

Git and GitHub

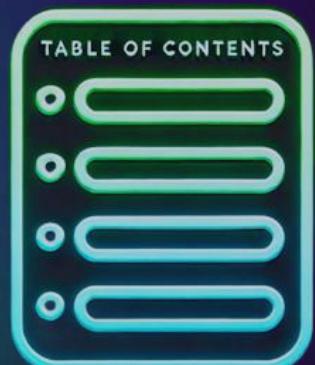
Versioning for AI-Assisted Development



Svetlin Nakov, PhD
Co-founder @ SoftUni

Agenda

1. **Git concepts**: repo, clone, commit, discard, push, pull
2. **Intro to GitHub**: repos, issues, code, commits
3. **Git in VS Code**: publish to GitHub, view changes, commit, push, discard changes
4. **Recovering from Mistakes**: checkpoints, discard changes, re-clone a repo, revert commit
5. **Import / export** projects with GitHub repo
6. **Publish app to Netlify** (from GitHub) + auto deploy
7. **Branches** and **pull requests** in GitHub
8. **Conflicts** and merges: avoiding and resolving conflicts



Sli.do Code

#Soft-Tech-AI

Join at

slido.com

#Soft-Tech-AI



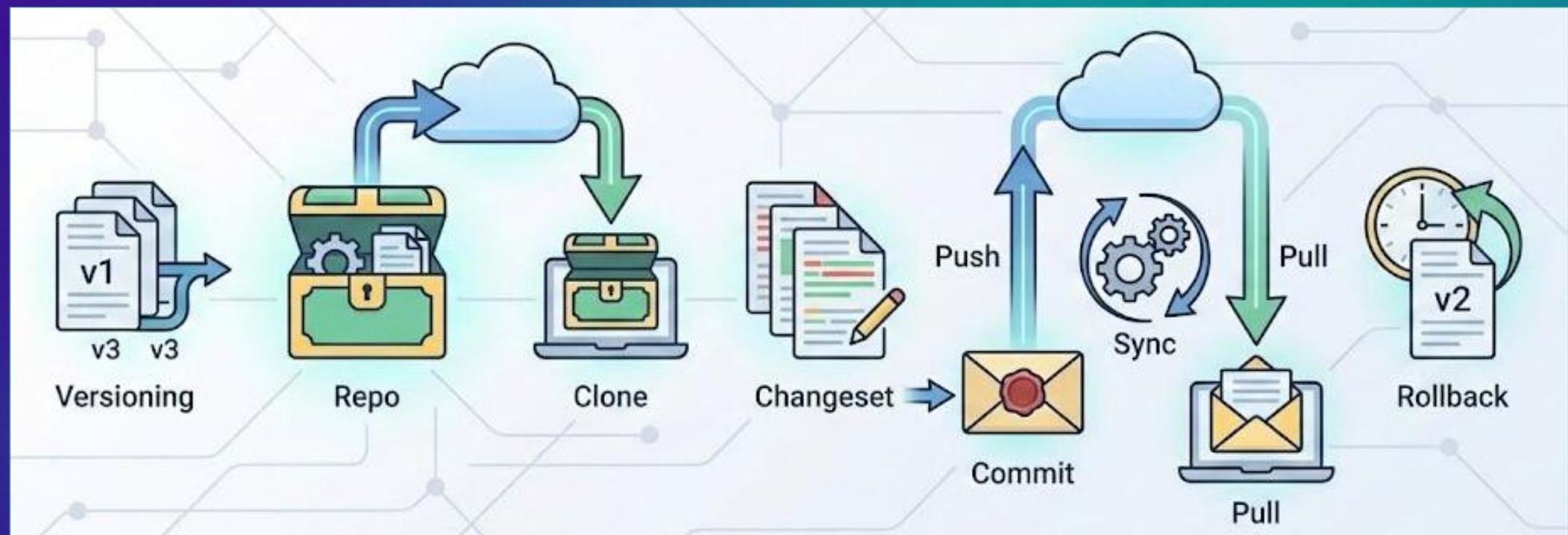
Breaks

20:00 / 21:00



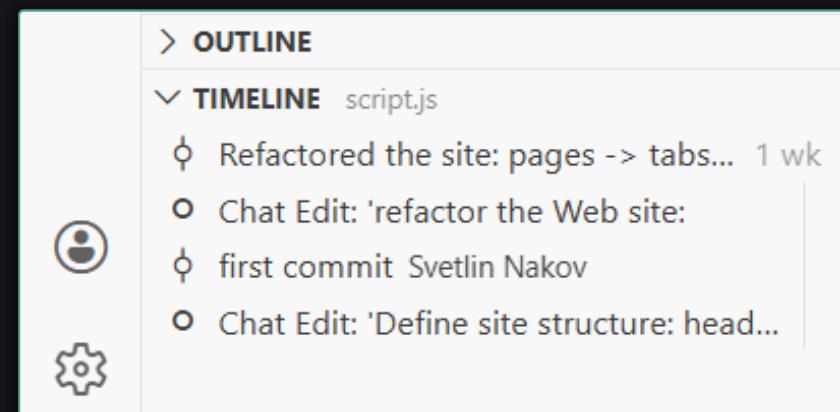
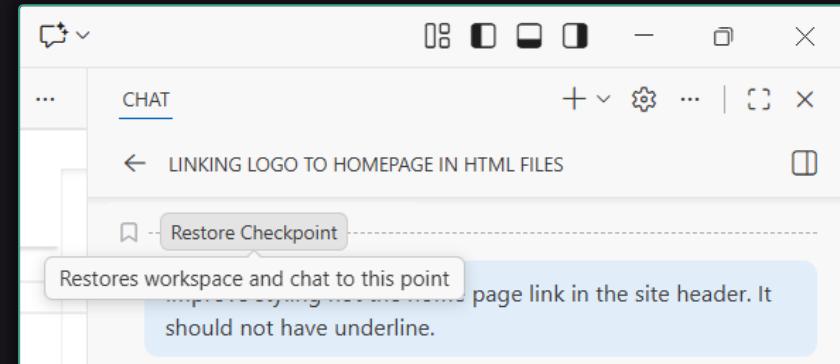
Git Concepts

Versioning, Repo, Clone, Commit, Push, Pull



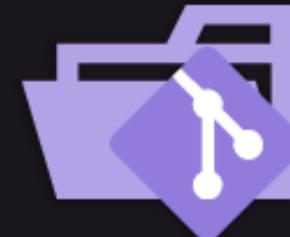
Checkpoints

- Why do we need "checkpoints" when using AI dev agents?
 - AI prompts can fail → **break all your codebase**
- **Checkpoints** allow you to **revert the changes** back to a stable version
- Types of checkpoints:
 - The "**Checkpoints**" in GitHub Copilot
 - The "**Timeline**" tab in VS Code
 - **Version control:** Git and GitHub



Git and Git Repos

- Git is like a "time machine"
 - Keeps track on changes in your project
 - Implements "checkpoints" (commits)
 - Allows reverting back to a checkpoint (an older version)
- Git holds projects in a repository (repo)
 - Git repo == project folder managed by Git
- Repos keep a "change history" of the project
 - A sequence of changesets (sets of changes files)



Git Concepts

- **Clone**: download a local working copy of Git repo
- **Commit**: create a "**checkpoint**"
- **Discard**: ignore pending changes
- **Push**: upload commits to the cloud
- **Pull**: download updates from the cloud
- **Sync**: Pull + Push
- Typical AI dev **workflow**:  discard
 - AI prompt → test → refine → commit → push



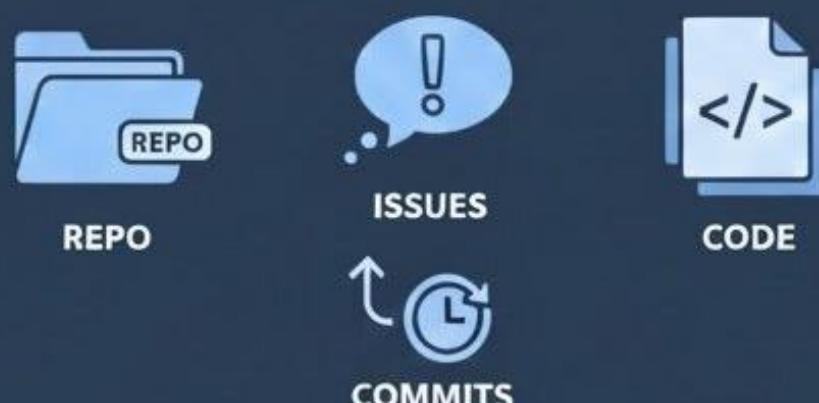
Demo: Git Commit History

[https://github.com/nakov/Company-
Web-Site/commits/](https://github.com/nakov/Company-Web-Site/commits/)

Intro to GitHub

Repos, Issues, Code, Commits

Intro to GitHub



The GitHub interface features four main icons: a folder labeled "REPO", a speech bubble with an exclamation mark labeled "ISSUES", a code editor labeled "CODE", and a circular arrow labeled "COMMITS".



A cartoon illustration of a blue cat sitting behind a computer monitor. The monitor displays icons related to GitHub: a message bubble, a code editor, a user profile, and a branching line. The background is light blue with faint white circles.

What is GitHub?

- GitHub is the world's #1 source code hosting site (cloud)
 - Free for open-source projects / personal projects
 - Paid plans for advanced functionalities
- GitHub provides:
 - Git source code repository
 - Issue tracker (bug tracker)
 - Project board (Kanban style)
 - Wiki pages (documentation)
 - Pull requests (branch merges + code reviews)
 - Build system (actions)
 - Site hosting (pages)
 - Discussions (forum)



Demo: Repos, Issues, Code, Commits

[https://github.com/tailwindlabs/
tailwindcss](https://github.com/tailwindlabs/tailwindcss)

Registration in GitHub

You're one step away from GitHub Copilot in VS Code

To get started, create a free GitHub account and start coding with AI today.

Here's what GitHub Copilot can help you do:

- ✓ Write code faster and smarter with AI-powered suggestions ranging from single lines to entire functions
- ✓ Use natural language chat and agent mode to ask questions, get explanations, and complete complex, multi-step development tasks

With Copilot Free, you also get:

- ✓ 50 chat or agent mode requests per month
- ✓ 2,000 code completions per month
- ✓ Access to Claude 3.5 Sonnet, GPT-4.1, and more

GitHub Copilot Free, Pro and Pro+ may show [public code](#) suggestions and we may use your data for product improvement. You can change these [settings](#) at any time.



Already have an account? [Sign in](#)

Sign up for GitHub

[Continue with Google](#)

or

Email*
steve-nak@nakov.com

Password*
.....

Password should be at least 15 characters OR at least 8 characters including a number and a lowercase letter.

Username*
steve-nak

Username may only contain alphanumeric characters or single hyphens, and cannot begin or end with a hyphen.

Your Country/Region*
Bulgaria

For compliance reasons, we're required to collect country information to send you occasional updates and announcements.

steve-nak

github.com/steve-nak

steve-nak

Overview Repositories 1 Projects ...

steve-nak

Set status Edit profile

Popular repositories Customize your pins

Tetris-Game-JS Public

JavaScript based in-browser Tetris game implementation

Creating a GitHub Repository

New repository

github.com/new

New repository

Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#). Required fields are marked with an asterisk (*).

1 General

Owner * steve-nak / Repository name * Tetris-Game-JS Tetris-Game-JS is available.

Great repository names are short and memorable. How about [verbose-fiesta](#)?

Description

JavaScript based in-browser Tetris game implementation

54 / 350 characters

2 Configuration

Choose visibility * Public

Choose who can see and commit to this repository

steve-nak/Tetris-Game-JS: JavaS x +

github.com/steve-nak/Tetris-Game-JS

steve-nak / Tetris-Game-JS

Code Issues Pull requests Actions Projects Wiki Security

Tetris-Game-JS Public

Pin Watch 0 Fork 0 Star 0

main Go to file + Cloud Code

steve-nak Initial commit 4c9155c · now

README.md Initial commit now

README

Tetris-Game-JS

JavaScript based in-browser Tetris game implementation

About

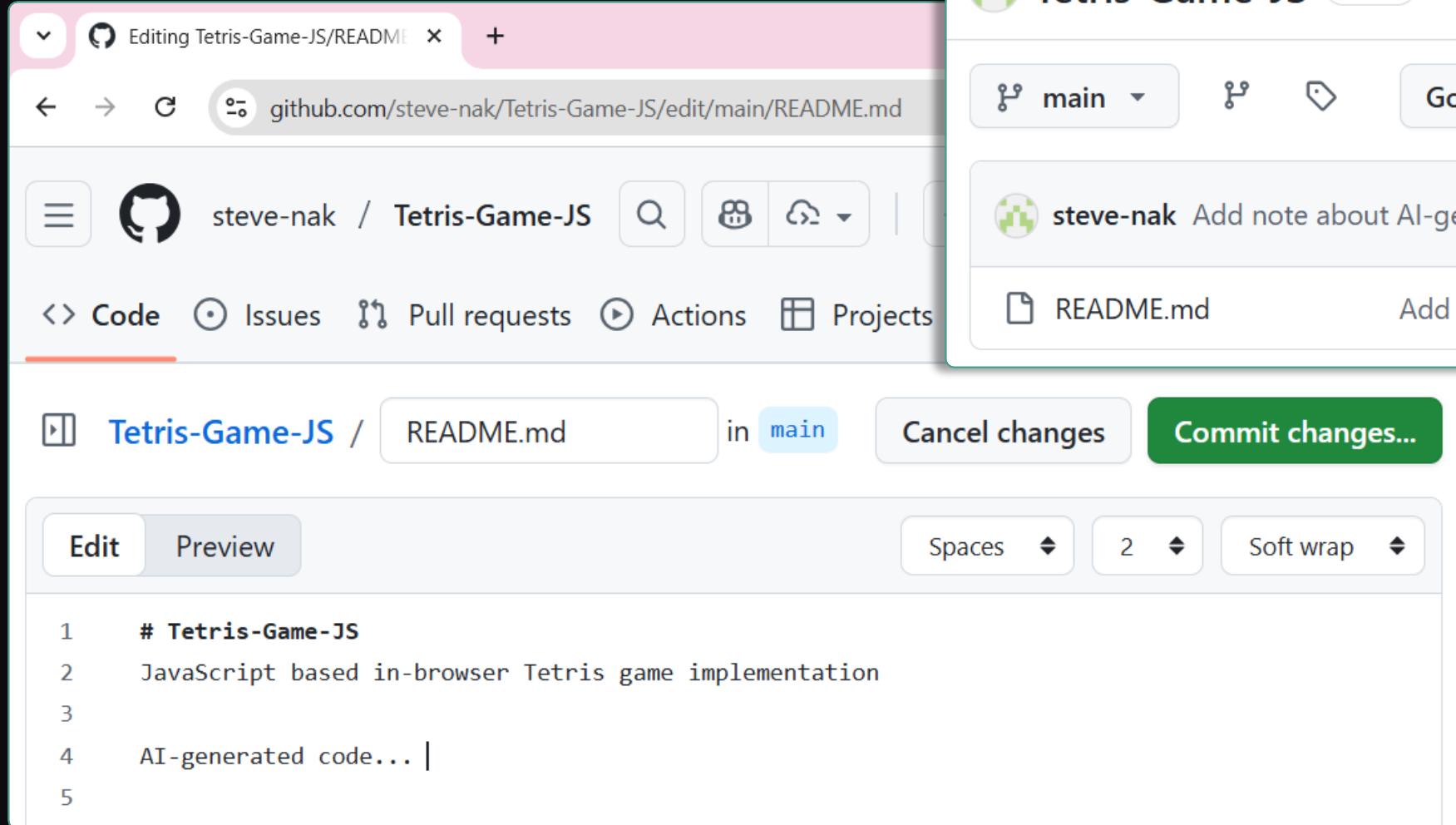
JavaScript based in-browser Tetris game implementation

Readme Activity 0 stars 0 watching 0 forks

Releases

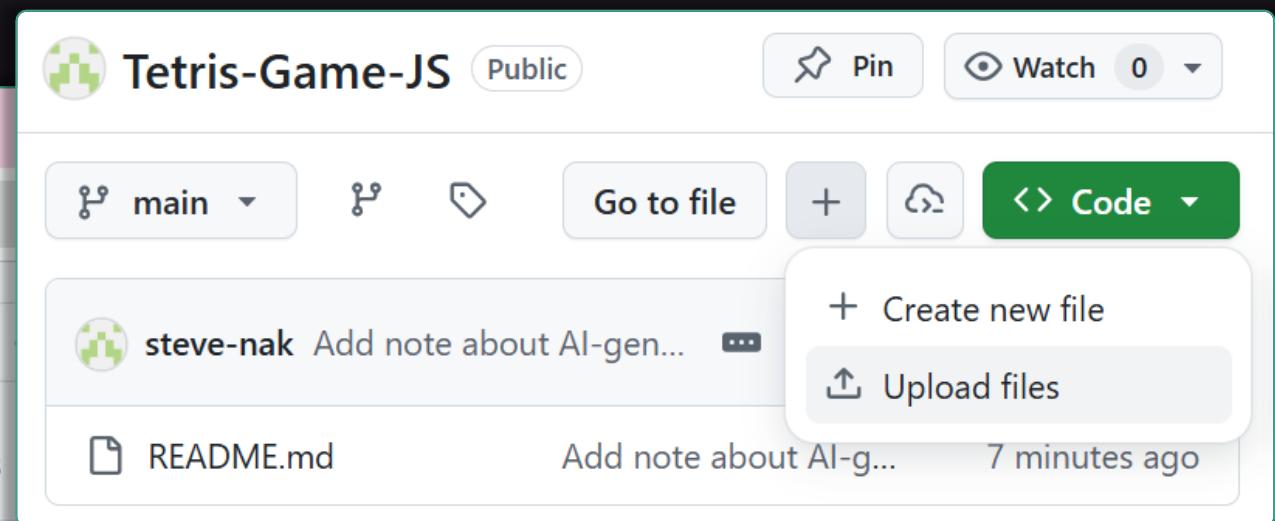
No releases published Create a new release

