

AI

Genex: how to create a computer game with AI, a beginner's guide

My research on Genex covers a first game, Desktop versus Tools, Claude and Codex connections, Windows options, assets, costs and publishing.

Gilad Dor Levi · Updated October 12, 2026

Latest version online: nex-ai.co/en/guides/genex-ai-game-creation/



In my research on Genex, I checked the official website, documentation and source repository. I wanted to understand how to start from a game idea, what needs installing and what someone using Windows can explore. This guide brings those findings together with a brief for a small first game.

Genex Desktop is an open-source application connecting AI agents to game development. Describe a game in chat, get code, preview it and continue fixing it. Its current focus is browser games using Three.js. Genex Tools is a separate route for generating assets and adding tools to an existing project.

I did not install Genex or benchmark it during this research. The example below is a game brief to try, rather than a game I claim to have built and played. Product details were checked on October 12, 2026.

What is Genex, and which route should you choose?

The distinction between the application and the tools helped me make sense of the setup.

Topic	Genex Desktop	Genex Tools
Main use	Building through an application with chat and a live preview	Assets and tools for an agent working in a project
Interface	Desktop application	Skill, CLI, MCP or API
Agent system	Planning, building and reviewing roles in supported modes	Depends on your coding agent and workflow
Downloads	Apple Silicon macOS and Linux x64	Check the tool environment; CLI needs Node.js 20 or newer
Windows	Official download page still says Coming soon	Explore a compatible Tools environment; this does not confirm every component works
Cost	Application is free; models and services are separate	Priced asset generation and actions use credits

The Desktop release verified on the research date is **0.1.7**, under MIT. Its assets include macOS and Linux packages; the checked list contains no Windows installer. [Genex 0.1.7 release](#).

For chat-based game building on a supported computer, start with Desktop. If a coding agent already works on your game and you want assets, explore Tools. [Desktop overview](#) and [Tools overview](#).

How do its AI agents build and check a game?

Genex offers model choices for a planning Main Agent, building Workers and checking Reviewers. Its technical documentation says Workers and Reviewers operate in Loop mode. Do not assume every chat message launches a parallel team.

Start with an available model and a small request. When using Loop, set a bounded working period. Documentation describes budgets from 30 minutes to no preset limit. A working budget does not guarantee completion within that period. [Model roles and modes](#).

Builds shows work and agent checks. Play the result yourself: movement may work while restarting fails to restore previously collected coins.

A fast video demonstration is not a delivery-time promise. Scope, model availability and fixes affect duration. I did not measure a seven-minute build.

What can you build now?

Try a coin collector, obstacle course, shooting mechanic or tower-defense prototype. Pick one mechanic you can explain and test. These are suggested projects, not a promise of a complete commercial game.

The current route focuses on Three.js browser games, local files and static web exports. Unity, Unreal and native applications remain planned product directions. Check availability before depending on a future integration. [Product capabilities and limits](#).

Install Desktop and connect Claude or ChatGPT

1. Check your computer and download officially

The product specifies Apple Silicon for macOS. On Linux, choose the x64 package appropriate to your distribution. Open the [official latest release](#), check its version and choose the appropriate DMG, DEB, RPM or ZIP.

Install using the package instructions and open the app. The documented interface includes New game, Plugins, Settings and a game list. If your screen differs, check the version before following an older screenshot.

2. Connect a coding tool with an eligible account

Genex uses Claude Code for Claude models and the Codex CLI for OpenAI models. Install and configure the appropriate provider tool, then follow the app's connection flow. Connect ChatGPT uses browser sign-in through Codex; it does not embed the ordinary ChatGPT website in your game.

The privacy documentation says subscription limits apply and this subscription route does not automatically switch to API-key billing when its allowance runs out. Supported local models through Ollama or Bonsai depend on hardware and model capabilities. [Provider requirements and local-model use](#).

3. Choose an available model and keep setup simple

Check the model picker for options available to your account. A listed name does not prove access. Start by confirming that a simple request works before mixing several models.

Your first game: Coin Runner

This example uses a small arena, ten coins, three obstacles, a score and a restart button. Specify obstacle behavior so the agent does not have to invent the rules.

1. Open a project and send the brief

Choose New game and paste:

Build a small, playable 3D browser game using Three.js.

Game: Coin Runner.

Create a third-person player controlled with WASD or arrow keys.

Use a small colourful arena with a visible floor and boundaries.

Place 10 collectible coins and three clearly visible obstacles.

Collect each coin once and show the score in the top-right corner.

When the player hits an obstacle, return the player to the starting point without resetting collected coins.

Show a victory screen after all 10 coins are collected.

Add a Restart button that restores the player, coins and score.

Use a smooth follow camera and lightweight placeholder graphics.

Build the minimal playable version first. Test movement, boundaries, obstacle collisions, coin collection, score, victory and restart. Fix blocking bugs before adding visual polish.

Do not use paid asset-generation services or publish the game without my approval.

Keep files local and editable.

Summarize what works, what you actually tested and remaining limitations. Do not report a test as passed unless you performed it.

2. Follow progress and play the result

Open the preview when available. Read what any reported tests actually checked. A screenshot is not evidence that movement or restarting works.

- ☐ Arrow keys and WASD move the player as intended.
- ☐ Each coin can be collected only once.
- ☐ The score matches the number collected.
- ☐ Obstacle collisions follow the specified return-to-start rule.
- ☐ Victory appears after ten coins.
- ☐ Restart restores the player, coins and score.
- ☐ A full round can be completed without a blocking error.

