Guide contents

Pick a project

Cities: Skylines

Cities: Skylines II

Test and share

Two first-map projects, kept separate

Status

Sources / References

Documentation: In progress **Quick start: In progress**

Offline copy from ModDocs. Source links remain clickable.

Cities: Skylines modding

Cities: Skylines and Cities: Skylines II belong on one finding page, but their modding workflows should not be mixed. Choose the game before following a tutorial or reusing code.

> **Quick start:** choose the first or second game, then pick a map, asset, or code mod. Open that game's current editor or toolchain and make one tiny change you can verify in game. The paths below stay separate so version-specific instructions do not get mixed.

Pick a project

Project	Cities: Skylines	Cities: Skylines II
Map	First game's Map Editor and community map guides	Second game's Editor and its map workflow
Custom asset	First game's Asset Editor and import guides	Second game's asset editor; support arrived later and is version-sensitive

Project	Cities: Skylines	Cities: Skylines II
Code mod	First-game C# community guides	Second-game toolchain and project template

Begin with a tiny asset or map rather than a whole city overhaul. Keep the two games' source files and test installations separate.

Cities: Skylines

An independent community guide separates code mods, asset importing, and maps. Start with its workflow that matches what you are making. Check the guide's age against your installed game and current tools: the guide itself says parts are incomplete.

For a map, learn the first game's editor and test a small area. Check terrain, outside connections, buildable space, and whether a new city can actually begin. For an asset, prepare a model and materials for the first game's import process, then inspect it in the Asset Editor and place it in a test city. For a code mod, follow the community guide's development setup and test a small, reversible behavior change first.

The guide covers folder structure, code-mod setup, debugging, asset importing, and maps. Because it is fan-maintained and explicitly incomplete, verify instructions against your installed game version and current tool releases before relying on an exact path or API name.

- Cities: Skylines community modding guide - code, assets, maps, and debugging topics.

Cities: Skylines II

The developer describes an integrated modding toolchain, project template, and Paradox Mods publishing path for code mods. The game uses systems and components that differ from the first game's approach; a first-game code tutorial is not a direct second-game recipe. Begin with the current in-game modding tools and game-specific documentation.

Maps

The second game's Editor includes terrain tools, heightmap import/export, water sources, resources, roads, and a map checklist. A first map can be deliberately small: shape land, add water if needed, place an outside highway connection, save, then start a new city on it. The developer's Map Editor diary stresses that a highway must reach the map edge and starting tile so citizens can arrive. It also describes the heightmap and resource-map formats; check the current editor's requirements before preparing files.

Assets

Asset modding was added to the Cities: Skylines II Editor in an asset-mods patch in December 2025. Earlier articles claiming that the second game has no asset editor are outdated. Update the base game and DLC before creating or testing new assets. Start with the current editor and the wiki guides linked by the official asset-mods patch notes. Make one simple asset, place it in a test city, and check appearance, scale, collisions, and save/reload behavior before publishing.

Code

The developer describes an integrated modding toolchain and project template for Visual Studio or Rider. It provides game dependencies and build steps. The second game uses systems and components, including Unity ECS/Burst concepts, so first-game code patterns may not translate directly. Begin with a minimal project generated by the second game's current toolchain; confirm it loads in a clean test profile before adding a feature. The official code-modding diary explains the design and Paradox Mods publishing route.

- Cities: Skylines II code-modding developer diary - official overview of toolchain and code-mod concepts.

Test and share

For either game, test the final package in a clean profile and a new save. A map should support starting a city and reaching outside connections. An asset should survive save/reload. A code mod should report errors clearly and avoid changing unrelated saves. State the exact game, game version, dependencies, and supported

content type on the release page. For Cities: Skylines II, check the current Paradox Mods and editor instructions before publishing; tooling has changed since its first release.

Two first-map projects, kept separate

For **Cities: Skylines**, open the first game's Map Editor and make a small map with buildable terrain and a functioning outside road connection. Save it, start a **new city** using that map, and check whether incoming traffic can reach a starting road. A map that looks fine in the editor may be unplayable because its outside connection or starting conditions are wrong. Add one feature at a time and use the first-game community map guide for the editor's file and terrain details. Install the exported map in a clean profile and start another city from the exported copy.

For **Cities: Skylines II**, use the second game's current Editor and make the same minimal map as a separate project. The developer map guide calls out a crucial connection: the outside highway needs to reach the map edge **and** the starting tile. Save, open the map as a new city, and check arrivals before adding dense roads, water, or custom assets. Publish through the current Paradox Mods workflow only after the exported map works in a new game. A first-game map or code project is not a reusable second-game package; carry over the design, not the files.

If your goal is **code** instead of maps, start with the second game's installed code-mod toolchain and its IDE template. The developer's code-modding notes explain that the template supplies dependencies and post-build placement. Confirm the generated empty mod appears in the game before adding a system, and test the packaged build after removing the development output. Do not assume first-game Harmony hooks cover second-game Burst-compiled systems.

Status

Unfinished: these workflows are source-based but not yet verified with a built mod for each game and content type. Original sources: Cities: Skylines community guide, Cities: Skylines II map editor, code modding, and asset-mods patch.

Sources / References

External project and documentation links cited in this guide. Check the upstream release and version before following version-sensitive steps.

- Cities: Skylines community modding guide
- Map Editor diary
- official asset-mods patch notes
- official code-modding diary