Uploading / Editing Files Online



The screenshot shows the GitHub interface for editing the `README.md` file of the `Tetris-Game-JS` repository. The repository is public and owned by `steve-nak`. The main branch is `main`. The file content includes a header, a description, and a placeholder for AI-generated code.

```
1 # Tetris-Game-JS
2 JavaScript based in-browser Tetris game implementation
3
4 AI-generated code... |
```

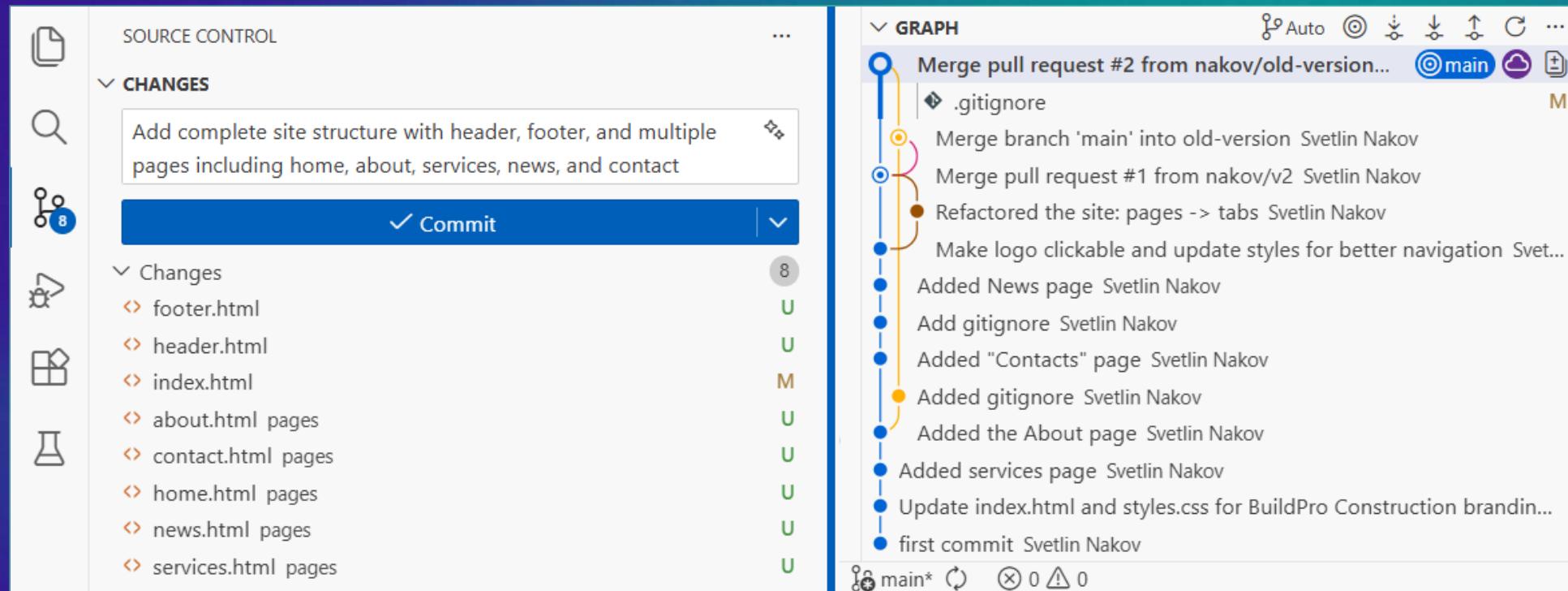


The screenshot shows the GitHub commit modal for the `README.md` file. A note has been added: `Add note about AI-gen...`. The modal also includes options to create a new file or upload files.

+ Create new file
Upload files

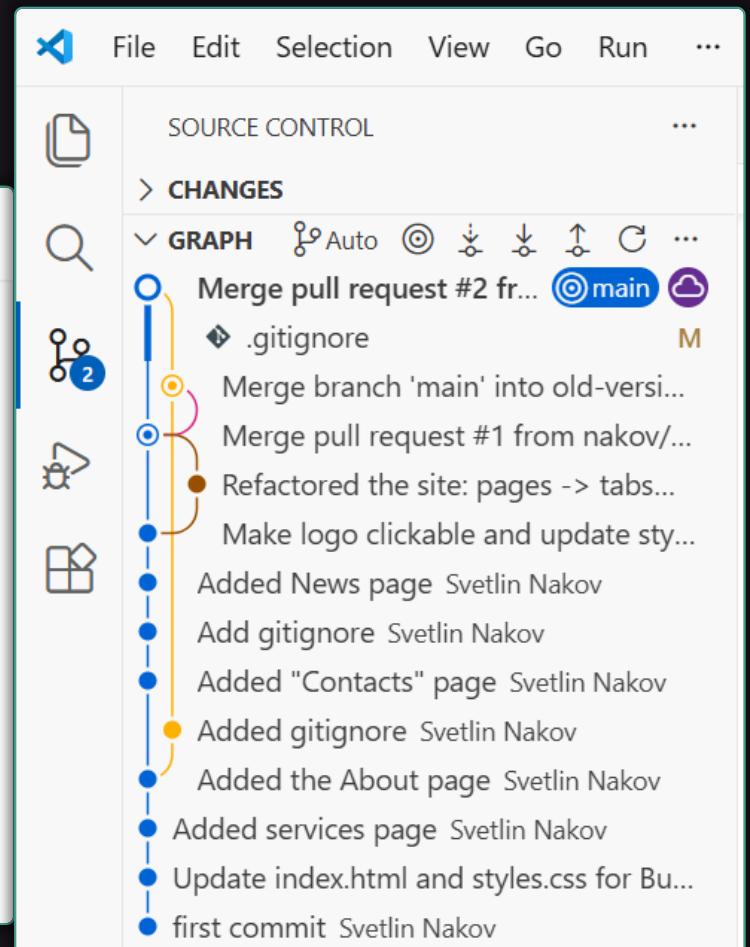
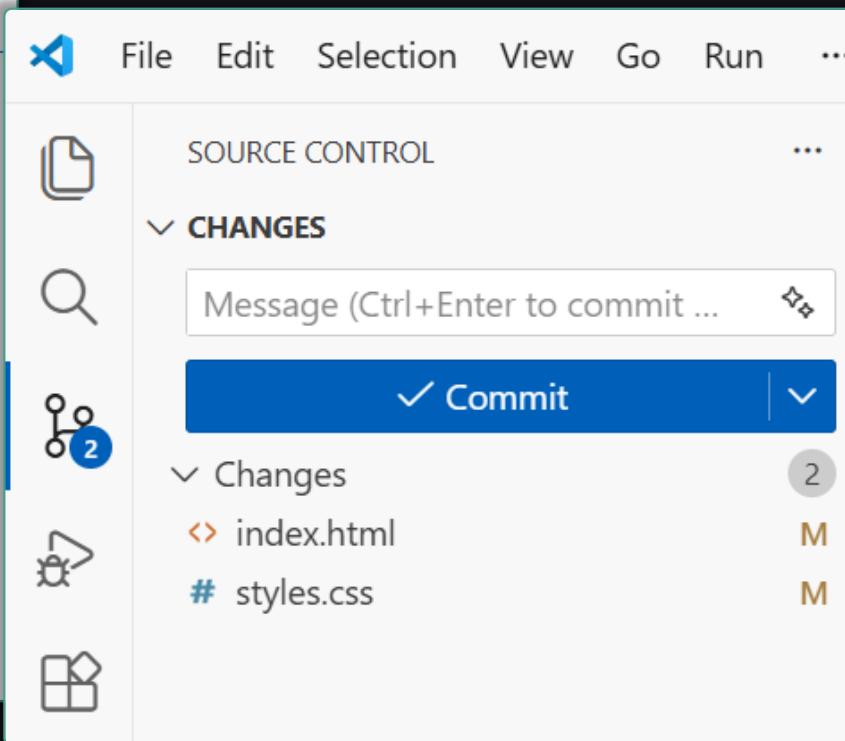
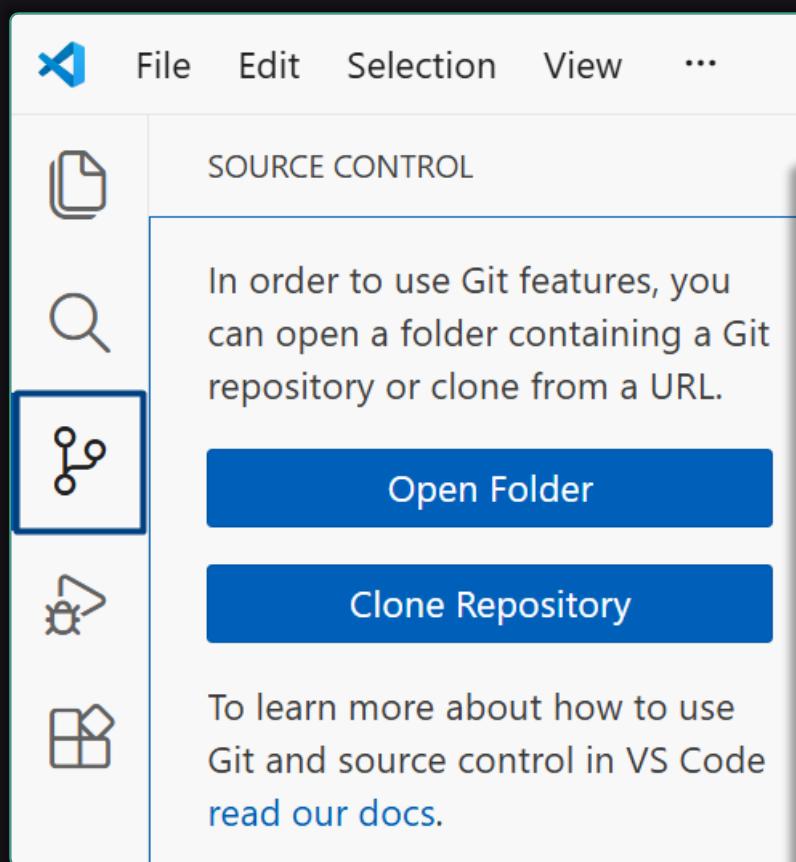
Git in Visual Studio Code

Publish a Project to GitHub, View Changes, Commit, Push, Discard Changes

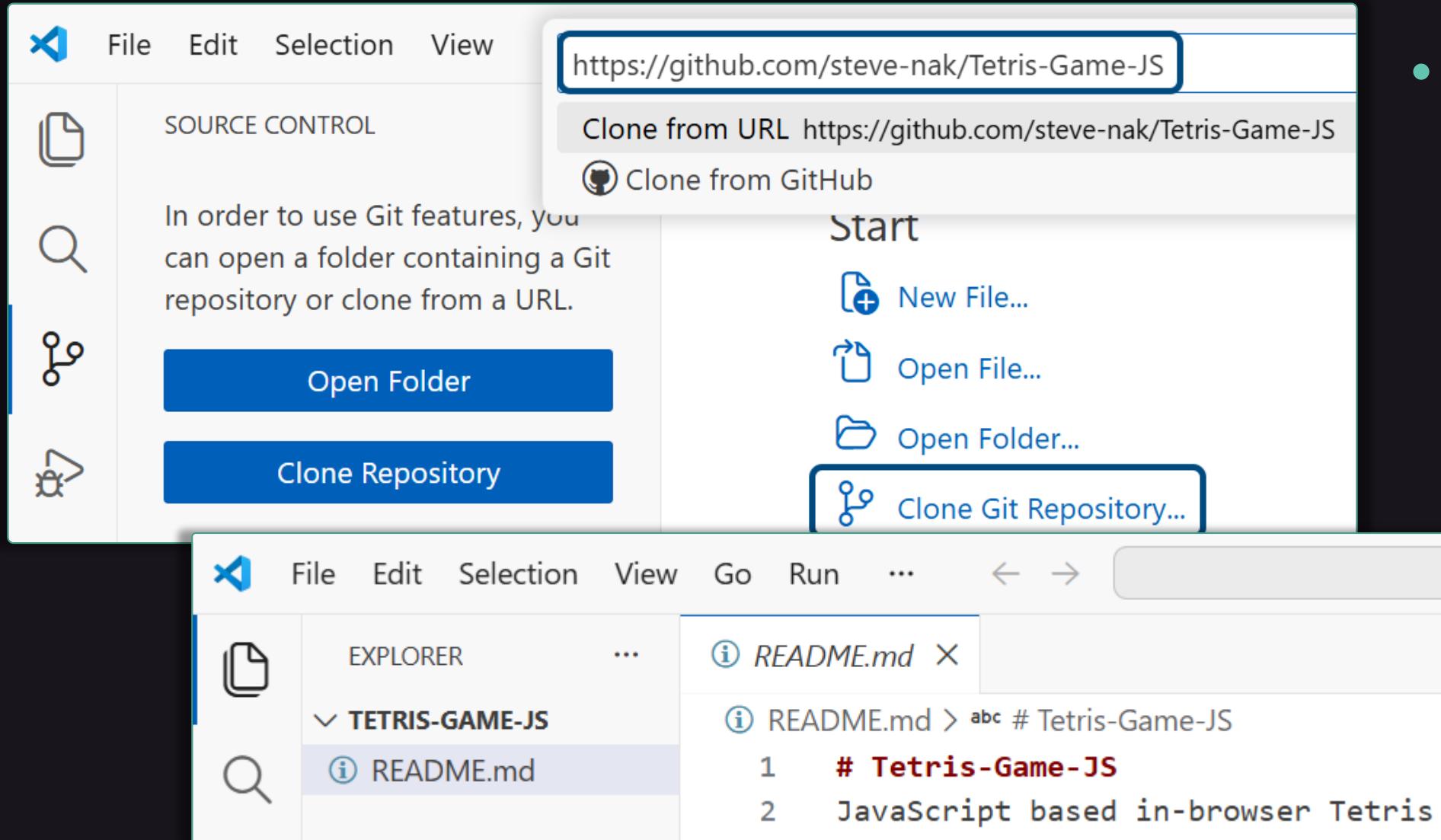


Git in VS Code

- Visual Studio Code has built-in Git support



Cloning Existing Project from GitHub



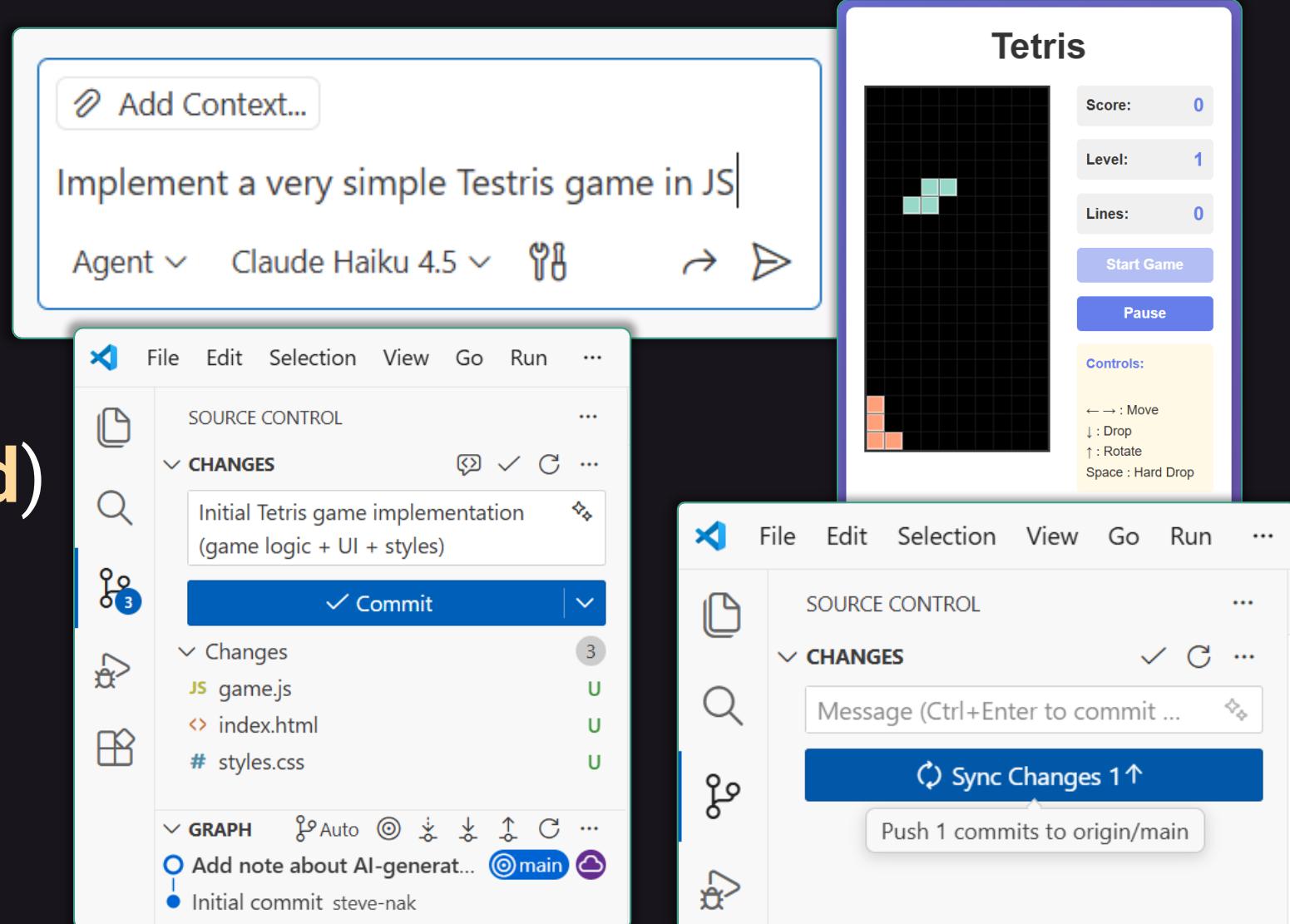
The screenshot shows a code editor interface with a dark theme. On the left, there are several icons: a file icon, a magnifying glass icon, a network icon, and a play icon. The main menu bar includes File, Edit, Selection, View, Go, Run, and an ellipsis. A sub-menu is open under the File menu, containing options like Clone from URL (with a URL of https://github.com/steve-nak/Tetris-Game-JS), Clone from GitHub, Start, New File..., Open File..., Open Folder..., and Clone Git Repository... (which is highlighted with a blue rectangle). Below the menu, a sidebar titled "SOURCE CONTROL" contains the message: "In order to use Git features, you can open a folder containing a Git repository or clone from a URL." It has two buttons: "Open Folder" and "Clone Repository". The bottom half of the screen shows the "EXPLORER" panel with a tree view. Under the "TETRIS-GAME-JS" folder, there is a file named "README.md" (also highlighted with a blue rectangle). The content of this file is displayed in the main editor area:

```
1 # Tetris-Game-JS
2 JavaScript based in-browser Tetris
```

