# Continuous Integration (CI) and Git Branches

#### **Objectives**

- To gain fundamental understanding and hands-on experience on CI
- To gain understanding about git branches and manage branches whenever you want to add a new work and push it to the original repository via merge request

This lab will give you some hands-on experience in using continuous integration tools to automate the integration when members of a team push new/modified code into the remote repository. There are several CI tools to choose from, for example Jenkins, GitLab CI, TeamCity, and Travis CI etc...

In this Lab, you will be able to set up a CI system using GitLab, which is one of the more popular CI tools. In GitLab, you can create projects for hosting your codebase, collaborate on code, and automate all sorts of tasks related to building, testing, and delivering or deploying software continuously with built-in GitLab CI/CD. Builds can be triggered by time or event based.

You will be using the FreeCol codebase (FreeCol is a turn-based strategy game), which resides at <u>https://gitlab.liu.se/jesji387/group3</u>, along with GitLab from LiU (<u>https://gitlab.liu.se/</u>), and docker hub images available in <u>https://hub.docker.com/</u> to build FreeCol application. FreeCol application [<u>http://www.freecol.org/</u>] is a Java module, aims to create an open-source version of the game Colonization. FreeCol project's build script, using **Ant**, is configured to build, generate HTML documentation, code coverage reports, automated testing report etc. Apache Ant (<u>https://ant.apache.org/</u>) is a Java library and command-line tool with a number of built-in tasks allowing to compile, assemble, test, and run Java applications

Recommended reading before you start working on this lab:

- Introduction to CI/CD with GitLab
   <u>https://docs.gitlab.com/ee/ci/introduction/index.html#how-gitlab-cicd-works</u>
- GitLab CI/CD Pipeline Configuration Reference <u>https://docs.gitlab.com/ee/ci/yaml/README.html</u>
- Creating and Tweaking GitLab CI/CD for GitLab Pages-<u>https://docs.gitlab.com/ee/user/project/pages/getting\_started\_part\_four.html</u>
- JUnit test reports <u>https://docs.gitlab.com/ee/ci/junit\_test\_reports.html</u>

## **Table of Contents**

| Part A – Tutorial on how to set up a CI using Gitlab                   | 3  |
|--|----|
| 1. Create a "helloworld" project in GitLab                             | 3  |
| 2. Clone your remote repository  | 5  |
| Questions:   | 6  |
| 3. Working on your local repository                                    | 7  |
| 4. Gitlab CI configuration for "helloworld" project                    | 8  |
| Part B – GitLab CI Configuration and Git Branches                      | 11 |
| 1. Setting up a shared Git repository                                  | 11 |
| 2. Forking the FreeCol repository                                      | 13 |
| 3. Cloning the repository  | 14 |
| 4. Run build scripts in FreeCol in your local computer                 | 16 |
| 5. Set up Continuous Integration in GitLab                             | 17 |
| 6. Step 4. Create a merge request in GitLab to the original repository | 20 |
| 7. Merge the proposed changes  | 21 |
| 8. Task 5. Fixing the test cases                                       | 21 |
| 9. Merge test fix into the master branch                               | 22 |
| Examination  | 24 |

## Part A – Tutorial on how to set up a CI using

## Gitlab

## 1. Create a "helloworld" project in GitLab

- 1. Sign in on https://gitlab.liu.se/ using your LiU account.
- 2. In your dashboard, click the green **New project** button or use the plus icon in the navigation bar.

| II.U Projects 🗸 Groups 🗸 More 🗸 🖓 🗠 🖿  |   | 0 ~ 2 | Search or jump to  | ٩                  | 0)            | n                    | ß        | <b>?</b> ~ | ÷ | ^ |
|--|---|-------|--|--------------------|---------------|----------------------|----------|------------|---|---|
| Welcome to GitLab<br>Code, test, and deploy together   | New project<br>New group<br>New snippet |       |  |                    |               |                      |          |            |   |   |
| Create a project<br>Projects are where you store your code, access<br>issues, wiki and other features of GitLab.         |   |       | Create a group<br>Groups are the bee<br>members.           | <b>p</b><br>st way | to man        | age pr               | ojects a | and        |   |   |
| Explore public projects<br>There are 121 public projects o<br>projects are an easy way to allo<br>have read-only access. | n this server. Public<br>w everyone to  |       | Learn more al<br>Take a look at the<br>of GitLab's capabil | docum<br>docum     | <b>GitLal</b> | <b>)</b><br>on to di | iscover  | all        |   | • |

