# GitHub Actions CI/CD Setup Wizard for Diplomagic

Follow these steps to configure GitHub Actions workflows for the Diplomagic project. Each step ends with a prompt; enter **y** to continue or **n** to repeat the step.

## Step 1: Configure MkDocs deployment for Diplomagic\_GDD

This step creates a workflow that builds the MkDocs documentation and publishes it to the gh-pages branch (the default branch for project pages[[1]](https://www.mkdocs.org/user-guide/deploying-your-docs/#:~:text=mkdocs%20gh)). In your **Diplomagic\_GDD** repository:

* Create a folder and file:
* mkdir -p .github/workflows
* Create .github/workflows/mkdocs.yml and add the following YAML. This workflow triggers on pushes to main, sets up Python, installs MkDocs, builds the site, and then deploys to the gh-pages branch using the [peaceiris/actions-gh-pages](https://github.com/peaceiris/actions-gh-pages) action:

name: Build and Deploy MkDocs  
  
on:  
 push:  
 branches:  
 - main  
  
jobs:  
 deploy:  
 runs-on: ubuntu-latest  
 steps:  
 - uses: actions/checkout@v2  
 with:  
 fetch-depth: 0  
  
 - name: Set up Python  
 uses: actions/setup-python@v2  
 with:  
 python-version: '3.x'  
  
 - name: Install dependencies  
 run: |  
 python -m pip install --upgrade pip  
 pip install mkdocs mkdocs-material  
  
 - name: Build documentation  
 run: mkdocs build  
  
 - name: Deploy to GitHub Pages  
 uses: peaceiris/actions-gh-pages@v3  
 with:  
 github\_token: ${{ secrets.GITHUB\_TOKEN }}  
 publish\_dir: ./site  
 publish\_branch: gh-pages

This example (adapted from a similar setup[[2]](https://bluegenes.github.io/mkdocs-github-actions/#:~:text=,pages%40v3%20with%3A%20github_token%3A)) checks out the code, installs MkDocs and a theme, builds the docs, and pushes the site/ output to gh-pages. MkDocs’ gh-deploy normally uses gh-pages by default[[1]](https://www.mkdocs.org/user-guide/deploying-your-docs/#:~:text=mkdocs%20gh).

*(If you have a custom domain for your docs, include a CNAME file in your docs folder so it’s carried to gh-pages*[*[3]*](https://www.mkdocs.org/user-guide/deploying-your-docs/#:~:text=GitHub%20Pages%20includes%20support%20for,that%20the%20file%20does%20not)*.)*

Save and commit mkdocs.yml, then push it to the main branch.

**Prompt:** Type y to continue or n to retry this step.

## Step 2: Configure MkDocs deployment for **goldmanvision.github.io**

Repeat the process for the GitHub Pages site repository (**goldmanvision.github.io**). This is a *user/organization* site, so the content must end up on main instead of gh-pages[[4]](https://www.mkdocs.org/user-guide/deploying-your-docs/#:~:text=Note%20that%20you%20need%20to,which%20you%20probably%20don%27t%20want). In the **goldmanvision.github.io** repo:

* Ensure .github/workflows exists. Create it if needed:
* mkdir -p .github/workflows
* Create .github/workflows/mkdocs.yml with contents similar to Step 1, but change publish\_branch to main (since user sites publish from main):

name: Build and Deploy MkDocs for User Site  
  
on:  
 push:  
 branches:  
 - main  
  
jobs:  
 deploy:  
 runs-on: ubuntu-latest  
 steps:  
 - uses: actions/checkout@v2  
 with:  
 fetch-depth: 0  
  
 - name: Set up Python  
 uses: actions/setup-python@v2  
 with:  
 python-version: '3.x'  
  
 - name: Install dependencies  
 run: |  
 python -m pip install --upgrade pip  
 pip install mkdocs mkdocs-material  
  
 - name: Build documentation  
 run: mkdocs build  
  
 - name: Deploy to GitHub Pages  
 uses: peaceiris/actions-gh-pages@v3  
 with:  
 github\_token: ${{ secrets.GITHUB\_TOKEN }}  
 publish\_dir: ./site  
 publish\_branch: main

This workflow builds the site and pushes the output to the main branch of the **goldmanvision.github.io** repo (which is the source for that Pages site[[4]](https://www.mkdocs.org/user-guide/deploying-your-docs/#:~:text=Note%20that%20you%20need%20to,which%20you%20probably%20don%27t%20want)). Commit and push mkdocs.yml.