- **Clone ==**
download a local copy
from GitHub

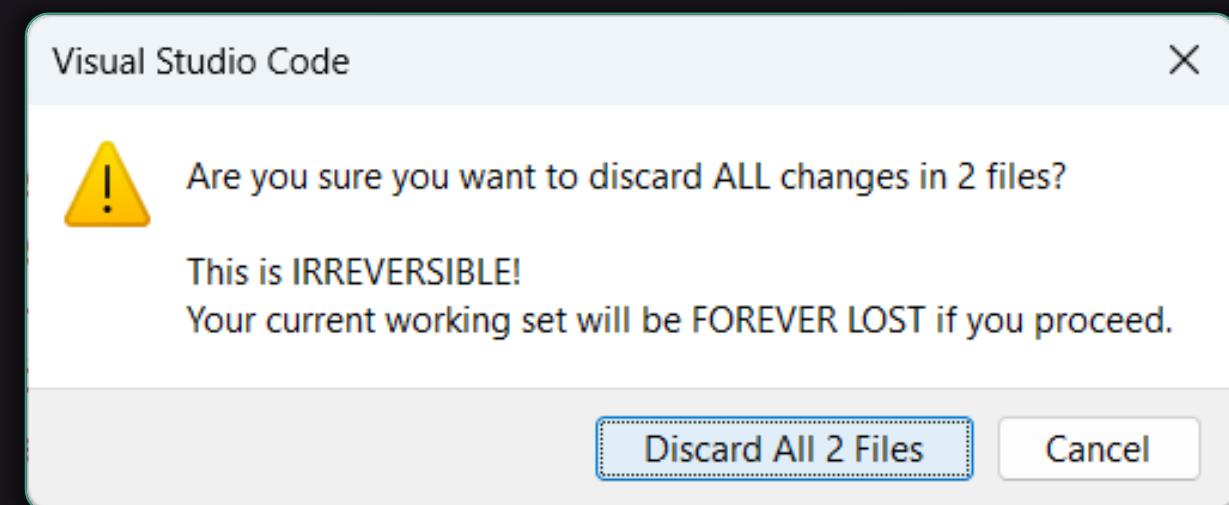
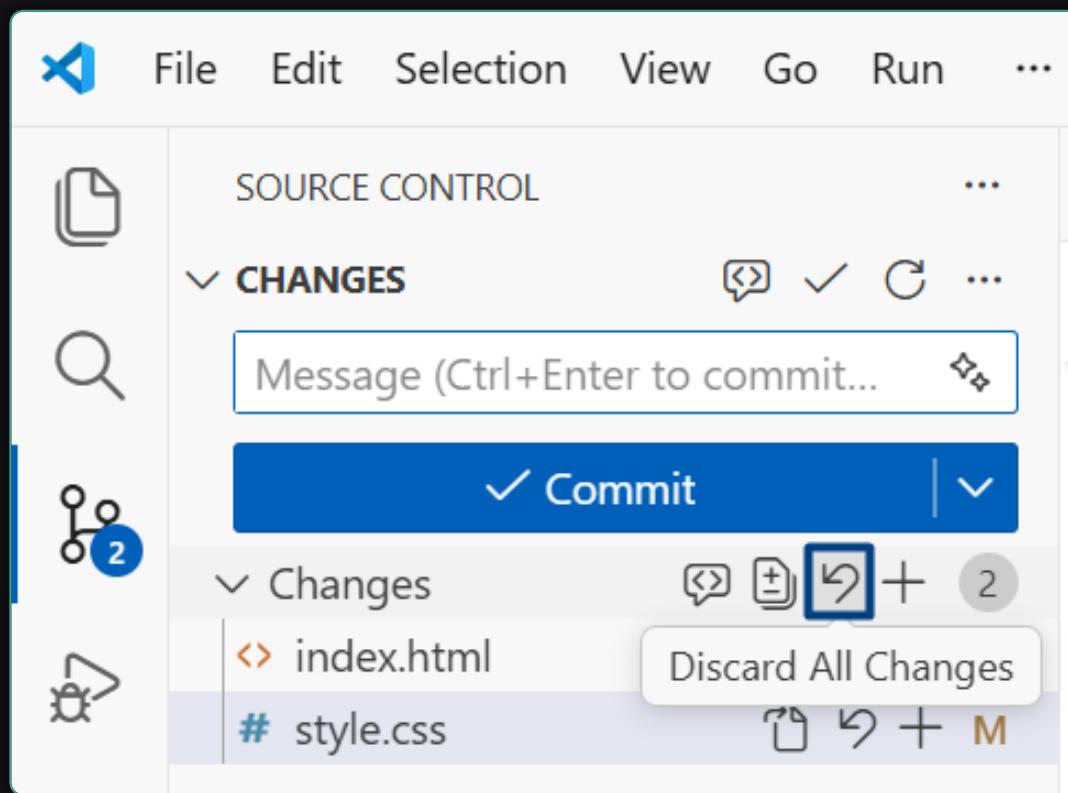
Modify → Commit → Push

- **Modify** the code (with AI prompt)
- **Commit** pending changes (or **discard**)
- **Push** commits to GitHub



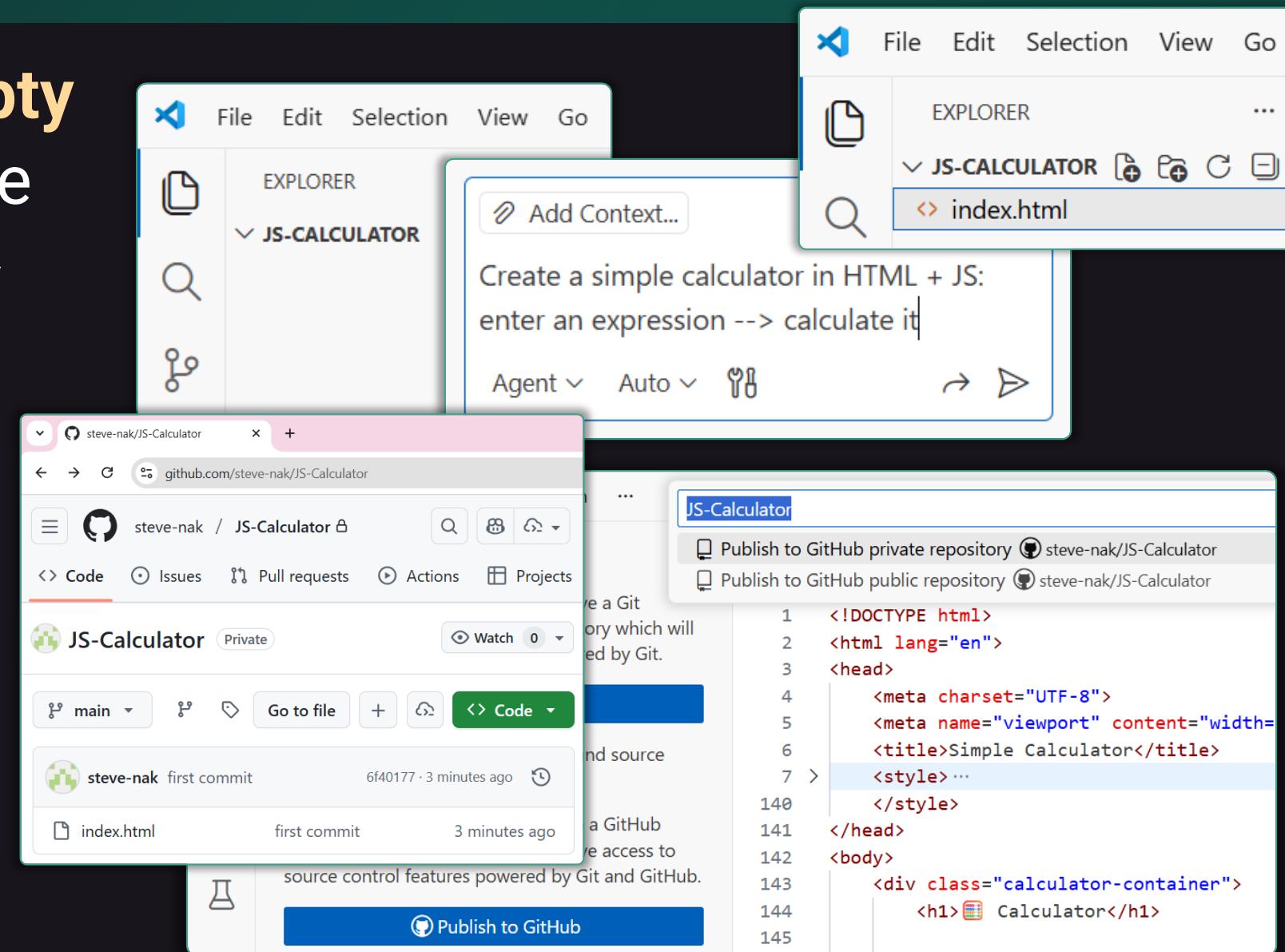
Modify → Discard

- Sometimes code changes are incorrect / unwanted
 - We may want to **discard uncommitted changes**



Publishing a Repo to GitHub

- Create a **new empty project** in VS Code
- Write / generate / **add** some code
- **Publish** a project to **GitHub**
 - Note: cannot publish empty projects



Exercise: Publish, Delete, Clone

1. Create a **new project** in VS Code
 - VS Code → Open Folder → New Folder
2. Generate / **add some code** / files
 - Run an AI prompt in GitHub Copilot
3. **Publish to GitHub**
 - VS Code → Source Control → Publish to GitHub
4. **Delete** the local working folder
 - Exit VS Code; File Explorer → Delete Folder
5. **Clone** the project again from GitHub
 - VS Code → Clone Git Repository

Break

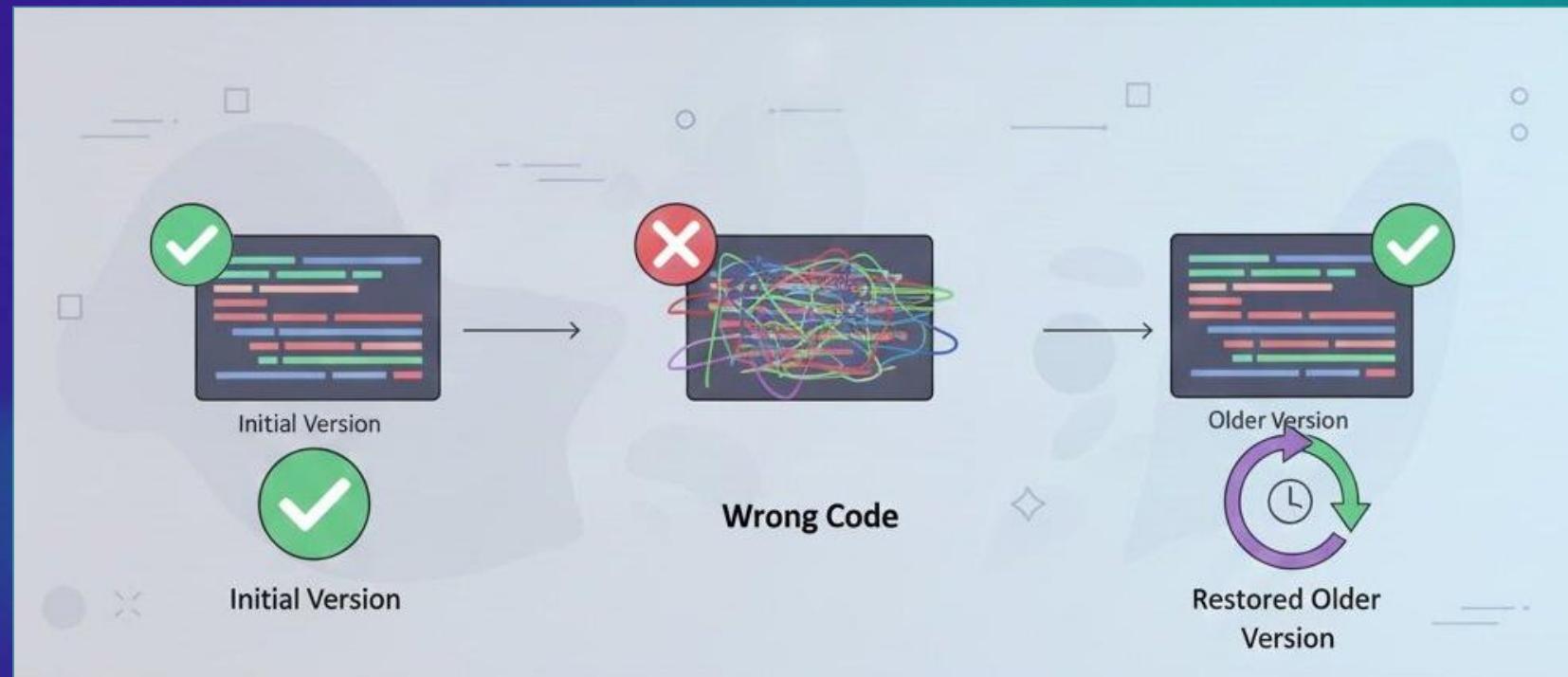
(10 minutes)

Start timer



Recovering from Mistakes

Checkpoints, Discard Changes, Undo Commit,
Re-Clone a Repo, Revert to Old Version



How to Recover from Mistakes?

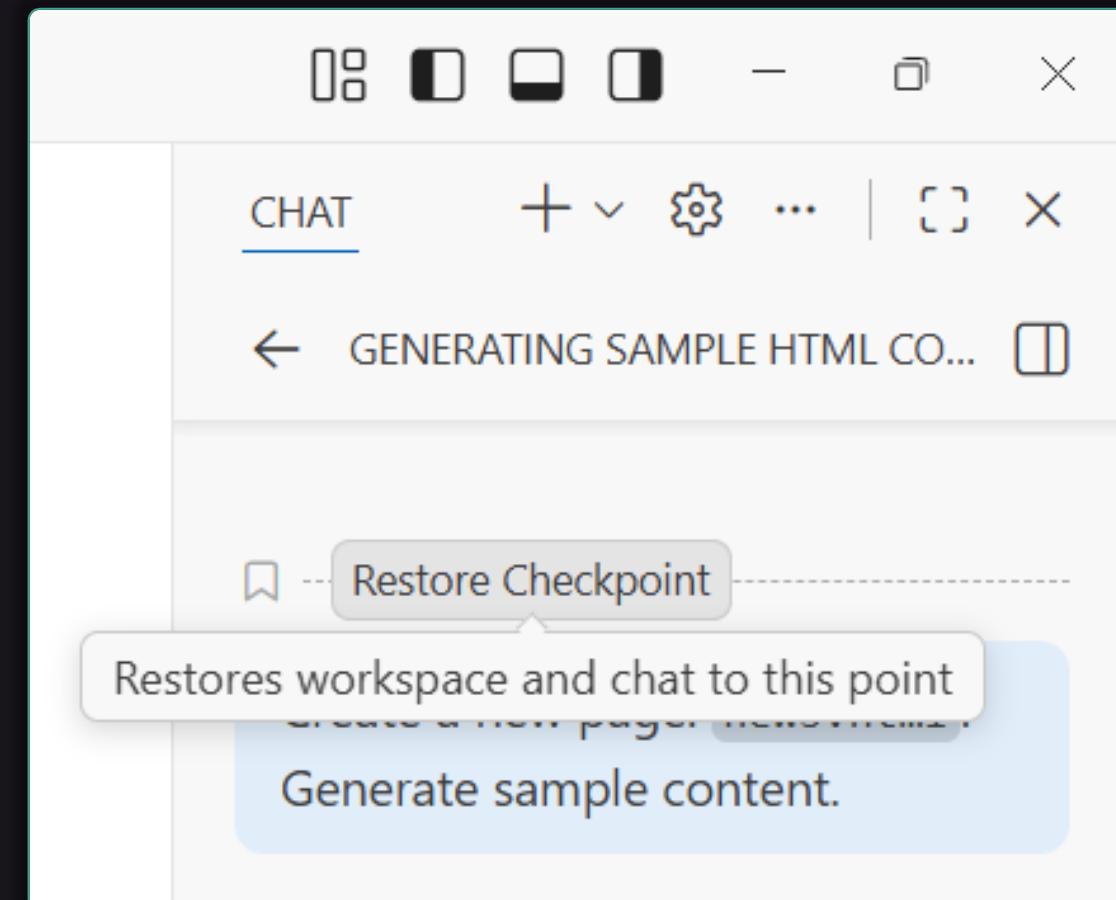


- Recovering from mistakes during development:
 1. **Restore a checkpoint** in the Copilot chat window
 - Works in the current chat session only
 2. **Discard changes** from the "Source Control" window
 - Works when the code is not committed
 3. **Undo last commit** → when wrong code is just committed
 4. **Re-clone from GitHub** → take the last pushed version
 5. **Download old version from GitHub** → go back in time