#### This opens the **New project** page.

| Blank project   | Create from template                                     | Import project                        | CI/CD for external repo |
|---|--|---------------------------------------|-------------------------|
| Project name  |  |                                       |                         |
| helloworld  |  |                                       |                         |
| Project URL   |  | Project slug                          |                         |
| https://gitlab.liu.se/alash325/                                     |  | helloworld                            |                         |
| Want to house several dependent                                     | projects under the same namespace?                       | Create a group.                       |                         |
| Project description (optional)                                      |  |                                       |                         |
| Description format  |  |                                       | ħ.                      |
| <ul> <li>Private</li> <li>Project access must be gra</li> </ul>     | nted explicitly to each user.                            |                                       |                         |
| <ul> <li>Internal</li> <li>The project can be accessed</li> </ul>   | d by any logged in user.                                 |                                       |                         |
| Public<br>The project can be accesse                                | d without any authentication.                            |                                       |                         |
| Initialize repository with a RE.<br>Allows you to immediately clone | <b>ADME</b><br>e this project's repository. Skip this if | you plan to push up an existing repos | itory.                  |
| Create project  |  |                                       | Cancel                  |

- 3. On the New project page, choose the blank project tab.
- 4. On the **Blank project** tab, provide the following information:
  - The name of your project (i.e., **helloworld**) in the **Project name** field.
  - Select the Initialize repository with a README option to create a README file.
- 5. Click **Create project.** Now your Git repository is initialized with a default branch (master).

#### 2. Clone your remote repository

| Н                          | helloworld 🔂<br>Project ID:                                     |                                |   | û ∽ 🛱 Star 0   |
|----------------------------|---|--------------------------------|---|----------------|
| <b>The re</b><br>You can g | pository for this project<br>et started by cloning the reposito | <b>ct is en</b><br>ry or start | adding files to it with one of the following options. |                |
| Clone V                    | Vew file Add RE   | ADME                           | Add LICENSE     Add CHANGELOG     Add CONTRIBUTING    | € Set up CI/CD |
| git@                       | gitlab.liu.se:  | Ĝ                              | using the instructions below.                         |                |
| Clone v                    | vith HTTPS  |                                |   |                |
| http                       | s://gitlab.liu.se/  | G                              |   |                |
| git co                     | nfigglobal user.email   |                                |   |                |

To start working locally on your remote repository, you must clone (download) a copy of its files to your local computer. You can clone it via HTTPS:

- From your **helloworld** project click on the **Clone** button
- Copy the URL from "Clone with HTTPS" field
- Open a terminal in the directory you wish to clone the repository files into, and run the following command:

#### git clone [repository\_URL]

where [repository\_URL] is the URL (remote repository path) you copied.

• Enter your credentials if asked.

Once completed, the following actions occur:

- A new folder called **hellworld** initialized as a Git repository is created in your local computer
- A remote named *origin* is created, pointing to the URL you cloned from. Go to helloworld folder and type *git remote -v* and press Enter. You will see the current configured remote repository:

```
origin https://gitlab.liu.se/<USERNAME>/helloworld.git (fetch)
origin https://gitlab.liu.se/<USERNAME>/helloworld.git (push)
```

• All of the repository's files and commits are downloaded there (for now only the README file)

• The default branch (usually called master) is checked out. Type *git branch* and press **Enter**. You will see your branches. A \* will appear next to the currently active branch: \* master.

#### **Questions:**

Write few lines (i.e., what, and why mostly) about each of the following terms, with respect to GIT:

- 1) Origin
- 2) Remote
- 3) Branching (master or other) – Creating a new branch
- 4) Forking
- 5) Merge Request
- 6) **Continuous Integration**
- 7) Continuous Delivery
- 8) Making a pull request
- 9) **Build Script**

}

10) .yml file in CI workflow

## 3. Working on your local repository

1. Create a file called HelloWorld.java in you project root, which contains the simplest program of Java printing "Hello World" to the screen.

```
class HelloWorld
```

```
{
  public static void main(String args[])
   {
     System.out.println("Hello, World");
   }
```

- 2. Push all your changes to Gitlab (your remote repository):
  - Run the command "git status" to know how many files have been added/changed.
  - Add the file to your local repository and stage it for commit to your local repository:

```
git add . or git add HelloWorld.java
```

Note: The . character means all file changes in the current directory and subdirectories.