3. Fix one problem, then add one improvement

Describe your action, the observed behavior and the expected behavior. For example:

I collected four coins and pressed Restart. The score returned to zero, but the collected coins did not reappear. Restart should restore all 10 coins, the starting position and the score. Fix this issue without changing movement or camera behavior. Test restart twice and state what you observed.

That is a hypothetical bug report, not a result from my research.

Once the game works, try a bounded improvement:

Improve lighting and add subtle shadows. Preserve movement, obstacle behavior, coin collection, score, victory and restart. Recheck these after the change. Do not add paid services or new gameplay features.

Assets, plugins and the harness

Blender supports local scene and model work, while plugins and services provide generated models, images and sound. Meshy, Tripo and ElevenLabs are among the advertised integrations. Check the connection and quote before using them. [Desktop repository and integrations](#).

The harness is the instructions and tools surrounding the agents. Self-improvement is experimental and presented as off until enabled. You do not need it for a first game. Review proposed changes and the required approval; the feature does not prove every build improves on the last.

Windows users: explore Genex Tools

Tools does not install Desktop. It adds asset creation to an agent or project you already run. Its CLI requires Node.js 20 or newer. I did not test installation on Windows, so check component requirements in your environment.

Option A: ask a coding agent to set it up

Inside your intended project, ask a compatible agent:

Set up <https://genex.games/SKILL.md> in this project.
Read the official setup instructions and explain the dependencies and project files that will change before installing.
Keep unrelated files intact. Let me complete the account authorization myself.
Do not generate paid assets or publish anything during setup.

This is a request for a future setup, not an action performed while writing this guide. The documented installer adds a CLI and skill cards, so save a version or backup first. [Skill setup instructions](#).

Option B: install the CLI in your project

```
node --version  
npx @genex-ai/cli-demo@latest tools
```

Complete account approval yourself, then check:

```
npx genex doctor
```

The check reports account status, balance and available capabilities. After setup, these example asset-generation commands may consume credits:

```
npx genex image "A golden treasure chest, game asset" --transparent  
npx genex model "A low-poly medieval castle"  
npx genex sfx "An arcade coin pickup sound"  
npx genex music "Upbeat retro arcade music, seamless loop" --duration 60
```

In a Tools workspace, files normally download into `./assets`. Use the local asset path in your game. Check a submitted generation with the waiting command instead of submitting the same creation again: another creation can incur another charge. [CLI quickstart](#) and [command reference](#).

Option C: connect MCP

The official server URL is:

```
https://mcp.genex.games/mcp
```

Add it through a compatible client's connection flow and complete OAuth. The connector can generate assets and read permitted account information, consuming your credits for generation. It returns asset URLs, unlike the CLI's local downloads. Its publishing tool supplies instructions rather than publishing directly. [MCP instructions and permissions](#).

What does Genex cost?

Separate the free application, your AI subscription and credits for services. Documentation states one credit is \$0.01, but an action's quote depends on its selected options. Five hundred credits represent \$5 before taxes and fees, not a promised number of games or assets.

For another currency, multiply the dollar amount by a rate you check at purchase time. This guide does not present an illustrative exchange rate as current.

The service documentation also describes metered builds and Blender time. Do not assume identical charges apply to all local Desktop work. Before a long operation, inspect Settings → Credits, the action quote and Build charges for the route you use. [Current pricing](#) and [billing routes](#).

Export and publish your game

After testing, export a static web bundle from Desktop to suitable hosting or use Genex's publishing route. A local preview does not mean the game is public.

A Tools project needs hosted game-workspace setup before platform publishing commands work. Read the [official publishing guide](#); setup may add configuration and upload the project to the service.

In a game workspace already configured for hosting:

```
npx genex preview
```

Check the preview, then release it when ready:

```
npx genex promote
```

To restore the previous build:

```
npx genex rollback
```

Review uploaded files, asset licenses and behavior on target devices. For phones, add touch controls and test actual performance.

Security and privacy for a first project

Use a separate game folder with backups or version control. Coding agents and native plugins can operate on files and run processes. Keep passwords and keys out of prompts, and check permissions before adding a plugin or MCP server. Playtest independently of agent checks. [Project security policy](#).

Frequently asked questions

Do I need programming experience?

You can start with a natural-language description. Basic code and debugging knowledge helps with failures, performance work and larger projects.

Can I start with ChatGPT alone?

Desktop supports Codex sign-in with an eligible account. Check installation, connection and remaining allowance. A subscription does not provide unlimited use.

Can I make a mobile game?

You can work on a browser game with touch controls. That does not establish native Android or iOS export; check that capability separately.

Can I work offline?

A local model reduces reliance on cloud inference, but downloads and external services still need a network. My research did not include a complete offline test.

Can I sell the result?

You can retain and export project files. Commercial use also requires checking asset and service licenses; the application's license does not resolve every rights question.

Does Genex replace a development team?

It is a young tool for exploring ideas and small projects. Larger games still need planning, testing, maintenance and cost checks. This guide contains no benchmark establishing replacement of a professional team.

The first step I suggest

Choose the route suited to your computer and tools. In Desktop, start with Coin Runner and simple assets, complete the checks, then consider visual polish or paid services. Keep the working version and evidence of your tests before the next change.

For clearer briefs to use with agents: [NexAi's ChatGPT command library](#).

For updates, one-click prompt copying and more resources, visit the guide online.