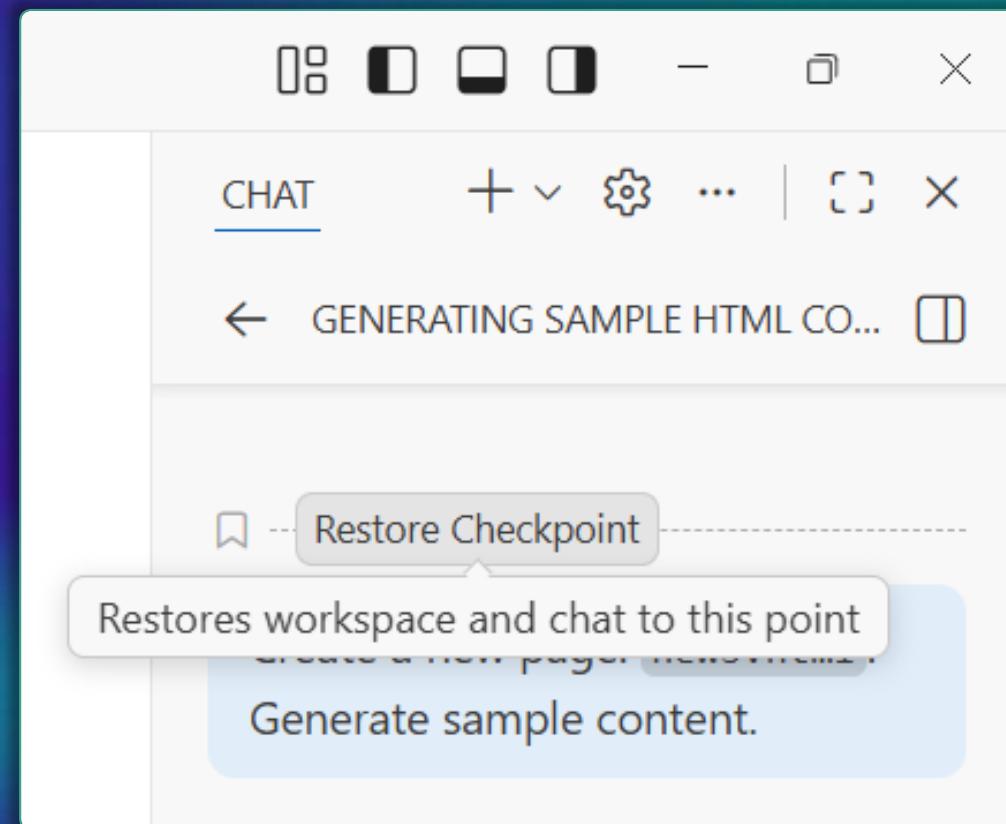
Restore a Checkpoint from the Chat



- **Restore a checkpoint** from the Copilot chat window
 - Works in the **current chat session** only!
 - Does not maintain full change history

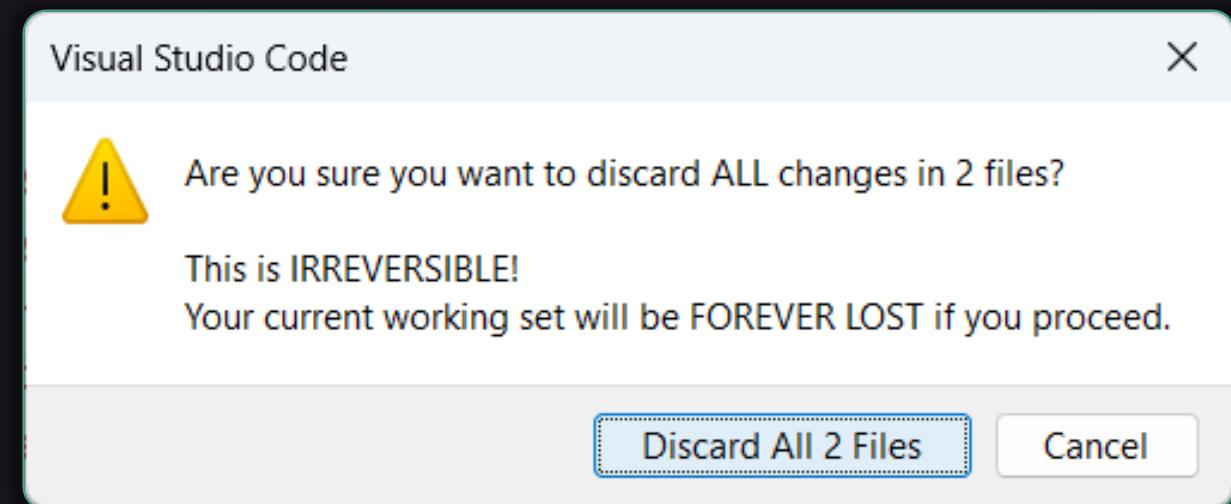
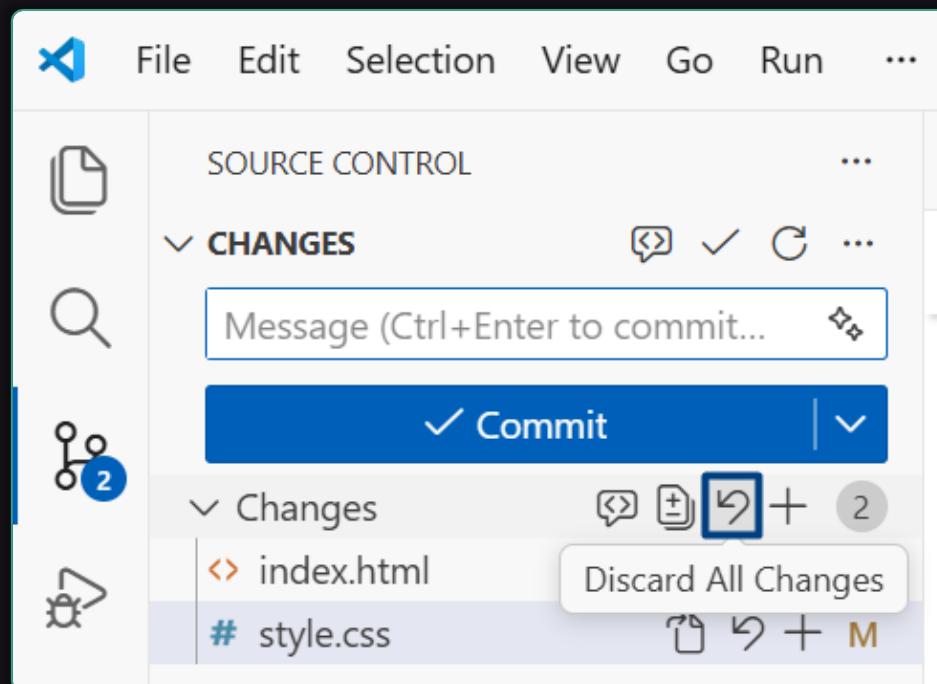


Restore a Chat Checkpoint: Live Demo



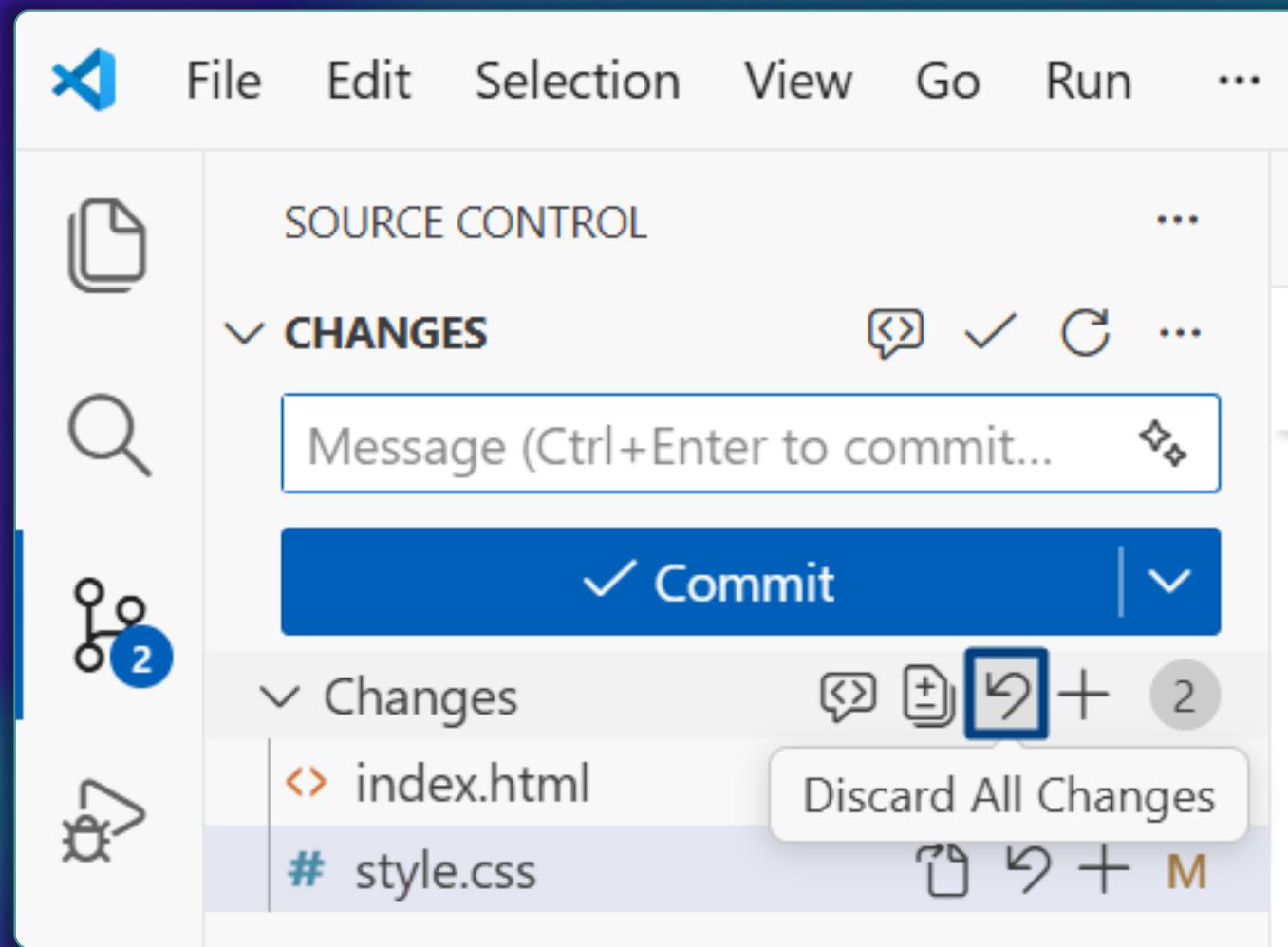
Git → Discard Changes

- **Discard changes** from "Source Control" in VS Code
 - Works only when the code is still **not committed**



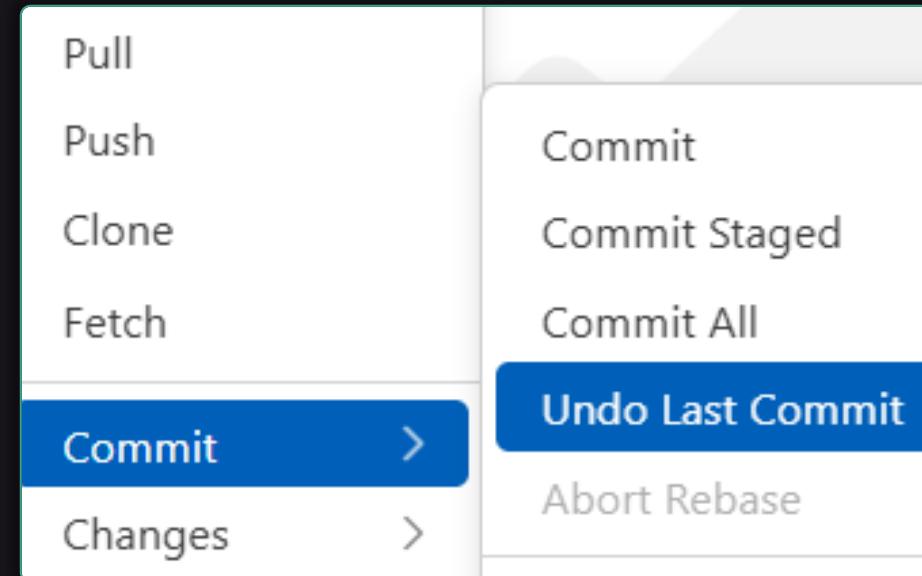
- Important rule: **don't commit broken code** → test before

Discard Changes: Live Demo

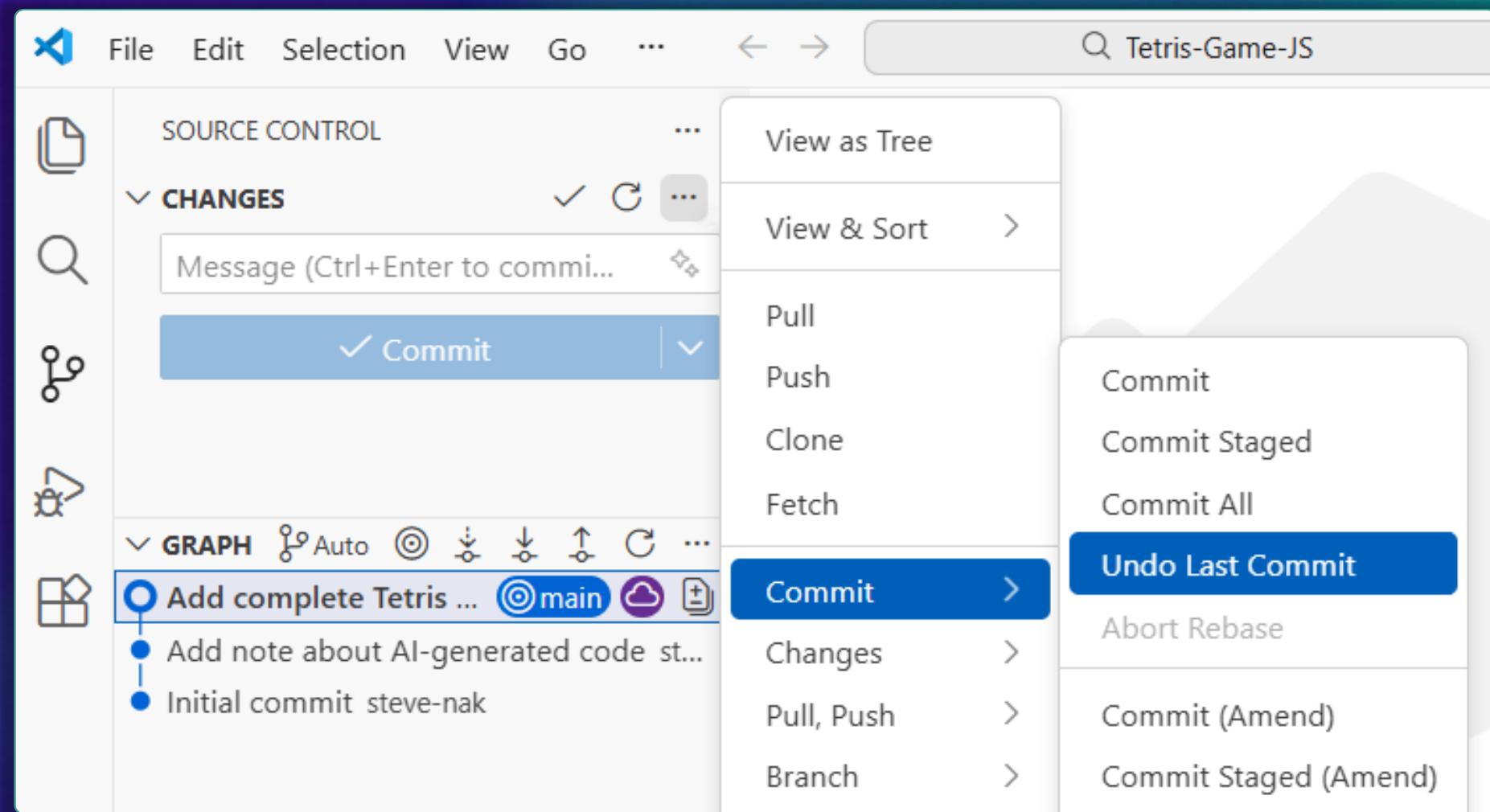


Undo Last Commit

- Committed wrong code (and pushed to GitHub)?
 - How to go back?
- "Undo Last Commit" in VS Code
 - VS Code → Source Control → Changes → Commit → Undo Last Commit
 - Goes back to the moment before the code is committed
 - Edit and fix the code → commit it again → push again