Commit the file that you've staged in your local repository:

git commit -m "helloworld", where -m stands for message, you want to associate with these changes so you remember, why you made these changes.

Push all commits in your local repository to GitLab (remote repository) • git push [name\_of\_your\_remote] [name\_of\_your\_branch]

On successful completion, your new file "HelloWorld.java" will be added to your remote repository in GitLab webpage as well as on local computer.

| Projects 🗸 Groups 🗸   | Activity Milestones Snippets 🙆 ~ 🖿     | 0 ~                  | Search or jump to  | o dy m          | ⊻ 0 × ∰ ×     |
|-----------------------|--|----------------------|--------------------|-----------------|---------------|
| H helloworld          | Alachew Mengist > helloworld > Details |                      |                    |                 |               |
| 🔂 Project             | H helloworld                           |                      | Q ~ \$             | Star 0 8 Fo     | ork 0 Clone • |
| Details               | Project ID. 1810                       |                      |                    |                 |               |
| Activity              | Add license → 1 Commit      P 1 Branch | 0 Tags 🛛 20 KB Files |                    |                 |               |
| Releases              |  |                      |                    |                 |               |
| Cycle Analytics       | master 👻 helloworld / 🕇 🗸              |                      | Histo              | ory Q Find file | Web IDE 🛛 🖓 🖌 |
| Insights              | helloworld                             |                      |                    |                 | e415ad92 6    |
| Repository            | Alachew Mengist authored 4 minutes ag  | 0                    |                    |                 |               |
| D) Issues             | Add README     Add CHANGELOG           | ☑ Add CONTRIBUTING   | Enable Auto DevOps | 🔁 Add Kubernete | es cluster    |
| 1 Merge Requests 0    |  |                      |                    |                 |               |
| 🚀 CI / CD             |  |                      |                    |                 |               |
| Security & Compliance | Name                                   | Last commit          |                    |                 | Last update   |
| Collapse sidebar      | HelloWorld.java                        | helloworld           |                    |                 | 4 minutes ago |

## 4. Gitlab CI configuration for "helloworld" project

• Create .gitlab-ci.yml and save it in your root directory of "helloworld" project. The

| <pre>image: openjdk:8</pre> |
|-----------------------------|
| stages:                     |
| - build                     |
| build:                      |
| stage: build                |
| script:                     |
| - javac HelloWorld.java     |
| - java HelloWorld           |
| artifacts:                  |
| paths:                      |
| /HelloWorld.*               |

contents of .yml file is as follows:

**NOTE**: YAML does **not** allow tab indentation, so use 2 spaces instead. If you are using vim to create the ci file, run **set ts=2 sw=2 et** and add a line containing **# vi: ts=2 sw=2 et** at the bottom of the file. Ensure that the file **~/.vimrc** contains the line **set modeline**.

**Question:** Explain each line in the figure above.

 Push all your changes to Gitlab (your remote repository) git add . git commit -m "initial CI" git push [name\_of\_your\_remote] [name\_of\_your\_branch] On successful completion, GitLab CI automatically starts to execute the jobs you've set. Go to option on left side on gitlab.liu.se "CI/CD" and click on "pipelines". You should see the status of you last commit changes from **pending** to either **running**, **passed** or **failed** as shown below (see your Gitlab repository).

| ∎.∪ Projects ∨ Groups ∨ | • Activity Milestones Snippets 🍙 ~ 🖿  | 🛢 ~ Search or jump to q 🕩 🏦 🗹 🚱 ~ 🌐 ~                    |
|-------------------------|---|--|
| H helloworld            | Alachew Mengist > helloworld > Pipelines  |  |
| 🔂 Project               | All 1 Pending 0 Running 1 Finished 0 Branches Tags  | Run Pipeline         Clear Runner Caches         Cl Lint |
| Repository              | Status Pipeline Triggerer Commit  | Stages   |
| D Issues 0              | running     #4131     #2131 |  |
| 11 Merge Requests 0     |   |  |
| 🤗 CI / CD               |   |  |
| ≪ Collapse sidebar      |   |  |

You can view all pipelines by going to the Pipelines page in your project.