*(Again, if using a custom domain for this site, include a CNAME file in the docs directory so MkDocs carries it into the published site*[*[3]*](https://www.mkdocs.org/user-guide/deploying-your-docs/#:~:text=GitHub%20Pages%20includes%20support%20for,that%20the%20file%20does%20not)*.)*

**Prompt:** Type y to continue or n to retry this step.

## Step 3: Create a placeholder game build/test pipeline

Now set up a stub CI workflow in the game repository (Unity or Unreal). This file is optional and contains only placeholder shell commands. In the game repo (e.g. **DiplomagicGame**):

* Create .github/workflows/game\_build.yml with content like this:

name: Game Build and Test (Placeholder)  
  
on:  
 push:  
 branches:  
 - main  
  
jobs:  
 build:  
 runs-on: windows-latest  
 steps:  
 - uses: actions/checkout@v2  
  
 - name: Install dependencies (placeholder)  
 run: |  
 echo "Installing game dependencies (placeholder)"  
  
 - name: Build game (placeholder)  
 run: |  
 echo "Building game project (placeholder)"  
  
 - name: Run game tests (placeholder)  
 run: |  
 echo "Running game tests (placeholder)"

This workflow triggers on pushes to main, uses a Windows runner (common for Unity/Unreal), checks out the code, and then runs dummy shell commands. Replace the echo lines with real build/test commands later. For now this ensures the workflow file structure is in place.

**Prompt:** Type y to continue or n to retry this step.

## Step 4: Set up branch protection rules

Protect the key branches so they cannot be force-pushed or deleted. We will protect gh-pages (from Step 1) and main. Perform these actions on **each repository** in GitHub:

1. In your repository on GitHub, go to **Settings** > **Branches**[[5]](https://docs.github.com/en/repositories/configuring-branches-and-merges-in-your-repository/managing-protected-branches/managing-a-branch-protection-rule#:~:text=1,main%20page%20of%20the%20repository).
2. Under **Branch protection rules**, click **Add rule**.
3. For **Branch name pattern**, enter the branch name (gh-pages or main).
4. Enable **Require status checks to pass** or reviews if desired (optional).
5. **Ensure that *Allow force pushes* is NOT enabled**. By default this setting is disabled on protected branches[[6]](https://docs.github.com/en/repositories/configuring-branches-and-merges-in-your-repository/managing-protected-branches/about-protected-branches#:~:text=By%20default%2C%20each%20branch%20protection,enable%20additional%20branch%20protection%20settings), which prevents history rewrites.
6. Click **Create** or **Save** to apply the rule.

Repeat for both gh-pages and main. By default, protected branches have force-pushes disabled[[6]](https://docs.github.com/en/repositories/configuring-branches-and-merges-in-your-repository/managing-protected-branches/about-protected-branches#:~:text=By%20default%2C%20each%20branch%20protection,enable%20additional%20branch%20protection%20settings), so leaving “Allow force pushes” unchecked will lock the history on those branches.

**Prompt:** Type y to continue or n to retry this step.

## Step 5: Enable GitHub Actions and check run history

Finally, ensure Actions is enabled and learn how to view workflow runs:

* **Enable Actions (if needed):** By default, GitHub Actions is enabled on new repositories[[7]](https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/enabling-features-for-your-repository/managing-github-actions-settings-for-a-repository#:~:text=By%20default%2C%20GitHub%20Actions%20is,GitHub%20Actions%2C%20see%20Writing%20workflows). You can check or change this in **Settings** > **Actions** > **General**[[8]](https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/enabling-features-for-your-repository/managing-github-actions-settings-for-a-repository#:~:text=1,main%20page%20of%20the%20repository). For example, navigate to your repo’s main page, select **Settings**, then **Actions** > **General** and confirm “Allow all actions” is selected.
* **View runs in the GitHub UI:** Go to the main page of the repository and click the **Actions** tab[[9]](https://docs.github.com/en/actions/how-tos/monitor-workflows/view-workflow-run-history#:~:text=1,main%20page%20of%20the%20repository). In the left sidebar, click the workflow you want (e.g. “Build and Deploy MkDocs”), then click a run to see details (logs, status)[[9]](https://docs.github.com/en/actions/how-tos/monitor-workflows/view-workflow-run-history#:~:text=1,main%20page%20of%20the%20repository).
* **View runs via CLI:** Using the GitHub CLI (gh), you can list recent workflow runs with:
* gh run list --limit 10
* This shows the last 10 workflow runs. To view details of a specific run, use gh run view <run-id> (add --log to see full logs)[[10]](https://docs.github.com/en/actions/how-tos/monitor-workflows/view-workflow-run-history#:~:text=To%20list%20the%20recent%20workflow,subcommand). For example:
* gh run list  
  gh run view 123456 --log

You have now set up CI/CD for documentation and a placeholder game build, protected the branches, and verified that Actions is enabled and runs are visible[[9]](https://docs.github.com/en/actions/how-tos/monitor-workflows/view-workflow-run-history#:~:text=1,main%20page%20of%20the%20repository)[[7]](https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/enabling-features-for-your-repository/managing-github-actions-settings-for-a-repository#:~:text=By%20default%2C%20GitHub%20Actions%20is,GitHub%20Actions%2C%20see%20Writing%20workflows).