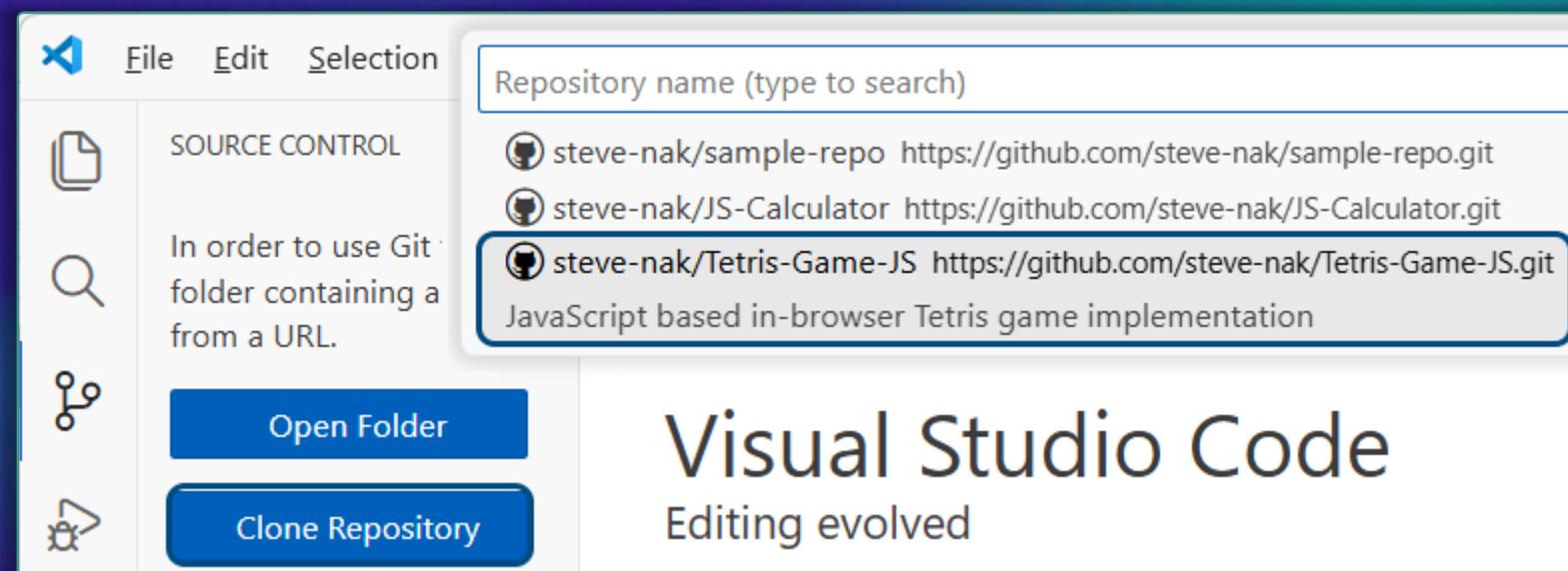
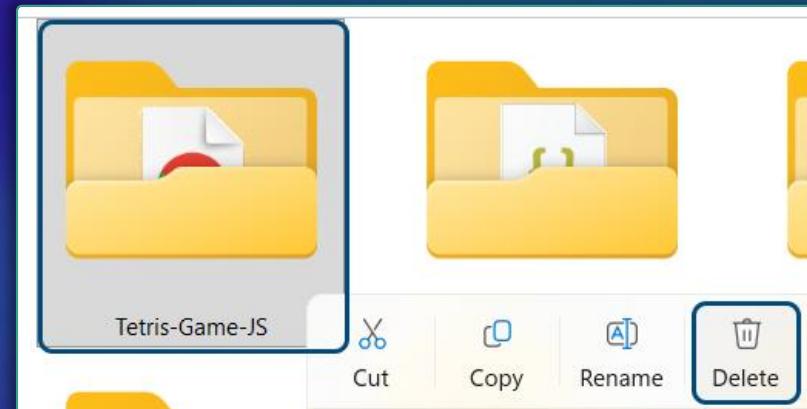
Undo Last Commit: Live Demo



Re-Clone from GitHub

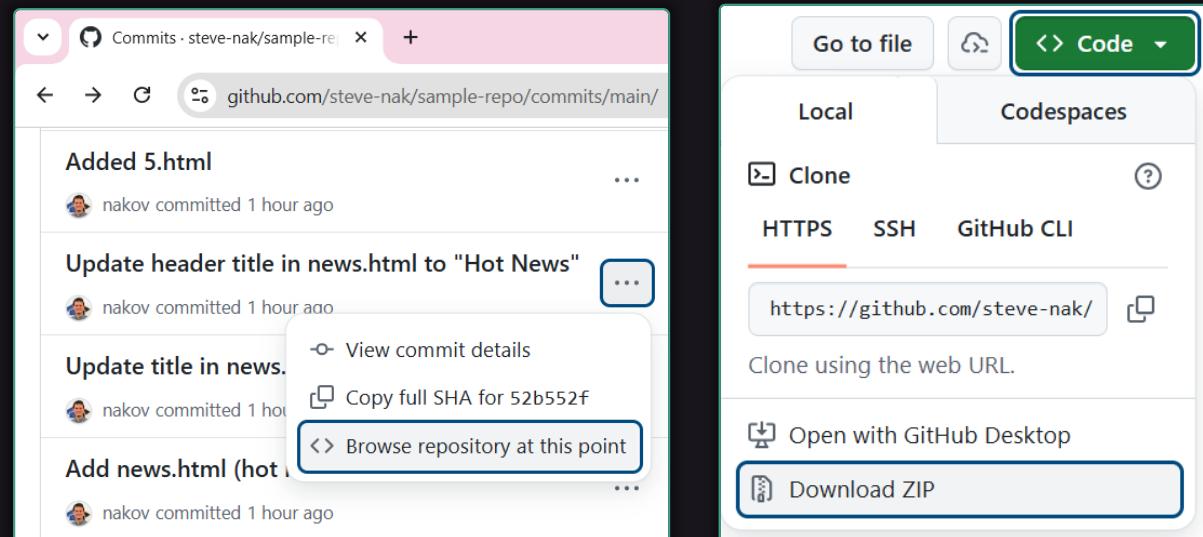
- My **local Git repo is broken** → my computer is a mess
 - I want to **reset everything** to match GitHub
 - How to **reset the repo** to its GitHub state?
- **Re-clone** the project **from GitHub**:
 1. **Exit** VS Code
 2. **Delete** the project folder (in File Explorer)
 3. **Clone** the GitHub repo again from VS Code
- Now your working repo is an exact copy of GitHub

Re-Clone from GitHub: Live Demo



Revert to Old Version from GitHub

- Sometimes the project can run in **wrong direction**
 - How **to go back in time**, e. g. few hours / few day ago?
- **Reverting to an old version** from GitHub:
 1. Find the correct version in the **Commit History**
 2. **Browse** repo at this point
 3. **Download** repo as **ZIP**
 4. **Unzip** in your local folder
 5. **Test → commit → push**



Revert Old Version from GitHub

Live Demo

A screenshot of a web browser displaying a GitHub commit history for a repository named "steve-nak/sample-repo". The commits are listed in reverse chronological order:

- Added 5.html** by **nakov** committed 48 minutes ago. The commit hash is **83e3c24**.
- Update header title in news.html to "Hot News"** by **nakov** committed 50 minutes ago. The commit hash is **52b552f**.

At the bottom of the commit list, there is a button labeled **Browse repository at this point**. A blue arrow points from this button to the commit hash **52b552f**, indicating that clicking it will take the user to the repository's state at that specific commit.

A screenshot of a web browser displaying a GitHub repository page for the commit hash **52b552f**. The URL in the address bar is github.com/steve-nak/sample-repo/tree/52b552f08b7b0679628cb4e7090c912f6a4cae48. The repository contains the following files:

- 1.html
- 2.html
- 3.html
- 4.html
- news.html

The "Clone" section shows the repository's clone URL: <https://github.com/steve-nak/sample-repo.git>. There are also links to "Open with GitHub Desktop" and "Download ZIP".

Import / Export Projects

Export from Bolt / Lovable / Replit to GitHub



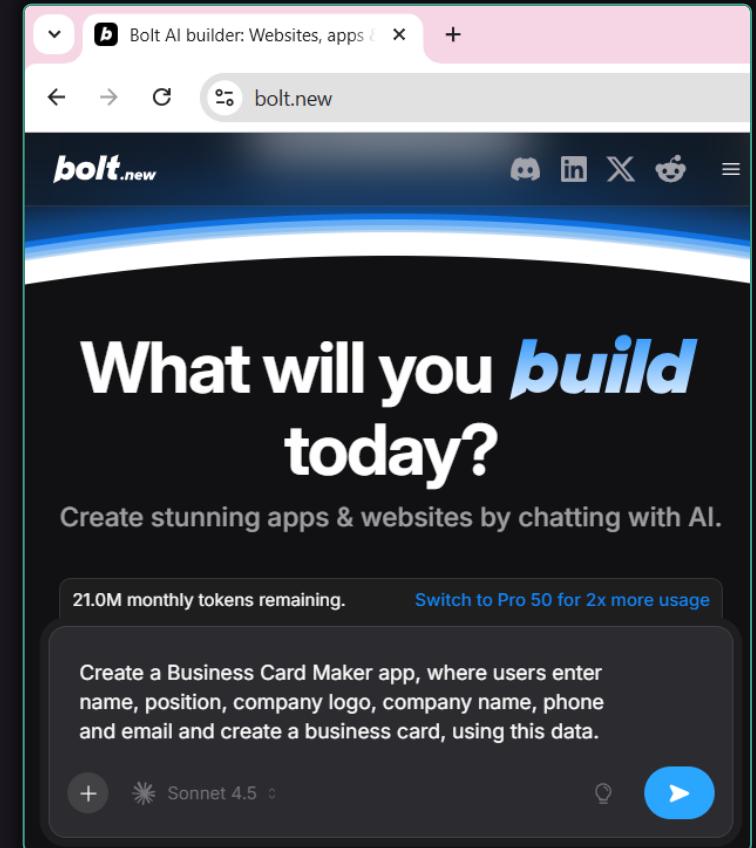
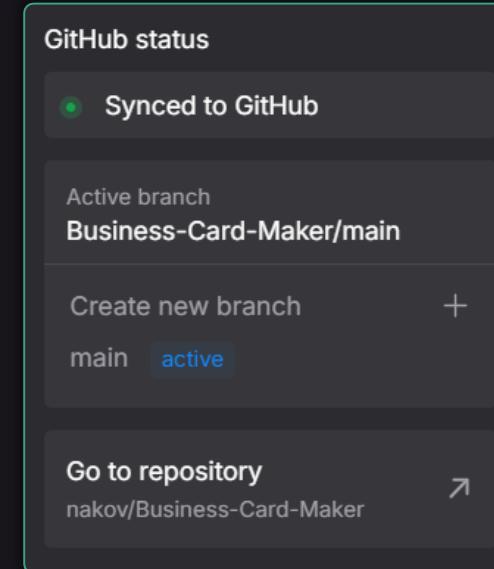
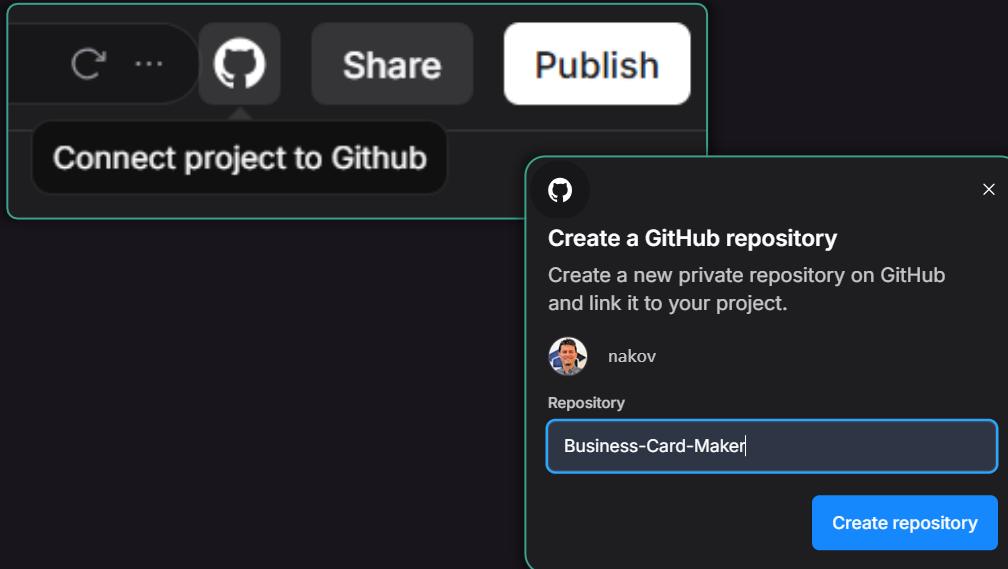
Export / Import Projects with GitHub

- Most vibe coding platforms allow **export / sync** the project with a **GitHub repo**
 - This unlocks moving projects between **Bolt**, **Lovable**, **Replit**, **Firebase Studio**, **v0** to **GitHub** and **VS Code + Copilot**

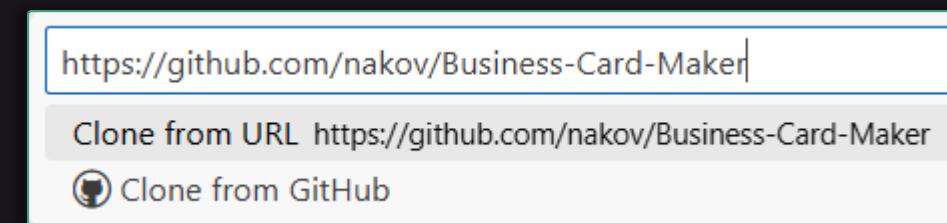


Example: Bolt → GitHub → VS Code

- AI generate a project with **Bolt**
- **Export** the project to **GitHub repo**

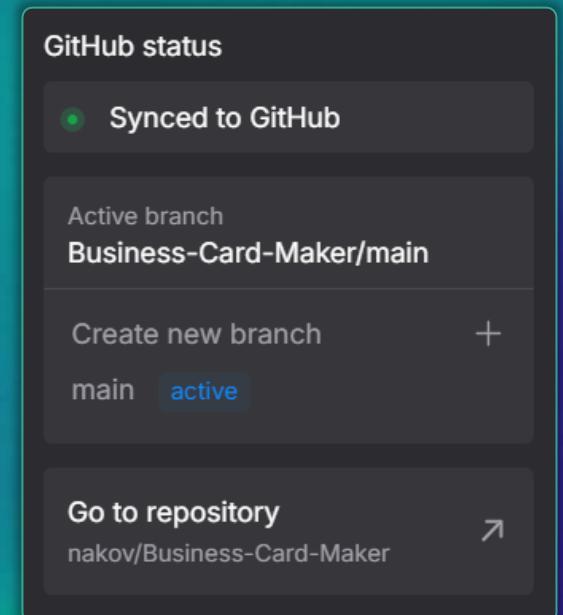
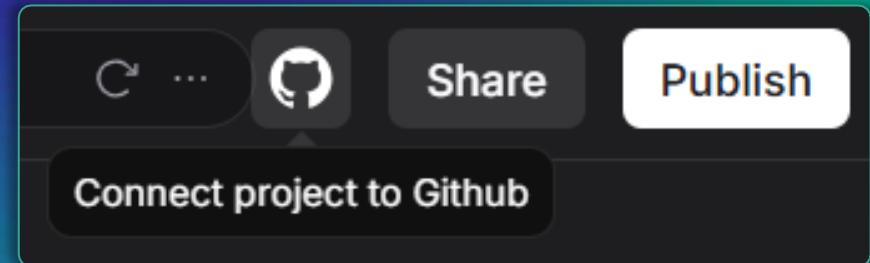
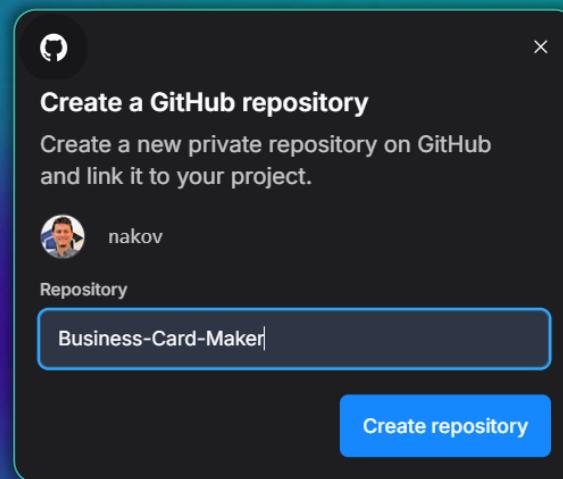
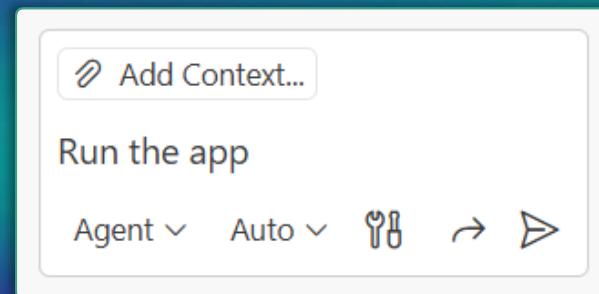
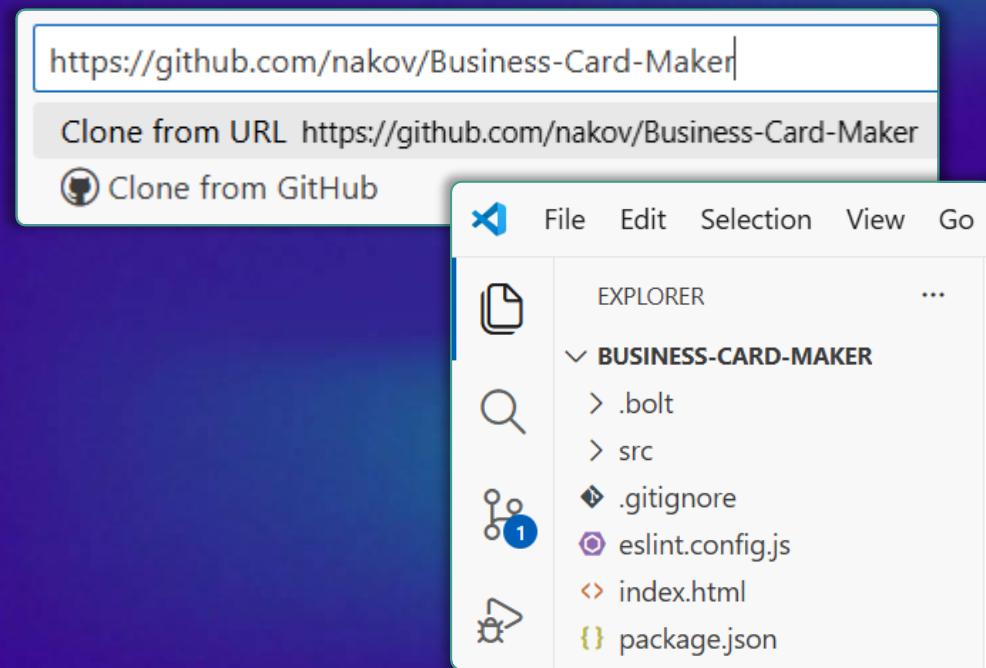