In GitLab CI, Runners run the code defined in **.gitlab-ci.yml.** GitLab CI not only executes the jobs you've set, but also shows you what's happening during execution, as you would see in your terminal. You can also view if artifacts were stored correctly using the build artifacts browser that is available from the build sidebar.

| I.U Projects ∨ Groups ∨ | Activity Milestones Snippets 🕢 – 🖿 🕄 – Search or jump to  | َ بِنَةٍ × 🕲 ۲ 🖞 ۲   |
|-------------------------|---|--|
| H helloworld            | Vaiting for pod gitlab-dynprov/runner-3bec7447-project-1816-concurrent-0jsktl to be running,  | build Retry  |
| 🔂 Project               | Naiting for pad gilab-dynprov/runner-3bec7447-project-1816-concurrent-0jsktl to be running,<br>status is Pending  | Duration: 32 seconds   |
| Repository              | Running on runner-3bec7447-project-1816-concurrent-0jsktl via shared-ci-gitlab-runner-<br>7d85664cdc-4gyff<br>* secho "Running on:S(gresp -Bl 'host LOCAL' /oroc/net/fib trie   head -nl   cut -df3)" | Timeout: 1h (from project)       Image: Comparison of the project of th |
| D Issues 0              | Running on: 10.65.15.191<br>Fetching changes with git depth set to 50   | Job artifacts  |
| Merge Requests          | Initialized empty Git repository in /builds/alash325/helloworld/.git/<br>Created fresh repository.<br>Checking out bioB0274 as master   | The artifacts will be removed in 6 days  |
| 🤗 CI / CD               | Skipping Git submodules setup   | Keep Download Browse   |
| Pipelines               | rrom nttps://gltlab.llu.se/alasn>z>/nelloworld<br>* [new branch] master -> origin/master<br>▼   | Commit b160b274  |
| Schedules               | \$ javac HelloWorld.java  | Initial CI set up  |
| Charts                  | > java Heilonoria<br>Hello, World   | Pipeline #4137 for master  |
| Security & Compliance   | <ul> <li>✓ Uploading artifacts</li> <li>./HelloWorld.*: found 2 matching files</li> </ul>   | build  |
| -                       | Uploading artifacts to coordinator ok id=10780 responseStatus=201 Created token-uyajt-3K  | →  |
| ≪ Collapse sidebar      | Job succeeded   |  |

## Part B – GitLab CI Configuration and Git Branches

In Part B, you have to work in pair and the project owner (In this case student-A) want the GitLab CI tool to continuously integrate, build and test his FreeCol project automatically triggered by push event, and publish the test results.

### 1. Setting up a shared Git repository

In this task, you need to import an existing FreeCol repository via HTTP by providing the Git URL from the New Project page:

| ILU Projects 🗸 Groups 🗸 More 🗸 🙆 🖌 🖿   |   | 🛢 🗸 Sear | rch or jump to   | ٩                                   | D)                        | n              | с G       | • | ÷. |
|--|---|----------|--|-------------------------------------|---------------------------|----------------|-----------|---|----|
| Welcome to GitLab<br>Code, test, and deploy together   | New project<br>New group<br>New snippet   |          |  |                                     |                           |                |           |   |    |
| Create a project<br>Projects are where you store y<br>issues, wiki and other features  | rour code, access<br>: of GitLab.         |          | <b>Create a grou</b><br>Groups are the be<br>members.          | <b>ip</b><br>est way to             | o mana                    | age proj       | jects anc | 1 |    |
| Explore public projects           There are 121 public projects           projects are an easy way to all           https://gitlab.liu.se/projects/new | on this server. Public<br>low everyone to |          | <b>Learn more a</b><br>Take a look at the<br>of GitLab's capab | <b>bout G</b><br>docume<br>ilities. | <b>iitLab</b><br>entatior | )<br>n to disc | cover all |   |    |

• Sign in on <a href="https://gitlab.liu.se/">https://gitlab.liu.se/</a> using "Student\_A" LiU account

