AutoCon 6 Workshop FAQ

WS:C1 · Loop Engineering, Spec-Driven Development & the Future of VibeOps

Four hours. One networking concept. A playable game in your own GitHub repo.

1. What should I bring?
Bring your laptop, your charger, and an active account for a frontier AI coding assistant (Claude Code or OpenAI Codex). Most of the morning is hands-on: you'll set up a repo, run GitHub Spec Kit, and build your own game with an AI coding agent.

2. Do I need access to a specific AI model or coding assistant?
Yes. You'll need Claude Code or OpenAI Codex running in VS Code with a terminal. The workshop's reference models are Claude Sonnet 5 and GPT-5.6, and lower-cost models also work.

For the best experience, use GPT-6 Astra (OpenAI) or Claude Opus 5.5 (Anthropic). They're the most expensive options, and the cost climbs as your context window fills, but they produce the strongest specs, plans, and code in the time available.

Whichever model you choose, confirm before you arrive that your plan or API key gives you access to it inside your coding assistant. If you plan to run a Ralph loop, set an iteration cap, because loops spend tokens on every pass.

3. What software should I install before the workshop?
Install and test these ahead of time so the 10-minute setup block is a check, not a scramble:

- Visual Studio Code (VS Code)
- Claude Code or OpenAI Codex, signed in
- Git and a GitHub account (the GitHub CLI, gh, is optional but turns repo setup into one command)
- Python 3.11 or later, which Spec Kit needs and which also serves your finished game locally
- uv, which installs the Spec Kit CLI
- GitHub Spec Kit: run uv tool install specify-cli --from git+https://github.com/github/spec-kit.git, then specify version to confirm it worked

Optional for Claude Code users: the Ralph loop plugin, installed with /plugin install ralph-loop@claude-plugins-official.

Want a head start? The workshop repo at github.com/automateyournetwork/NAF_AC6 has the lab walkthrough, the starter templates, and two worked examples.

4. Do I need programming or network automation experience?
Basic networking and Linux knowledge is enough. Intermediate programming experience is recommended. You don't need any prior experience with loop engineering, Spec-Driven Development (SDD), Ralph loops, or Spec Kit. The opening lectures cover LLMs, RAG, Model Context Protocol (MCP), agent skills, Ralph loops, and SDD before you put them to work.

5. What does the day look like?
We run from 9:00 to 1:00. The first 50 minutes cover how AI coding went from a chat window to loops that prompt themselves, SDD with Spec Kit, and loop engineering, plus a live demo of Itential Builder Skills from Joksan Flores and Elliot Conner. Then comes project setup, a short session on gamifying networking, group

design time around a 30-minute break, a 60-minute build block, and a closing show-and-tell. John Capobianco, Joksan, and Elliot are on the floor for every lab, so raise a hand any time.

6. Will I be working individually or in a group?

Both. You'll join a group based on the number you're given. Your group brainstorms game ideas and personalizes three templates together (constitution, spec, and plan), spending about 10 minutes on each. Then you build solo, as a pair, or as a small team. For teams we recommend one driver: one person's repo, everyone added as a collaborator, one person typing, and everyone steering. Two room-wide checkpoints, at clarify and analyze, let you compare what your agent asked and found with the rest of the room.

7. What will we actually build during the workshop?

A browser-based, networking-themed educational game, built with Spec Kit and your AI coding agent. Three.js, HTML, and CSS are the suggested stack. You'll run seven Spec Kit commands in order (constitution, specify, clarify, plan, tasks, analyze, and implement) and review each artifact before moving to the next. Exact command names vary slightly by agent.

The recommended scope for the 60-minute build is one core mechanic, three short levels, deterministic answers, and no backend, so the game itself doesn't need an AI model to run. Want an agentic game, like a tutor character that explains answers? That's a decision for your spec. It needs an API key or a local model, and it adds token cost and accuracy risk.

Whatever you build, the starter constitution's core rule holds: the game never presents incorrect information.

8. Do I need to come with a project idea?

No. The starter templates include an Inspiration section, and the group activity is designed for refining ideas. A good recipe is to pair a networking concept you know cold with a game mechanic people already love: ACLs as Arkanoid, routing where you play as the packet, or QoS queuing as a racing game. The workshop repo also includes a few working examples including "Per-Hop Works" (a playable 3D game with twelve DiffServ lessons). Already have a concept? Bring it.

9. Can I add MCP servers or other tools to my build?

Yes, and we encourage it. Think of it as loading your build harness: anything that helps your agent build a better game, none of which has to ship inside the game. Ideas from the session include Blender or Unreal Engine MCP servers for 3D assets, the GitHub MCP server, RAG over RFCs or vendor docs, a Three.js skill, and AI-generated images or voice.

For networking-specific MCP servers, start with Itential's Ultimate MCP Guide for Network Automation, which catalogs 78 servers across 12 categories. A few fit game builds especially well. The Reference & Standards servers (Wikipedia, RFC Lookup, and Subnet Calculator) help keep your questions and explanations accurate. The diagramming servers (Draw.io, Excalidraw, and Markmap) help with topologies and visuals. Lab servers like ContainerLab let you check scenarios against a real network. Install and test any MCP server before the workshop, since setup time is short.

10. Will I leave with a working project?

That's the goal. By 12:30 you'll have a repo you own with your constitution, spec, plan, tasks, and a game you can play in your browser. Commit and push, play each other's games in your group, and each group picks two to show the room. Afterward, if you want, have your agent fill in the ARCADE.md template and send John your GitHub URL to get your game listed at automateyournetwork.com/arcade.