

Github Actions Code

Action 1: Unit testing & automated builds (unit-tests.yaml)

```
name: Run Unit Tests
# description: For any pushes to the main branch, run all unit tests and then, if testing is successful, build a .jar file

on:
  push:
    branches: [ "main" ]

jobs:
  test:
    runs-on: ubuntu-latest
    permissions:
      contents: read
      checks: write # Needs to write extended output of tests

    steps:
      - name: Checkout latest commit
        uses: actions/checkout@v4

      - name: Set up JDK 17
        uses: actions/setup-java@v4
        with:
          java-version: '17'
          distribution: 'temurin'

      - name: Make gradlew executable
        run: chmod +x ./gradlew

      # Configure Gradle for optimal use in GitHub Actions, including caching of downloaded dependencies.
      # See: https://github.com/gradle/actions/blob/main/setup-gradle/README.md
      - name: Setup Gradle
        uses: gradle/actions/setup-gradle@a1da67850ed9a4cedd57bfd976089dd991e2582 # v4.0.0

      - name: Run all unit tests
        run: ./gradlew test

      - name: Publish Test Results
        uses: EnricoMi/publish-unit-test-result-action@170bf24d20d201b842d7a52403b73ed297e6645b
        if: always() # Want to run this step even if the testing step itself fails
        with:
          comment_mode: off # Don't want to leave a comment on related PRs
          files: |
            headless/build/test-results/**/*.xml

      # If tests pass, make a .jar file available to download
    build:
      runs-on: ubuntu-latest
      needs: test # Will be skipped if tests fail

      permissions:
        contents: read

      steps:
        - name: Checkout latest commit
          uses: actions/checkout@v4

        - name: Set up JDK 17
          uses: actions/setup-java@v4
          with:
            java-version: '17'
            distribution: 'temurin'

        - name: Make gradlew executable
          run: chmod +x ./gradlew

        # Configure Gradle for optimal use in GitHub Actions, including caching of downloaded dependencies.
        # See: https://github.com/gradle/actions/blob/main/setup-gradle/README.md
        - name: Setup Gradle
          uses: gradle/actions/setup-gradle@a1da67850ed9a4cedd57bfd976089dd991e2582 # v4.0.0

        - name: Build with Gradle Wrapper
          run: ./gradlew build

        - name: Upload build artifacts
          uses: actions/upload-artifact@v4
          with:
            name: UniSim Java Package
            path: lwjgl3/build/libs/* # Upload generated Java file ('*' used in case output file name changes)
```

Action 2: Automatic Merge (automatic_merge.yaml)

```
name: 'Automatic Merge'
# description: Automatically merge changes from main into active development branches

on:
  push:
    branches: ["main"]

jobs:
  merge-branches:
    name: Merge main into develop
    runs-on: ubuntu-latest

    strategy:
      matrix: # Use matrix to merge into any number of active development branches
        branch: ["dev1", "dev2"]

    permissions:
      contents: write
      pull-requests: write

    steps:
      - name: Checkout latest code version
        uses: actions/checkout@v4

      # Attempt to automatically merge latest changes
      - name: Merge main -> develop
        id: execute_merge
        uses: devmaxx/merge-branch@6ec8363d74aad4f1615d1234ae1908b4185c4313
        with:
          type: now
          from_branch: main
          target_branch: ${matrix.branch}
          message: 'CI: Merge latest changes from main'
          github_token: ${ secrets.GITHUB_TOKEN }
          continue-on-error: true

      # If unable to merge automatically, create a pull request to attract manual attention
      - uses: actions/checkout@v4
        if: always() && steps.execute_merge.outcome == 'failure' # Outcome property provides the real result (before continue-on-error)
        with:
          ref: ${matrix.branch}

      - name: Reset staging branch
        if: always() && steps.execute_merge.outcome == 'failure' # Have to include the always() property otherwise GitHub will assume success() && ...
        run: |
          git fetch origin main:main
          git reset --hard main

      - name: Create Pull Request
        if: always() && steps.execute_merge.outcome == 'failure'
        uses: peter-evans/create-pull-request@5e914681df9dc83aa4e4905692ca88beb2f9e91f
        with:
          title: 'CI: Merge latest changes from main'
          reviewers: Lime-Parallelogram
          branch: staging
          labels: 'ci-automated'
```