- From your GitLab dashboard click New project
- Switch to the Import project tab
- Click on the Repo by URL button
- Fill <u>https://gitlab.liu.se/jesji387/group3.git</u> in the "Git repository URL". Use Public Checkbox. By Default, it is private.
- Click Create project to begin the import process

| New project   | Blank project Create from temp  | late Import project CI/CD for external repo   |  |  |  |  |  |
|---|---|---|--|--|--|--|--|
| A project is where you house your files<br>repository), plan your work (issues),  | buin project  |   |  |  |  |  |  |
| and publish your documentation<br>wiki), among other things.  | Import project from   |   |  |  |  |  |  |
| All features are enabled for blank  | G Google Code   | e 🕷 Fogbugz 🕶 Gitea git Repo by URL   |  |  |  |  |  |
| projects, from templates, or when<br>mporting, but you can disable them<br>afterward in the project settings.                 | 🗈 Manifest file   |   |  |  |  |  |  |
| o only use CI/CD features for an<br>external repository, choose <b>CI/CD for</b>  | Git repository URL  |   |  |  |  |  |  |
| external repo.  | https://gitlab.ida.liu.se/tddb84/freecol.git  |   |  |  |  |  |  |
| nformation about additional Pages<br>templates and how to install them can<br>be found in our Pages getting started<br>guide. | Username (optional) Password (optional)   |   |  |  |  |  |  |
| Fip: You can also create a project from   |   |   |  |  |  |  |  |
| the command line. Show command  | The repository must be accessible over ht   | tp://, https:// or git://.  |  |  |  |  |  |
|   | <ul> <li>If your HTTP repository is not publicly acce</li> <li>The import will time out after 180 minutes.</li> </ul> | 25 ible, add your credentials.<br>Is For repositories that take longer, use a clone/push combination. |  |  |  |  |  |
|   | To import an SVN repository, check out this   | is document.  |  |  |  |  |  |
|   | Once imported, repositories can be mirrore  | ed over SSH. Read more here.  |  |  |  |  |  |
|   | Mirror repository   |   |  |  |  |  |  |
|   | Automatically update this project's branches and tag  | gs from the upstream repository every hour.   |  |  |  |  |  |
|   | Project name  |   |  |  |  |  |  |
|   | My awesome project  |   |  |  |  |  |  |
|   | Project URL   | Project slug  |  |  |  |  |  |
|   | https://gitlab.liu.se/alash325/   |   |  |  |  |  |  |
|   | Want to house several dependent projects under the same namesnare? Create a group                                     |   |  |  |  |  |  |
|   | Project description (optional)  |   |  |  |  |  |  |
|   | Course Course   |   |  |  |  |  |  |
|   | Description format  |   |  |  |  |  |  |
|   | Visibility Level 📀  |   |  |  |  |  |  |
|   | 🗇 🔒 Private   |   |  |  |  |  |  |
|   | Project access must be granted explicitly to  | each user.  |  |  |  |  |  |
|   | Internal  | in user.  |  |  |  |  |  |
|   | The project can be accessed by any logged   |   |  |  |  |  |  |
|   | The project can be accessed by any logged   |   |  |  |  |  |  |
|   | The project can be accessed by any logged<br>Public<br>The project can be accessed without any au                     | ithentication.  |  |  |  |  |  |
|   | The project can be accessed by any logged   Public The project can be accessed without any au                         | ithentication.  |  |  |  |  |  |
|   | The project can be accessed by any logged  Create project   | uthentication.<br>Cancel  |  |  |  |  |  |

Once completed, you will be redirected to your newly created FreeCol project accessible at https://gitlab.liu.se/<student\_A\_LiU\_ID>/freecol, which will be your original remote repository.

Your task is now to configure CI for FreeCol project using built-in GitLab CI/CD and propose changes to the original repository (i.e., to the GitLab repository reside in student A). You are expected to set up the following stages: build, test, and publish HTML test report page in GitLab. The second requirement for this lab is to create a new branch with git whenever you want to add a new work and push it to the remote/original repository via create merge request.

## 2. Forking the FreeCol repository

A fork is a copy of original repository. Forking a repository allows you to freely experiment with changes without affecting the original project. In order to push to the original repository, you will push to your own forked repository and create a merge request from your forked repository in GitLab.

- Sign in on <a href="https://gitlab.liu.se/">https://gitlab.liu.se/</a> using "Student\_B" LiU account
- Navigate to the FreeCol repository of student\_A in your browser (<u>https://gitlab.liu.se/<student\_A\_LiU\_ID>/freecol</u>)
- In the top-right corner of the navigated page, click Fork
- Select your namespace to fork the project

Now if you look at the URL of your forked repository it is changed into <<u>student\_B\_LiU\_ID</u>>:

#### https://gitlab.liu.se/<student B LiU ID>/freecol

It also shows you where your repository is forked from (see Figure below).

