parameter models (e.g. qwen2.5-coder:32b) will sometimes *load* — but are usually:

- Slower
- VRAM stretched to the limit (11–12GB)
- Prone to out-of-memory errors

Use 32B models only if: - You're certain you need deeper reasoning - You accept performance hits - Nothing else answers your queries

For almost all coding: **14B → best balance 16B → best large context option**

---

4. GPU Monitoring While Running Models

To verify load, VRAM, heat, and utilization, open two terminals:

Terminal A:

```
nvidia-smi -l 1
```

Terminal B:

```
ollama run qwen2.5-coder:14b
```

You should see: - 8–10GB VRAM in use - 80–100% GPU utilization - Temps: ~60–75°C

---

5. Optional Tools for Live Monitoring

You may use: - **GPUTop:** Linux-style live usage - **OpenHardwareMonitor:** Windows temps + VRAM - **GreenWithEnvy:** (Linux, but excellent)

These are optional — `nvidia-smi` is enough.

---

6. Disable Windows "Hardware-Accelerated GPU Scheduling"

Sometimes causes latency spikes.

Steps: - Windows Search → **Graphics Settings** - Turn OFF "Hardware-Accelerated GPU Scheduling" - Reboot

---

7. Add Temperature Stabilizing to Continue Models

Example addition under each model config:

```
temperature: 0.2
```

Why? - Lower temperature = - More deterministic coding - Fewer hallucinations - Cleaner diffs and patches

---

8. CLI Speed Test Command

So you can directly measure improvements:

```
time ollama run qwen2.5-coder:14b <<< "Write Fibonacci in Python"
```

Expected on RTX 3060: - 25–30 tok/sec - Full reply in 3–6 seconds

---

9. When to Use Quantized (-q4) Models

Quantized models are smaller and use less VRAM.

Example:

```
ollama pull qwen2.5-coder:14b-q4
```

Use quantization if: - You want **two models in VRAM at once** - You push up against 12GB VRAM - You run background GPU tasks - You want to prevent OOM failures

Trade-off: - ~5–10% quality loss - ~50% VRAM savings

---

10. Auto-Run Ollama at Boot (Windows)

If Ollama sometimes fails to start:

1. Open: `services.msc`
2. Find: **Ollama Service**
3. Right-click → Properties
4. Set **Startup Type → Automatic**
5. Apply, OK

This ensures your Mac can always connect.

---

11. Recommended GPU Upgrade Paths

If you later want: - **More VRAM** - **Bigger models** - **Lower power draw**

Top upgrade suggestions:

16GB - RTX 4070 Ti Super (16GB)

24GB - RTX 3090 (used — great value)

High end - RTX 4080 Super (16GB)

---

12. Optional Alternative IDE Workflow

If you prefer VS Code instead of Continue CLI:

• Install "Continue" VS Code extension
• Same config.yaml
• Features:
• Inline code edit
• Panel-based chat
• Auto model switching

But Continue CLI **remains faster + lighter**.

Summary of These Improvements

• Faster setup via TL;DR
• Safer CUDA/driver guidance
• Clear warnings about 32B models

- GPU monitoring instructions
- Windows scheduling fix
- Stable `temperature: 0.2` recommendation
- Built-in speed benchmark command
- When to use quantized models
- Auto-start Ollama fix
- Future GPU upgrade routes
- Optional IDE workflow info

These enhancements make the RTX 3060 AI coding workflow more stable, smoother, and easier to maintain — especially for multi-day development sessions.