# Git Simulation Game – Classroom Activity

This activity helps students understand how Git works by acting out the steps of version control. Students will role-play Git commands such as add, commit, push, pull, and merge. Each group should have 3–4 students and some file cards (post-it notes or index cards labelled 'file1.txt', 'file2.txt').

## Roles

* Coder: Makes changes to files.
* Stager: Stages files ready to be committed.
* Committer: Commits staged files with a message.
* Pusher/Puller: Pushes changes to the remote repository or pulls updates from it.
* Remote Repo (Teacher): Acts as the central GitHub repository.

## Stage 1: Making Changes

1. Coder writes a simple change on a file card (e.g., 'Changed colour to blue').

2. File is currently unstaged.

Command: git status – shows what has changed.

## Stage 2: Staging and Committing

1. Stager places the file into the 'staging area' pile.

 Command: git add <file>

2. Committer writes a short commit message on the back of the card.

 Command: git commit -m "Changed header colour"

## Stage 3: Pushing to Remote

1. Pusher takes the committed file and hands it to the Remote Repo (teacher).

 Command: git push

2. Now the remote repo has the updated file.

## Stage 4: Pulling Updates

1. Another group pulls the updated file from the Remote Repo.

 Command: git pull

2. Their local repo is now in sync with the remote repo.

## Stage 5: Merge Conflict (Optional)

1. Two groups edit the same file differently (e.g., 'blue' vs 'red').

2. Both try to push their versions to the Remote Repo.

3. Teacher announces: Conflict! Git cannot merge automatically.

4. Groups must decide how to resolve the conflict (choose one version or combine).

 Command: git merge <branch>

## Wrap-Up Discussion

Ask students:

 - What did you notice about how Git tracks changes?

 - How does Git help teams collaborate safely?

 - Why is communication important when working in teams?

## Key Takeaway

Git is a system of checkpoints that helps developers save work, collaborate, and resolve conflicts when multiple people work on the same code.