| Projects V Groups V | Activity Milestones Snippets 🕢 🕨 📑 🗸   | Search or jump to ९ 🛈 🐧    | ≌ @ × ∰ ×     |
|---------------------|--|----------------------------|---------------|
| F freecol           | Alachew Mengist > freecol > Details  |                            |               |
| 🔂 Project           | F freecol ₽  | Q ∨ ☆ Star 0 % Fe          | ork 0 Clone 🗸 |
| Details             | Project ID: 1900   |                            |               |
| Activity            | 💀 Add license 🗢 8 Commits 🤌 1 Branch 🥏 1 Tag 🕒 0 Bytes Files                 |                            |               |
| Releases            | Forked from Alachew Mengist / freecol  |                            |               |
| Cycle Analytics     |  |                            |               |
| Insights            | master v freecol / + v   | History <b>Q</b> Find file | Web IDE 🛛 🖓 🖌 |
| Repository          | Reversed order of parameters to fix test<br>Ola Leifler authored 2 years ago |                            | a1df9888      |
|                     |  |                            |               |

## 3. Cloning the repository

Log in on the lab terminal as **Student\_B**.

To start working locally, you must clone (download) a copy of the original repository files (available in Student\_A GitLab repository) to <Student\_B> local computer. You can clone it via HTTPS:

| Projects 🗸 Groups 🗸 | Activity Milestones Snippets 🕢 🗸 🖿   | 🕈 🗸 Search or ju        | imp to Q       | D) 11        | ≝ <b>0</b> ~ | ∰~      |
|---------------------|--|-------------------------|----------------|--------------|--------------|---------|
| F freecol           | Alachew Mengist > freecol > Details  |                         |                |              |              |         |
| 🔂 Project           | F freecol A  |                         | û v ☆ Star     | 0 % Fo       | rk 0         | Clone 🗸 |
| Details             | Project ID: 1059 Leave project   |                         | Clone with SSI | H            |              |         |
| Activity            | Mo license. All rights reserved 8 Commits V 1 Branch                         | 7 1 Tag 🖸 0 Bytes Files | git@gitlab.    | liu.se:alame | e60/freec    | G       |
| Releases            | master v freecol / + v   |                         | Clone with HT  | TPS          |              |         |
| Cycle Analytics     |  |                         | https://git    | lab.liu.se/a | lame60/f     | G       |
| Insights            | Reversed order of parameters to fix test<br>Ola Leifler authored 2 years ago |                         |                |              | a1df9888     | G       |
| ≪ Collapse sidebar  |  |                         |                |              |              |         |

• Go to <u>https://gitlab.liu.se/<student A LiU ID>/freecol</u>

- Click on the **Clone** button
- Copy the URL from "Clone with HTTPS" field
- Open a terminal in the directory you wish to clone the repository files into and run the following command.

git clone [repository\_URL], where [repository\_URL] is the URL (original remote repository path) you copied.

After cloning the repository successfully, you should see the copy of files in your local git repository as shown below (see the directory where you cloned the repository).

|   | Na | me              | Date modified    | Туре           | Size  | ^     |
|---|----|-----------------|------------------|----------------|-------|-------|
|   |    | .git            | 2019-09-02 15:57 | File folder    |       |       |
|   | ø  | build           | 2019-09-02 15:57 | File folder    |       |       |
|   | Ø  | config          | 2019-09-02 15:57 | File folder    |       |       |
|   | Ø  | data            | 2019-09-02 15:57 | File folder    |       |       |
|   | Ø  | doc             | 2019-09-02 15:57 | File folder    |       |       |
|   | Ø  | jars            | 2019-09-02 15:57 | File folder    |       |       |
|   | Ø  | packaging       | 2019-09-02 15:57 | File folder    |       |       |
|   | Ø  | schema          | 2019-09-02 15:57 | File folder    |       |       |
| п | Ø  | src             | 2019-09-02 15:57 | File folder    |       |       |
|   | Ø  | test            | 2019-09-02 15:57 | File folder    |       |       |
| e | Ø  | unused          | 2019-09-02 15:57 | File folder    |       |       |
|   | Ø  | www.freecol.org | 2019-09-02 15:57 | File folder    |       |       |
|   | 0  | .classpath      | 2019-09-02 15:57 | CLASSPATH File | 1 KB  |       |
|   | 5  | .gitignore      | 2019-09-02 15:57 | Text Document  | 1 KB  |       |
|   |    | .project        | 2019-09-02 15:57 | PROJECT File   | 1 KB  |       |
|   | 1  | build           | 2019-09-02 15:57 | XML Document   | 44 KB | ~     |
|   |    |                 |                  |                |       | :== . |