- **Clone** the project in **VS Code** and continue development



Export from Bolt to GitHub

Live Demo



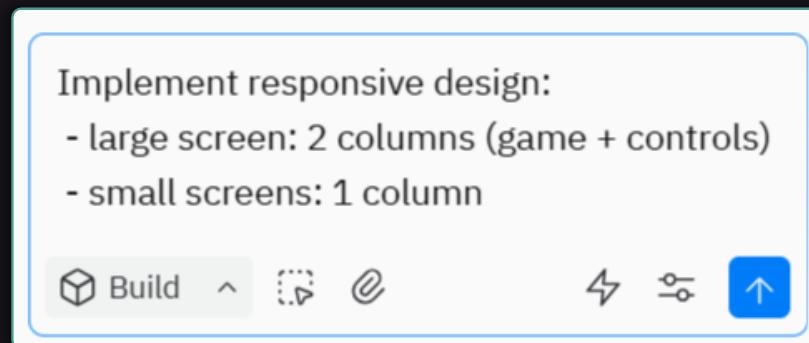
Example: GitHub → Replit



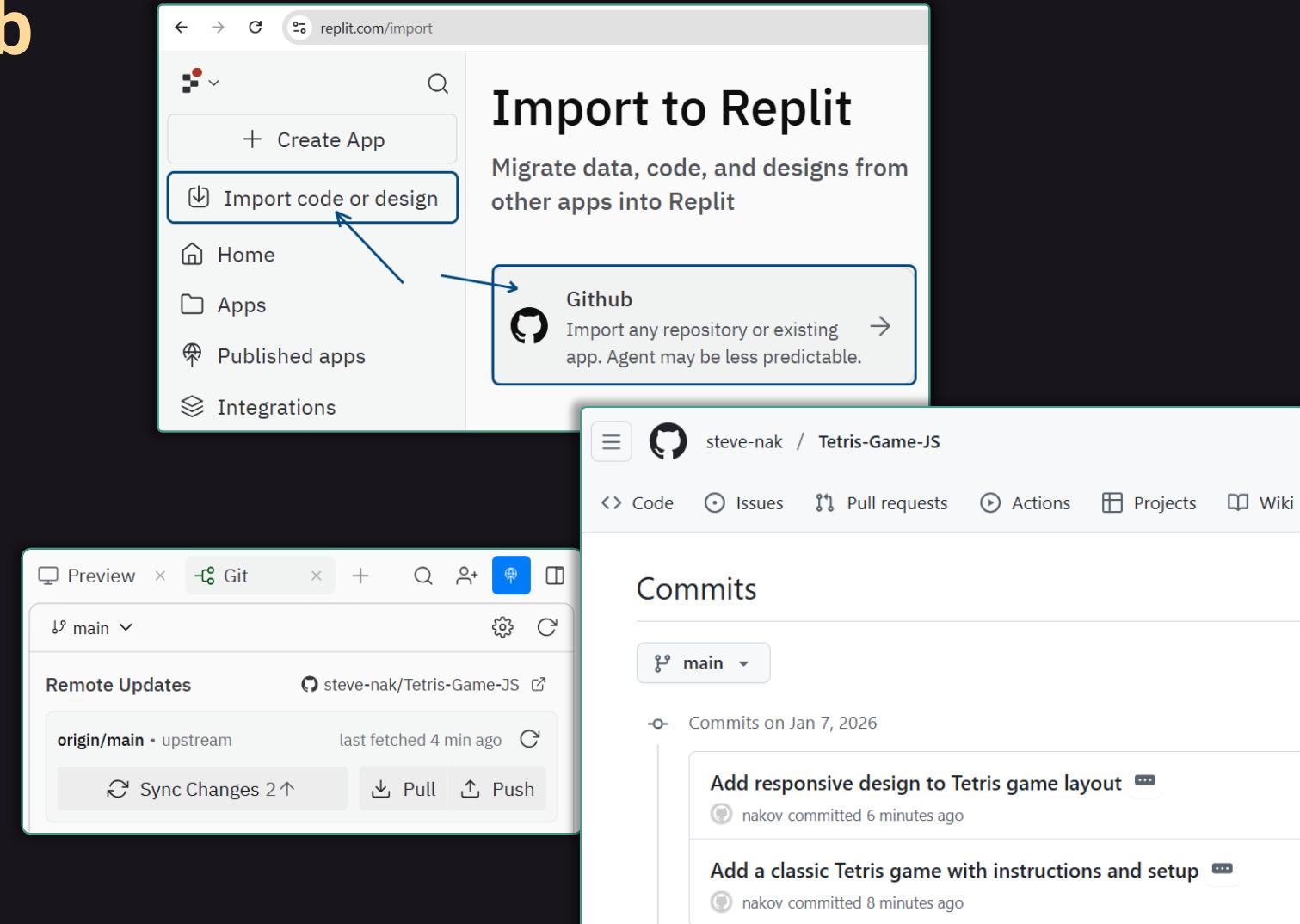
- Import exiting GitHub repo into Replit
 - Prompt Replit agent to modify the code

Implement responsive design:

- large screen: 2 columns (game + controls)
 - small screens: 1 column



- Push the changes to GitHub (Sync)

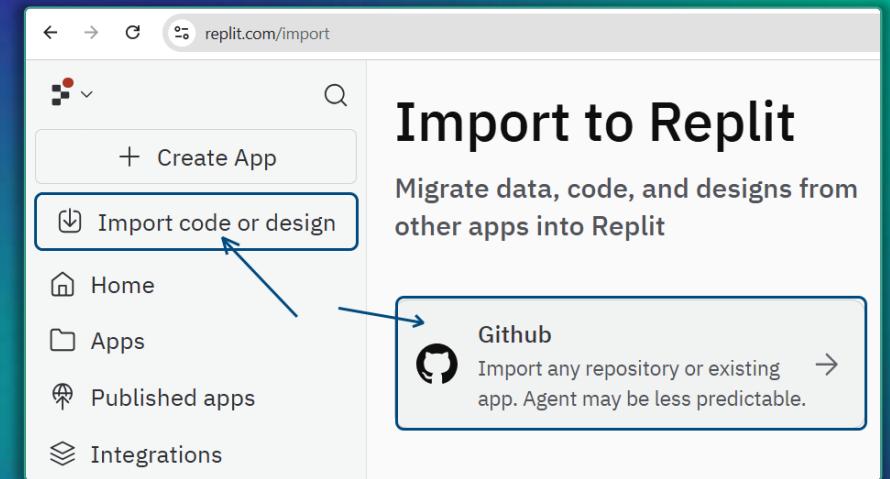
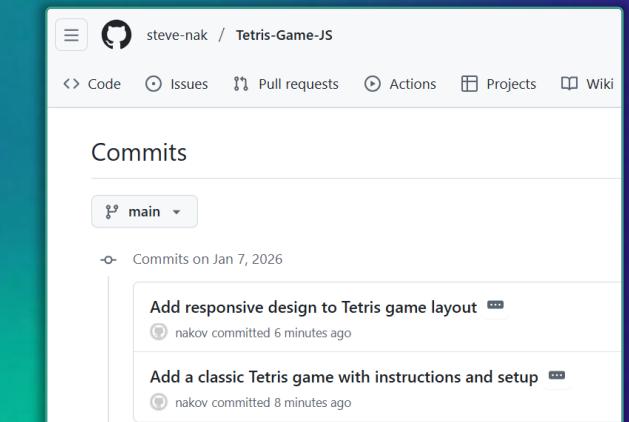
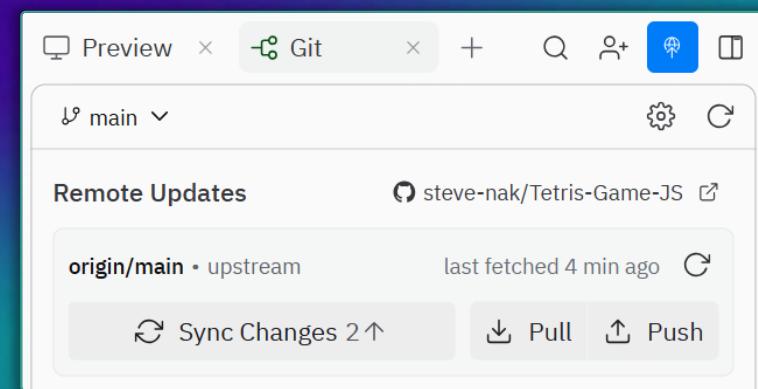
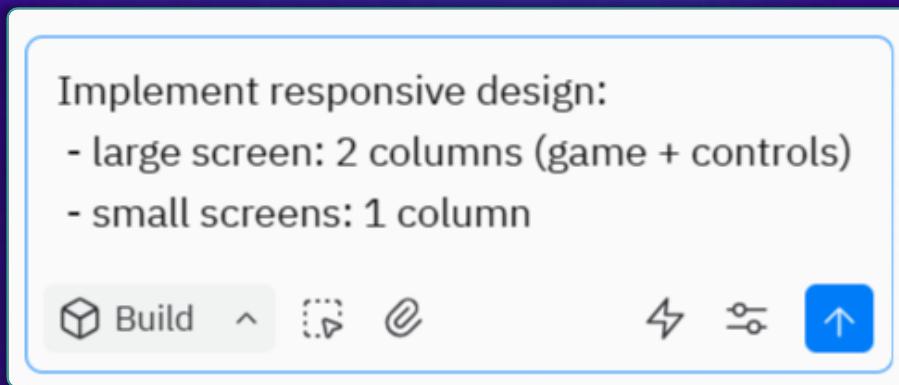


Import from GitHub to Replit

Live Demo

Implement responsive design:

- large screen: 2 columns (game + controls)
- small screens: 1 column



Break

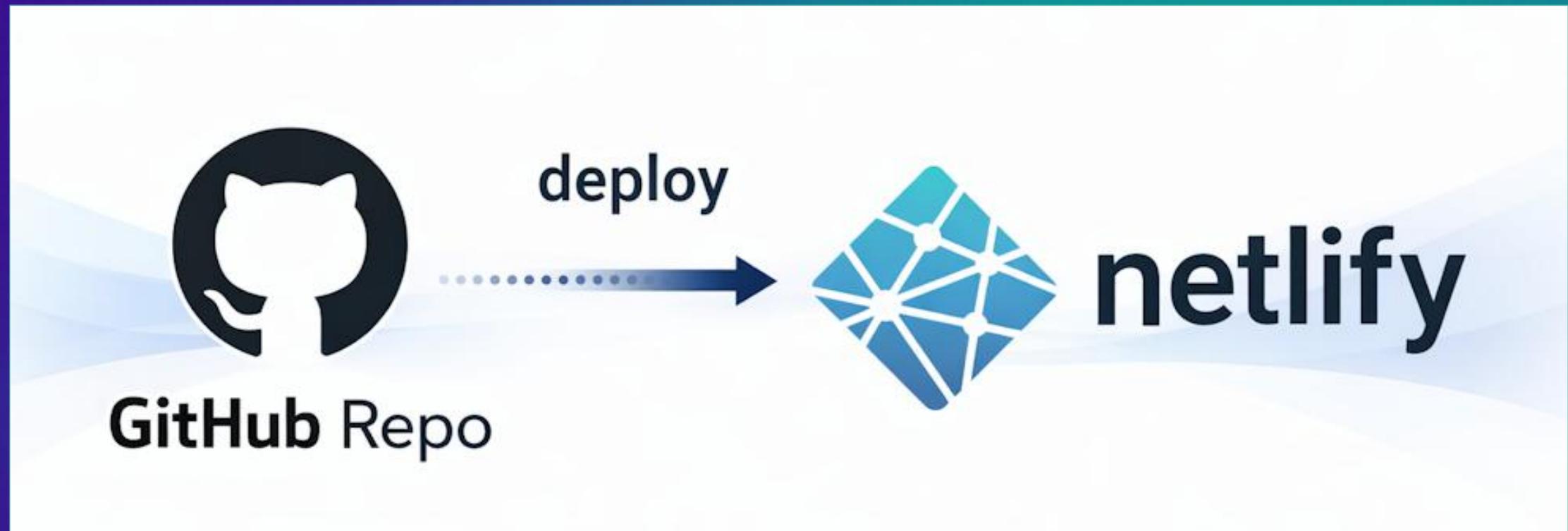
(10 minutes)

Start timer



Publish to Netlify

Deploy a Project from GitHub to Netlify

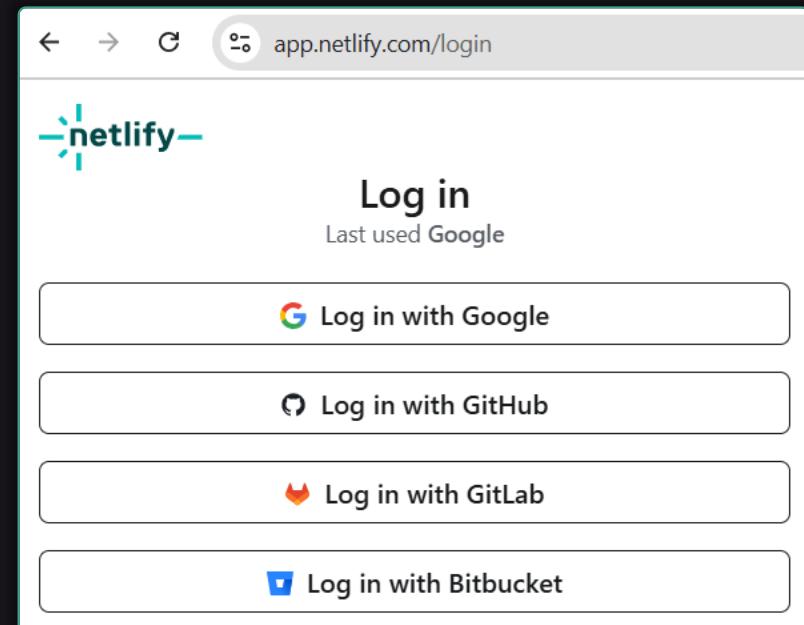
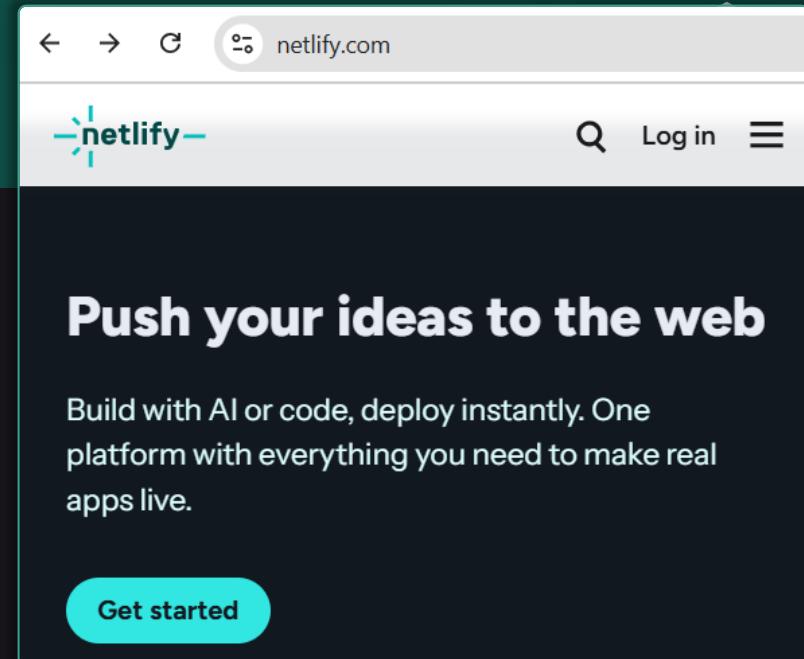


App Hosting Environments

- Apps can run **locally** or can be **hosted** in Internet
 - **Local app** example: you install a game on your smartphone
 - **Hosted app** example: a Web site, e. g. <https://apple.com>
- **App hosting environment** == computing resources, where a web site / app is running
 - **Server** in a data center, which holds your **app** and **data**
 - **Cloud** environment, e.g. **Netlify**, **Vercel**, **Cloudflare**, **Azure**
 - **Paid** and **free** app hosting environments
- Example of app hosting platform: <https://netlify.com>

Deployment to Netlify

- To **publish an app** on the Internet, we need to **deploy it somewhere**
- **Netlify** is a popular cloud-based app hosting service: <https://netlify.com>
 - **Free plan**, ideal for simple sites and vibe-coded apps
- **Register** a free Netlify account
 - <https://app.netlify.com/signup>



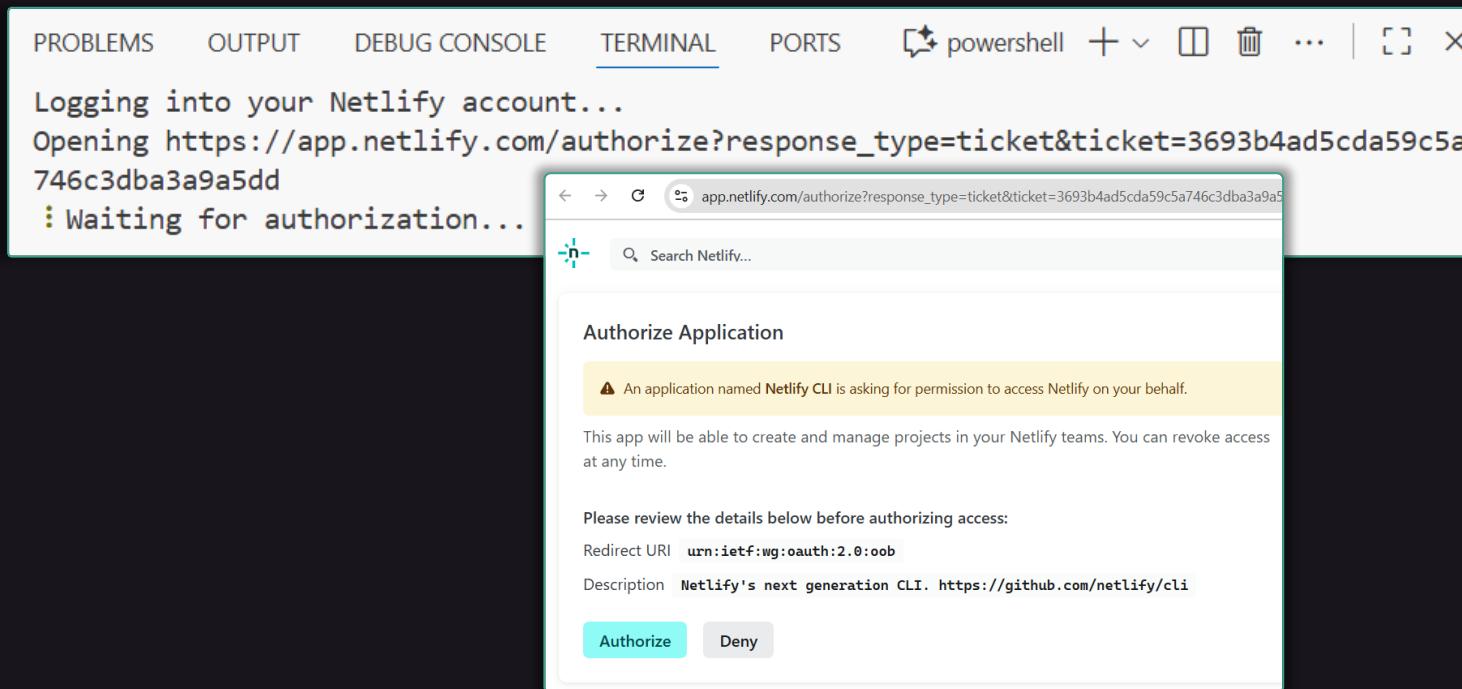
Deployment with Copilot to Netlify



1. Type this simple **prompt** in the GitHub Copilot chat:

Deploy this app to Netlify.

