Branch merging guidance

Since you're using Unity Version Control (Plastic SCM) rather than Git, here's how it works specifically in Unity's environment:

Scenario Recap

You and your teammate are each working on separate branches.

Example:

You → feature-A

Teammate → feature-B

You merge feature-A → main.

Now, main has your new changes.

Your teammate's branch (feature-B) is now behind main — meaning it doesn't include your recent updates.

What your teammate should do next

Step 1: Update their branch with the latest from main

Before they push or merge their own work, they need to merge main into their branch so both sets of changes align.

In Unity Version Control (Plastic SCM):

Open the Plastic SCM window (Window > Plastic SCM in Unity).

Go to the Branches tab.

Right-click on main and choose:

→ Merge from this branch

In the dialog, choose their branch (feature-B) as the destination.

Click Merge.

This pulls your latest updates from main into their branch — it won't delete or overwrite their local work.

Step 2: Handle any conflicts

If both of you changed the same files (like the same scene, prefab, or script), Unity/Plastic will flag conflicts.

They'll need to:

Use the Merge Tool to review each conflict.

Keep either their version, your version, or manually combine both.

After resolving, they check in (commit) the merged result.

Step 3: Merge their branch into main (when ready)

Once they've verified everything works, they can:

Merge their branch (feature-B) → main.

Test again to confirm nothing broke.

Delete or archive their feature branch when done

What not to do

Don't just "Update workspace to main" while still having uncommitted work — this can overwrite local changes.

Don't merge to main directly without first merging main into their branch — otherwise, their merge might overwrite your recent work.