Now add the forked repository as well:

• Run git remote add <Student\_B> <URL to Student\_B's fork>

**git remote -v** shoud now output the following lines: origin https://gitlab.liu.se/<Student\_A>/freecol.git (fetch) origin https://gitlab.liu.se/<Student\_A>/freecol.git (push)

<Student\_B> https://gitlab.liu.se/<Student\_B>/freecol.git (fetch)

<Student\_B> https://gitlab.liu.se/<Student\_B>/freecol.git (push)

### 4. Run build scripts in FreeCol in your local computer

As mentioned in the introduction, FreeCol application uses the **Apache Ant** build script. By default, **Ant** uses **build.xml** as the name for a buildfile. You can find the FreeCol buildfile in your local repository (see ./build.xml). It includes targets for building FreeCol, distribution packages, running tests, creating documentation etc.

Now try to build, run tests, and create documentation for FreeCol java application using the **Ant** build script:

- Open a terminal in the directory where your FreeCol project is copied and run the following command:
- ant build to build the FreeCol app
- ant -lib test/lib/junit.jar -Dtest=AllTests testall to run the JUnit tests and create a browsable HTML report (see the result index.html file under ./build/report/).

**NOTE**: if the "ant" command does not work. Skip to next stage. We have provided the docker image including "ant" below.

### 5. Set up Continuous Integration in GitLab

The workflows required to have working CI and pass this lab is summarized as follows:

- 1. Create a new branch
- 2. Add your code for CI configuration. You CI configuration should include the following:
  - Build the FeeCol application.
  - Run the FreeCol test suite.
  - Store the result as a build artifact (for the test suite).
  - Publish the test results with GitLab pages.
- 3. Push your changes to your forked repository.
- 4. Create merge request in GitLab to the original repository (i.e., to student A).
- 5. Merge the proposed changes in GitLab (from student\_A Gitlab repository).

You are free to work with your own starting point but If you have trouble finding a starting point, follow the following steps:

 Create a new branch called "CI\_<student\_B\_id>" on your local machine and switch in this branch:

#### git checkout -b <your\_new\_branch\_name> where <your\_new\_branch\_name> is "Cl\_<student\_B\_id>"

This will switch your local git repository to a new branch called **"CI\_<student\_B\_id>".** You can verify this by running **git branch** in your terminal. The output shall be:

\* CI \_<student\_id>

Master

2. Create a new file called **.gitlab-ci.yml**. The first line of yaml file i.e a Docker image to run your script written in .gitlab-ci.yml file is given below.

image: alash325/javaant:latest

- 3. Complete other parts using tutorials given in Part A and reference materials or your own material.
- 4. Once you are satisfied with you CI configuration, push your changes to your forked repository with **git push <Student\_B> CI\_<student\_b\_id>**.

On successful completion, GitLab CI automatically starts to execute the jobs you've set as shown below ( Go to <u>https://gitlab.liu.se/<student B LiU ID>/freecol</u> and see your Gitlab repository).

| I.∪ Projects ∨ Groups ∨ | Activity Milestones Snippets     | <b>@</b> ~ <b>⊾</b>     | 🛨 🗸 Sea      | arch or jump to | Q ()        | រា ៤            | <b>0</b> ~ 👯 | ð ~    |
|-------------------------|----------------------------------|-------------------------|--------------|-----------------|-------------|-----------------|--------------|--------|
| F freecol               | Alachew Mengist > freecol > Pipe | lines                   |              |                 |             |                 |              |        |
| ✿ Project               | All 1 Pending 0 Runnin           | g 1 Finished 0 Branches | Tags         | Rur             | Pipeline    | Clear Runner Ca | iches Cl     | Lint   |
| Repository              | Status Pipeline                  | Friggerer Commit        | Stages       |                 |             |                 |              |        |
| D) Issues 0             | #4138                            | PCI_alash325            | 018cec67     |                 |             |                 |              |        |
| 11 Merge Requests 0     | latest                           | 👾 🌐 Initial CI se       | t up         |                 |             |                 |              |        |
| 🤗 CI / CD               |                                  |                         |              |                 |             |                 |              |        |
| Pipelines               |                                  |                         |              |                 |             |                 |              |        |
| ≪ Collapse sidebar      |                                  |                         |              |                 |             |                 |              |        |
| Repository              | All 15 Pending 0 Runnin          | ng 0 Finished 6         |              | _               |             |                 | C            | 1 lint |
| () Issues 0             | Status Job                       | Pipe                    | line Stage   | Name Timi       | ng          | Coverage        |              |        |
| 11 Merge Requests 0     | (⊗ skipped) #10859 ¥             | master - #418           | 34 by deploy | pages           |             |                 |              |        |
| 🤗 CI / CD               | #10858 ¥                         | macter - #418           | 84 by        | <b>A</b> 0      | 0.00.24     |                 |              | _      |
| Pipelines               | S failed bc0c4445                |                         | test         | test 🛗 1        | 7 hours ago |                 |              | C      |
| Jobs                    | #10857 ¥                         | master -0- #418         | 34 by        | ō0              | 0:02:37     |                 | Ģ            |        |
| Schedules               | bc0c4445                         | (B)                     | build        | build 🛗 1       | 7 hours ago |                 | C            |        |