2. Authorize Netlify-CLI with Netlify



Deployment with Copilot to Netlify (2)



3. Configure the deployment settings:

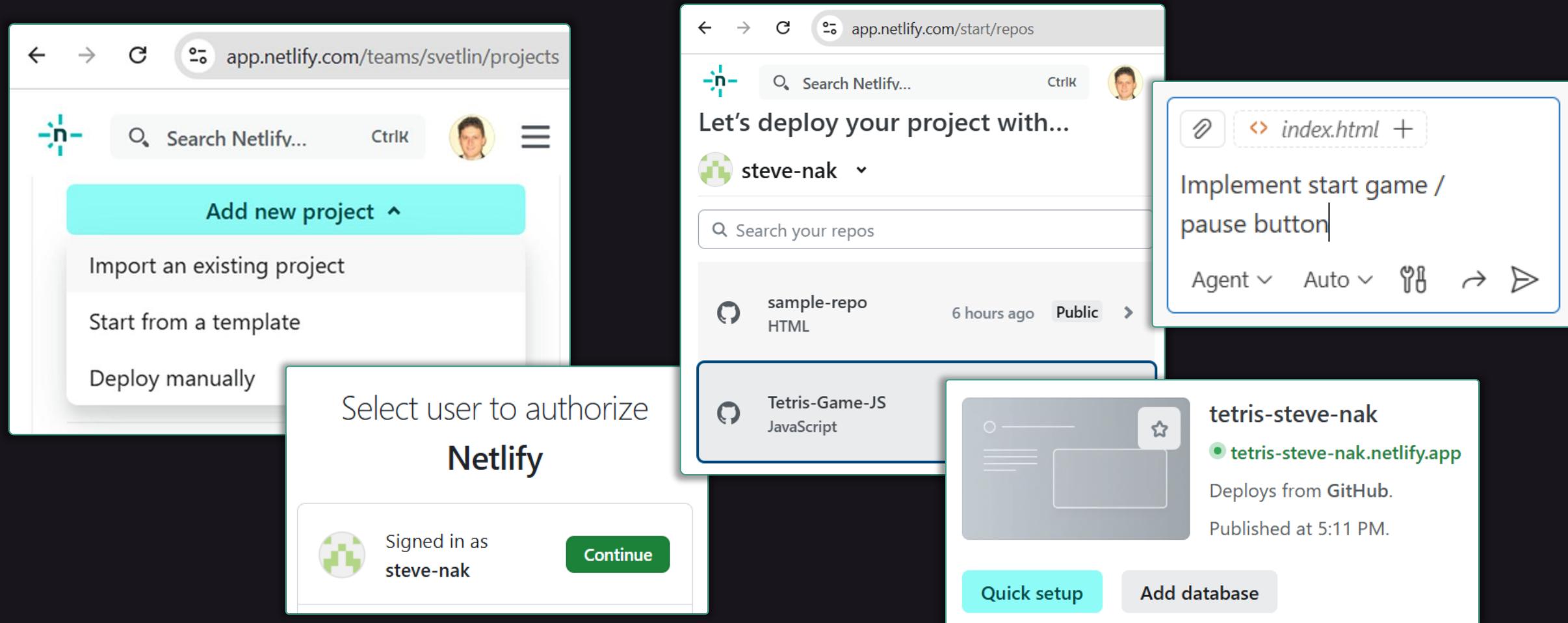
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS node + ⌂ ⚡ ... [ ] X  
PS C:\Work\sample-app> netlify deploy --prod --dir=.  
  
To create and deploy in one go, use: netlify deploy --create-site <SITE_NAME> --dir . --prod  
? What would you like to do?  
  ↵ Link this directory to an existing project  
> + Create & configure a new project  
  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS node + ⌂ ⚡ ... [ ] X  
PS C:\Work\sample-app> netlify deploy --prod --dir=.  
  
To create and deploy in one go, use: netlify deploy --create-site <SITE_NAME> --dir . --prod  
? What would you like to do? + Create & configure a new project  
? Team: SoftUni Nakov  
? Project name (leave blank for a random name; you can change it later): bathroom-calculator
```

4. Now your app is live on the Internet

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell + ⌂ ⚡ ... [ ] X  
PS C:\Work\sample-app> netlify deploy --prod --dir=.  
  
◆ Production deploy is live ◆  
  
Deployed to production URL: https://bathroom-calculator.netlify.app  
Unique deploy URL: https://68dee1b4d242173083d5bbc7--bathroom-calculator.netlify.app
```

Publishing a GitHub Project to Netlify

- Netlify can **auto-deploy** projects from a **GitHub repo**

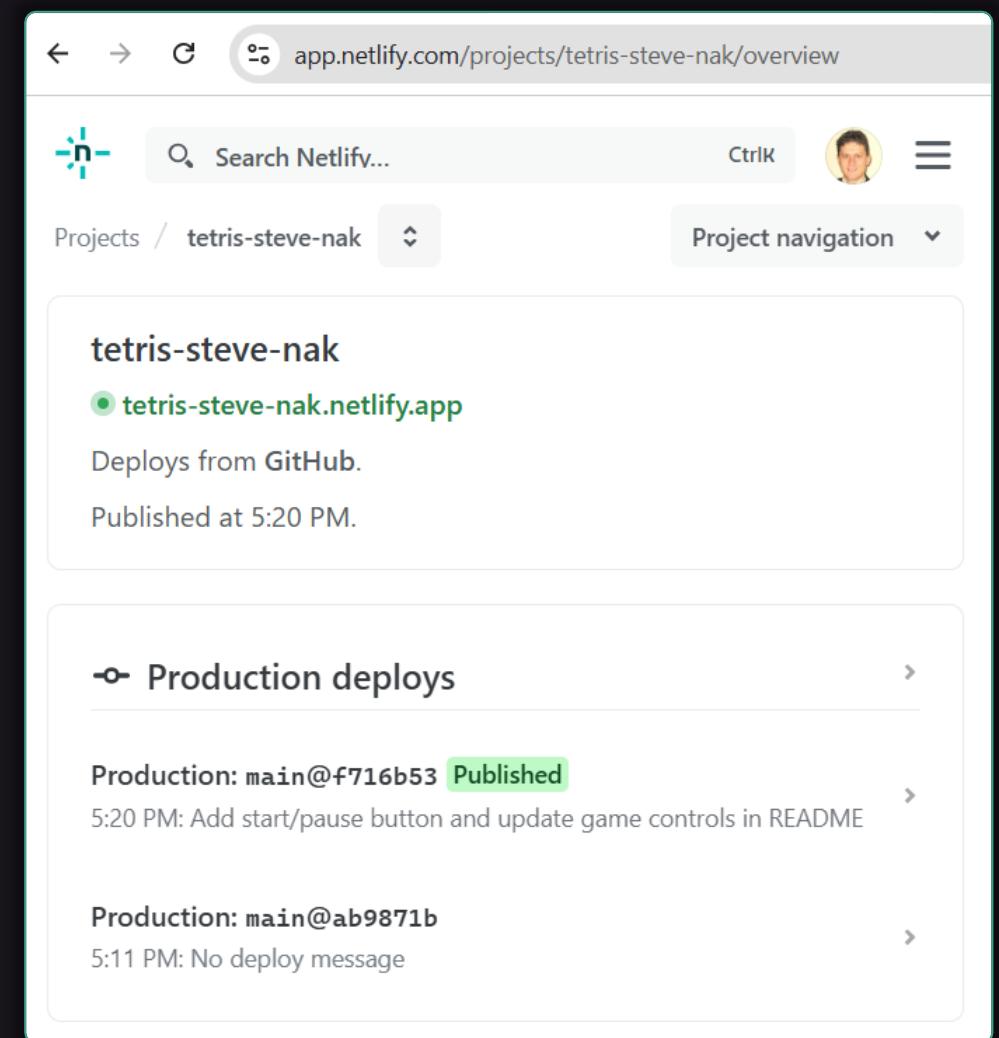
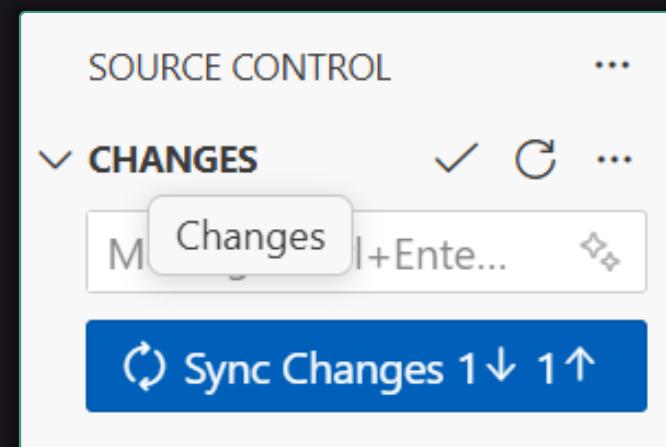
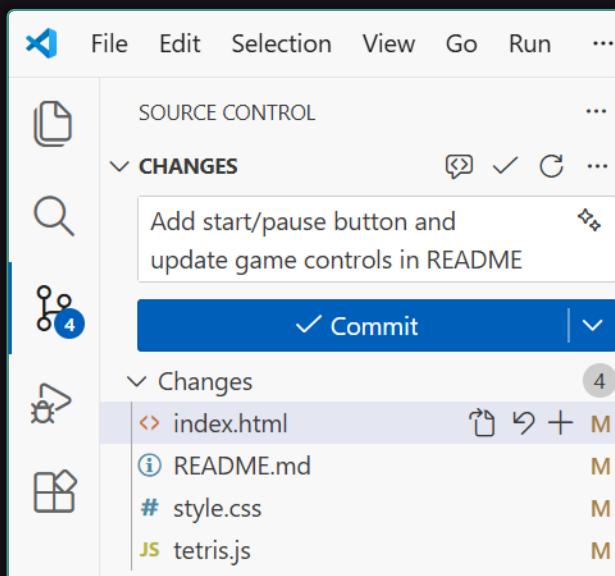


The screenshot illustrates the process of publishing a GitHub project to Netlify. It shows three main components:

- Left Panel:** Shows the URL `app.netlify.com/teams/svetlin/projects`. It features a prominent teal button labeled "Add new project". Below it are options: "Import an existing project", "Start from a template", and "Deploy manually". A modal window titled "Select user to authorize Netlify" is open, showing the user "steve-nak" is signed in, with a "Continue" button.
- Middle Panel:** Shows the URL `app.netlify.com/start/repos`. It displays a search bar and the message "Let's deploy your project with...". It lists a "sample-repo" (HTML) deployed 6 hours ago, and a "Tetris-Game-JS" (JavaScript) repository.
- Right Panel:** Shows a detailed view of the "Tetris-Game-JS" project. It includes a code editor with the file "index.html" containing the text "Implement start game / pause button". Deployment settings show "Agent" and "Auto" selected. The project was published at 5:11 PM.

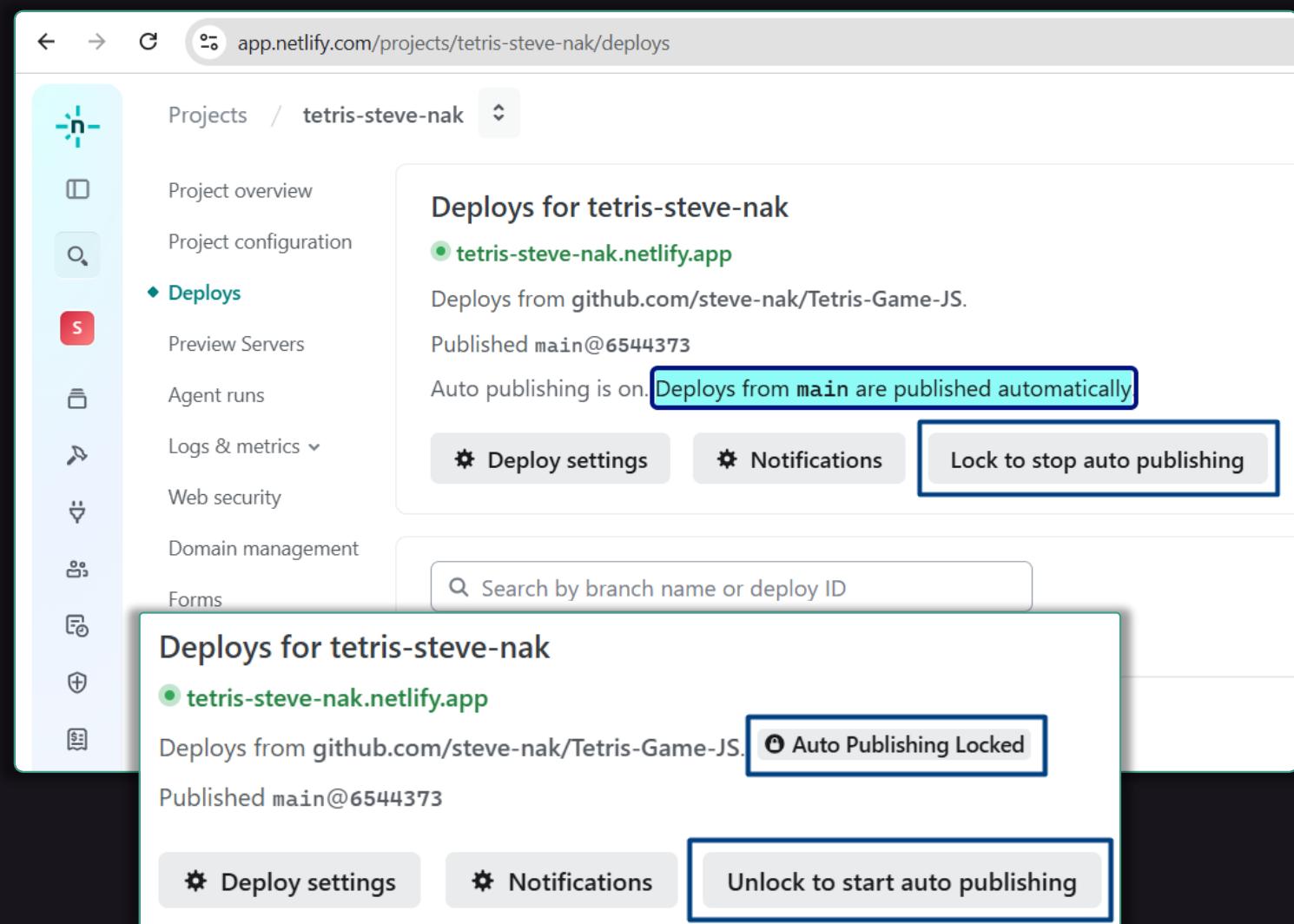
Continuous Deployment

- Netlify support "continuous deployment" from GitHub
 - Auto-deploy a GitHub repo** when a new version is pushed



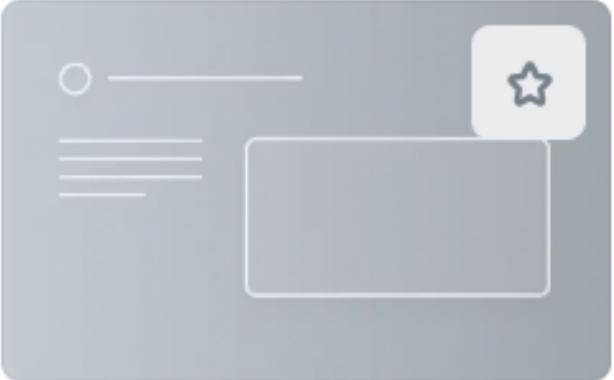
Enables / Disable Auto-Deploy

- Enable / disable auto-deploy on push in the "main" branch
 - Enabled by default
- Disable to save Netlify credits
 - Deploy manually for more control
- Use [Lock to stop auto publishing] button