**Prompt:** Type y to finish or n to review any steps.

**Done.**

**Sources:** Guidance was based on GitHub and MkDocs documentation and examples[[1]](https://www.mkdocs.org/user-guide/deploying-your-docs/#:~:text=mkdocs%20gh)[[9]](https://docs.github.com/en/actions/how-tos/monitor-workflows/view-workflow-run-history#:~:text=1,main%20page%20of%20the%20repository)[[6]](https://docs.github.com/en/repositories/configuring-branches-and-merges-in-your-repository/managing-protected-branches/about-protected-branches#:~:text=By%20default%2C%20each%20branch%20protection,enable%20additional%20branch%20protection%20settings)[[7]](https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/enabling-features-for-your-repository/managing-github-actions-settings-for-a-repository#:~:text=By%20default%2C%20GitHub%20Actions%20is,GitHub%20Actions%2C%20see%20Writing%20workflows)[[2]](https://bluegenes.github.io/mkdocs-github-actions/#:~:text=,pages%40v3%20with%3A%20github_token%3A)[[5]](https://docs.github.com/en/repositories/configuring-branches-and-merges-in-your-repository/managing-protected-branches/managing-a-branch-protection-rule#:~:text=1,main%20page%20of%20the%20repository).

[[1]](https://www.mkdocs.org/user-guide/deploying-your-docs/#:~:text=mkdocs%20gh) [[3]](https://www.mkdocs.org/user-guide/deploying-your-docs/#:~:text=GitHub%20Pages%20includes%20support%20for,that%20the%20file%20does%20not) [[4]](https://www.mkdocs.org/user-guide/deploying-your-docs/#:~:text=Note%20that%20you%20need%20to,which%20you%20probably%20don%27t%20want) Deploying Your Docs - MkDocs

<https://www.mkdocs.org/user-guide/deploying-your-docs/>

[[2]](https://bluegenes.github.io/mkdocs-github-actions/#:~:text=,pages%40v3%20with%3A%20github_token%3A) Deploying Mkdocs via Github Actions – Tessa Pierce Ward

<https://bluegenes.github.io/mkdocs-github-actions/>

[[5]](https://docs.github.com/en/repositories/configuring-branches-and-merges-in-your-repository/managing-protected-branches/managing-a-branch-protection-rule#:~:text=1,main%20page%20of%20the%20repository) Managing a branch protection rule - GitHub Docs

<https://docs.github.com/en/repositories/configuring-branches-and-merges-in-your-repository/managing-protected-branches/managing-a-branch-protection-rule>

[[6]](https://docs.github.com/en/repositories/configuring-branches-and-merges-in-your-repository/managing-protected-branches/about-protected-branches#:~:text=By%20default%2C%20each%20branch%20protection,enable%20additional%20branch%20protection%20settings) About protected branches - GitHub Docs

<https://docs.github.com/en/repositories/configuring-branches-and-merges-in-your-repository/managing-protected-branches/about-protected-branches>

[[7]](https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/enabling-features-for-your-repository/managing-github-actions-settings-for-a-repository#:~:text=By%20default%2C%20GitHub%20Actions%20is,GitHub%20Actions%2C%20see%20Writing%20workflows) [[8]](https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/enabling-features-for-your-repository/managing-github-actions-settings-for-a-repository#:~:text=1,main%20page%20of%20the%20repository) Managing GitHub Actions settings for a repository - GitHub Docs

<https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/enabling-features-for-your-repository/managing-github-actions-settings-for-a-repository>

[[9]](https://docs.github.com/en/actions/how-tos/monitor-workflows/view-workflow-run-history#:~:text=1,main%20page%20of%20the%20repository) [[10]](https://docs.github.com/en/actions/how-tos/monitor-workflows/view-workflow-run-history#:~:text=To%20list%20the%20recent%20workflow,subcommand) Viewing workflow run history - GitHub Docs

<https://docs.github.com/en/actions/how-tos/monitor-workflows/view-workflow-run-history>