Auto-Deploy from GitHub to Netlify

Live Demo



A screenshot of a Netlify deployment preview card. It features a small icon of a computer monitor with a single line of code on the screen, followed by a star icon. To the right of the icon, the repository name "tetris-steve-nak" is displayed in bold black text. Below it, a green circular icon with a white dot indicates the site is live at tetris-steve-nak.netlify.app. A descriptive text block below states "Deploys from GitHub." and "Published at 5:11 PM." At the bottom, there are two buttons: a teal button labeled "Quick setup" and a grey button labeled "Add database".

tetris-steve-nak

• tetris-steve-nak.netlify.app

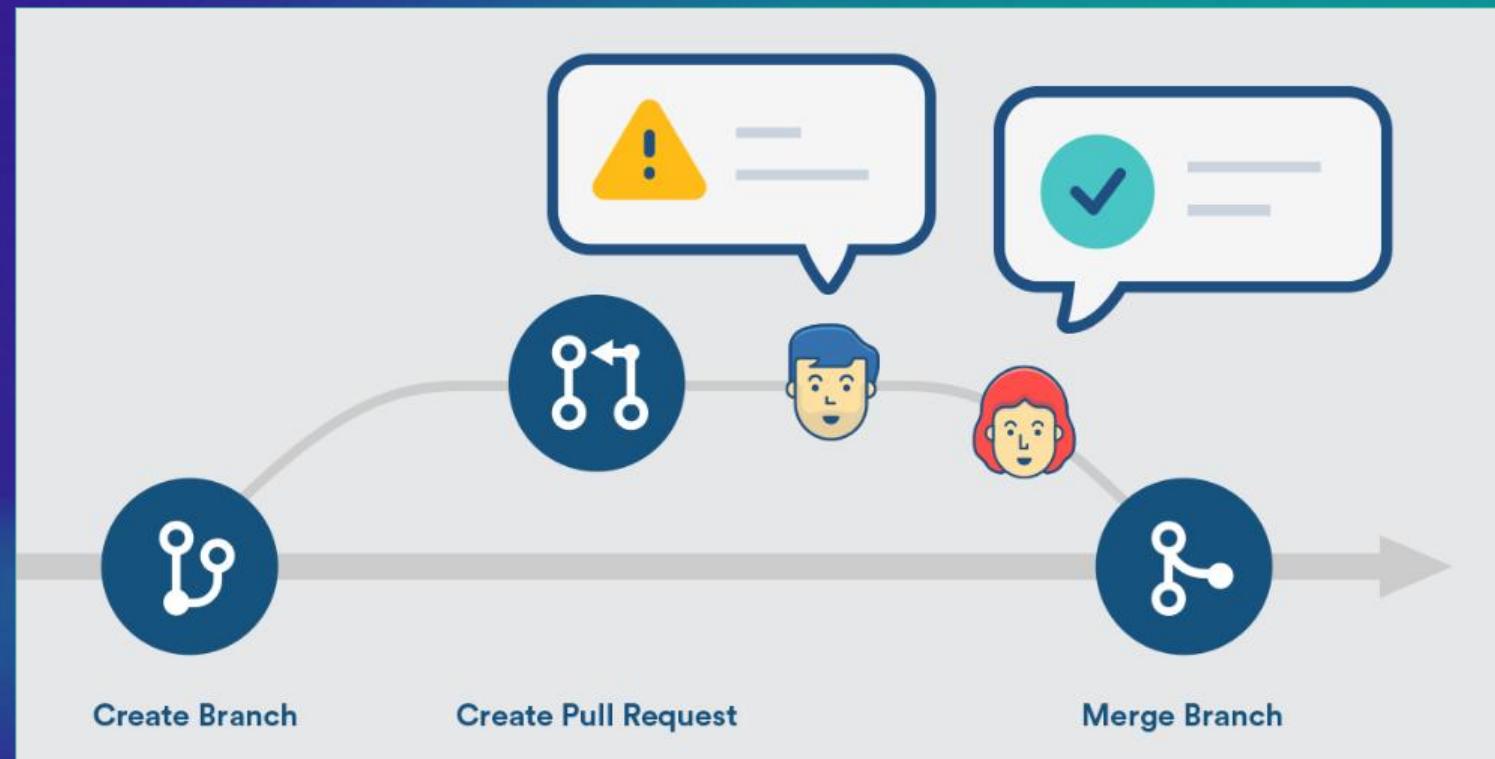
Deploys from GitHub.

Published at 5:11 PM.

Quick setup Add database

Branches and Pull Requests

Brief Intro to Branches and Pull Requests



The Concept of "Branches"

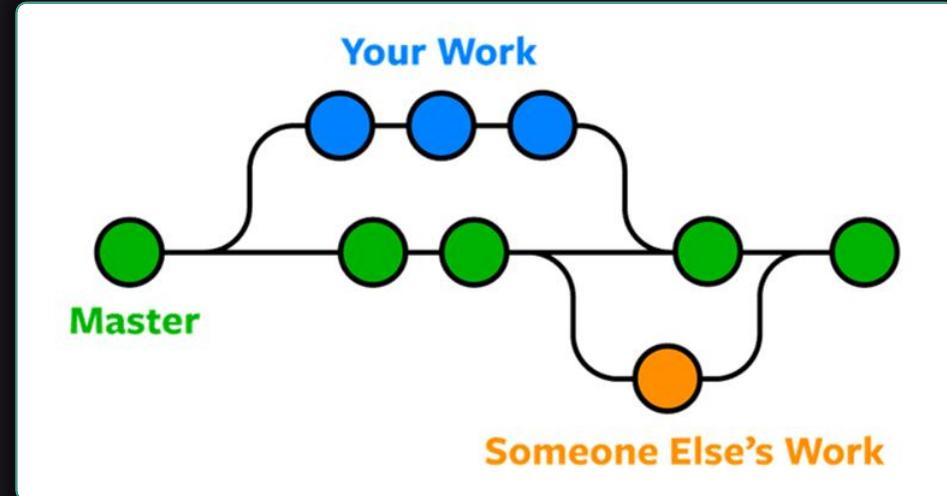
- **Branches** are separate versions of your project
 - Typical **workflow**: create a **branch**, implement a **feature**, **merge** to the **main** branch



- Example
 - <https://github.com/tailwind-labs/tailwindcss/branches>
- **Branches** allow several developers to **work concurrently** and finally **merge** their work

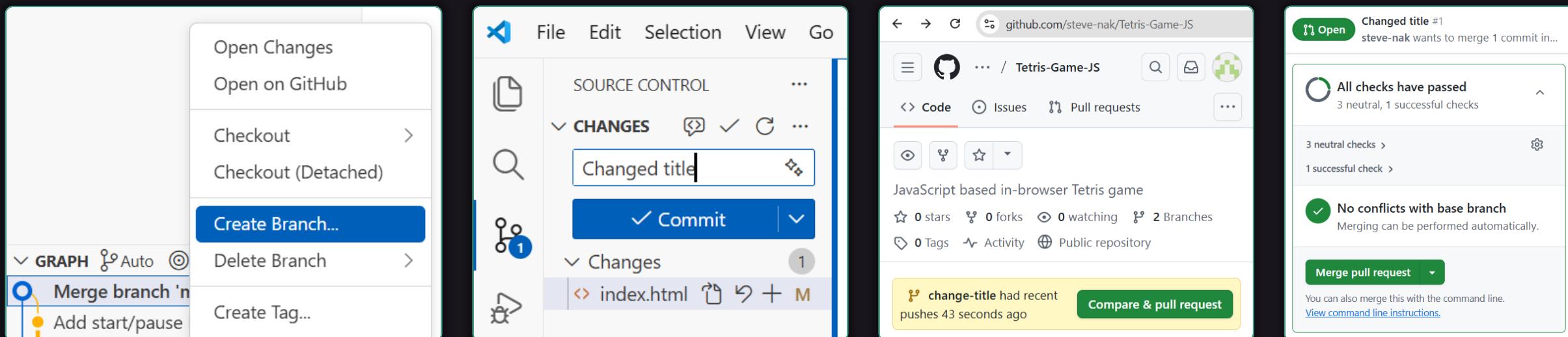
Branches

- Branches are **parallel development tracks**
- When the work is **complete**, a branch can be **merged** with the main project
 - You can even **switch between branches** without them interfering with each other
 - **Branching** is advanced technique, for professional software engineers → **avoid** when possible



The Concept of "Pull Request"

- The "pull requests" in GitHub are **branch merge requests**, which typically pass **quality checks** and a **code review**
- Sample pull request: <https://github.com/twbs/bootstrap/pull/38626>
- Create **branch** → **modify** → **pull request** → **merge**





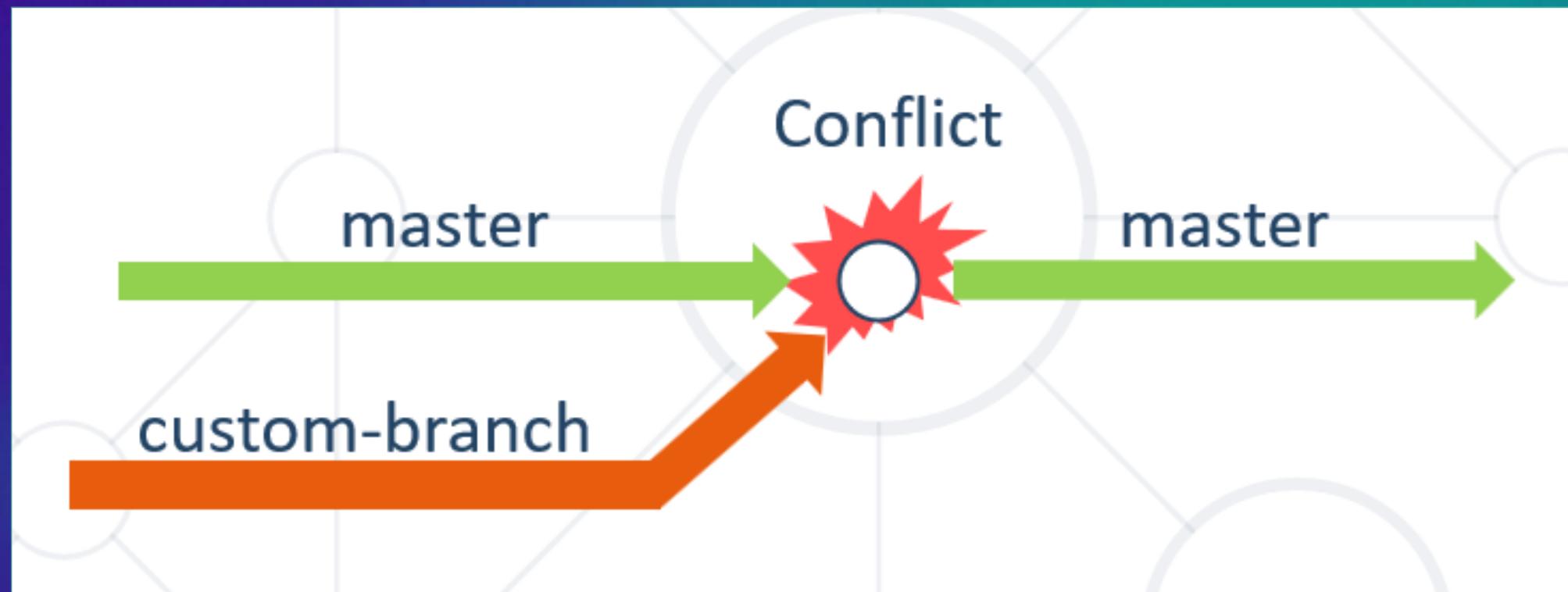
Merging a Pull Request

Live Demo

Create a **branch** → **modify** → **pull request** → **merge** → **delete branch**

Conflicts and Merges

How to Avoid and Resolve Conflicts?

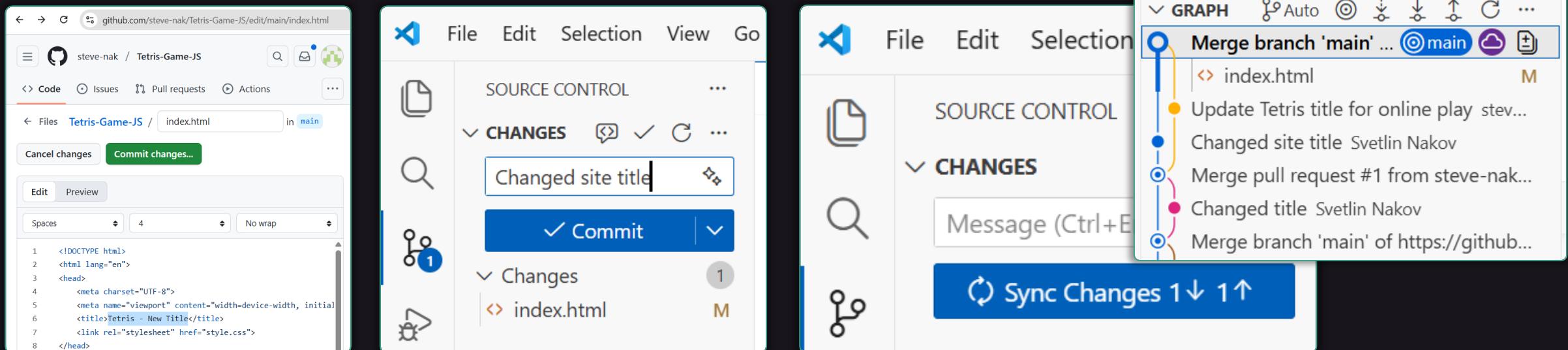


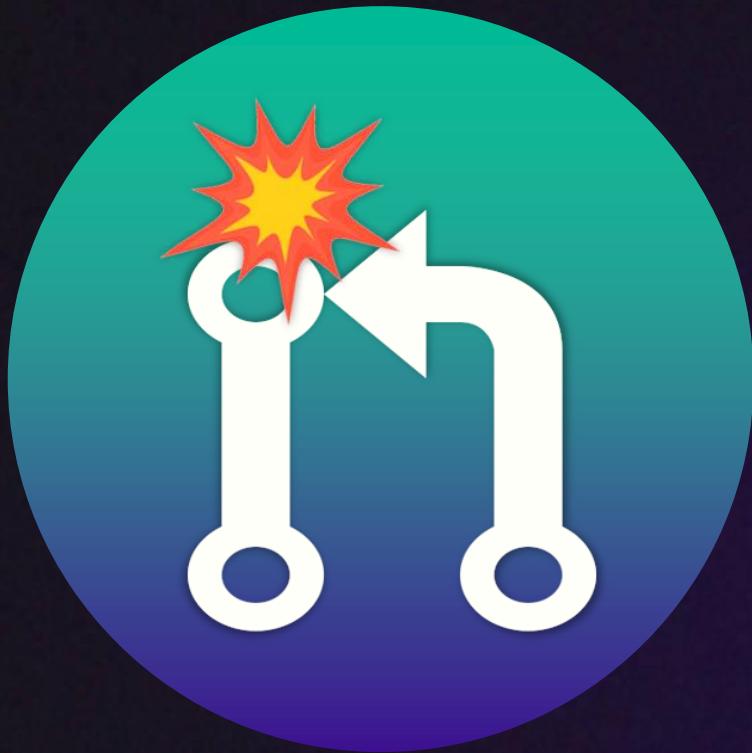
Conflicts in Git

- **Conflicts** are natural in software development
 - When two developers **modify the same code**
 - When you modify your code from VS Code and Github.com
- **Example:** Alan sets title to "New Title", Bob deletes title
- **Avoid conflicts:** always **pull** before you prompt the AI dev agent
 - This reduces the chance of conflicts
 - Still, **conflicts are possible**
- When conflict occurs, parallel changes should be **merged**

Handling Conflicts

- Handling a conflict in VS Code – example
- 1. **Modify** something from VS Code (don't commit still)
- 2. Go to **GitHub.com** and modify the code online
- 3. Try to commit and push → **conflict!** → merge





Resolving a Conflict

Live Demo

Edit from VS Code → **Edit** from GitHub.com
→ **Commit & Push** → **Resolve** a Conflict

Lesson Summary

- **Git** holds **version history** for software projects
 - Concepts: **repo**, **clone**, **commit**, **discard**, **push**, **pull**, **sync**
 - **VS Code** has built-in UI for basic **Git** interactions
- **GitHub** is a public cloud for Git projects: repos, code, commits, pull requests, builds, issues, wikis, releases
- AI dev **workflow** with Git:
 - Clone → AI edits → test → commit / discard → push
 - **Recovering** from mistakes: checkpoints, discard changes, undo commit, re-clone a repo, revert to old version

Questions?



Postbank – Exclusive Partner for SoftUni AI



- One of the leading **banking institutions** in Bulgaria
- Member of the Eurobank Group with € 103 billion assets
- Innovative trendsetter with next generation **beyond banking**, transforming today, empowering tomorrow
- Certified **Top Employer 2025** by the international Top Employers Institute
- Proven people care and **wellbeing initiatives**
- Benefits and unlimited access to professional, **personal and leadership trainings and programs**
- www.postbank.bg / careers.postbank.bg



Diamond Partners of Software University



Diamond Partners of SoftUni Digital



**SUPER
HOSTING
.BG**



Diamond Partners of SoftUni Digital



HUMAN

NETPEAK
DIGITAL GROWTH PARTNER



Marmalab | Е-комерс агенция

ETIEN YANEV
Break Your *Limits*. Live Your *Brand*.

1FORFIT



Diamond Partners of SoftUni Creative



Organization Partners of SoftUni Creative



THE
BUCKS
TOWN'S
WORK