Wait until all the jobs (stages) are executed. Once completed, go to **CI/CD->Jobs** from your project page and see the result for all the jobs you were expected to complete (i.e., build, test, publish html pages). If the results are as shown below in the graph which can be interpreted as: The CI for build job is **passed**, test job is **failed**, and publish html page **skipped**, then go to **Step 4** otherwise try to fix your CI configuration until you reached the expected results. The reason for the test job failure is due to the sound mixer is not available in our computer system. However, for the GitLab runner to execute and publish html page for the test, you are expected to fix those tests otherwise GitLab runner skips to execute the deploy (publish html page) stage. You are going to do this in **step 8**.

### 6. Step 4. Create a merge request in GitLab to the original repository

Go to <a href="https://gitlab.liu.se/<student\_B\_LiU\_ID>/freecol">https://gitlab.liu.se/<student\_B\_LiU\_ID>/freecol</a>

| I.U Projects ~ Groups ~ Ad              | tivity Milestones Snippets 🕢 🗸 🖿   | Search or jump to | Q D)       | ង ខ        | 0 × 🌐 × 🕯 |
|---|--|-------------------|------------|------------|-----------|
| F freecol                               | Alachew Mengist > freecol > Details  |                   |            |            |           |
| ✿ Project                               | You pushed to CI_alash325 at Alachew Mengist / freecol 1 minute ago  | Create merg       | ge request |            |           |
| Details<br>Activity                     | F freecol @<br>Project ID: 1909  |                   | ar 0       | Fork 0     | Clone 🗸   |
| Releases<br>Cycle Analytics<br>Insights | Add license → 8 Commits      Y 2 Branches      O 1 Tag     1 M 41 KB Files     Forked from Alachew Mengist / freecol |                   |            |            |           |
| Repository                              | master ~ freecol / + ~   | History           | Q Find f   | ile Web ID | ₽         |
| ≪ Collapse sidebar                      | Reversed order of parameters to fix test   |                   |            | aldfor     | 2000 🕞    |

- Click on Create merge request button
- Click on Submit merge request button

#### 7. Merge the proposed changes

• Go to your original repository

#### https://gitlab.liu.se/<student A LiU ID>/freecol

• Go to the **Merge Requests** tab and open the merge request. Once you open the merge request you will be redirected to the page like the figure below.

| ∎.∪ Projects ∨ Groups ∨ | Activity Milestones Snippets 🕢 - 🖿 🕄 - Search or jump                                  | np to ۹ Dr 🏦 🕑 V 🍘 V                         | ^ |
|-------------------------|--|--|---|
| F freecol               | Open Opened 1 minute ago by 🙀 Alachew Mengist Edit Close merge request                 | st 🔻 To Do 🛛 Add a To Do 🔹                   |   |
| 🔂 Project               | Ci initial set up  | 0 Assignees Edit                             |   |
| Repository              |  | Milectopo Edit                               |   |
| D Issues 0              | Request to merge alash325:Ci_alash325 😚 into master Open in Web IDE Check out branch 🗣 | None   |   |
| Merge Requests          |  | Time tracking O<br>No estimate or time spent | 1 |
| ₽ CI/CD                 | Pipeline #4192 failed for 861d29ed on alash325:Ci_alash325                             | Labels Edit                                  |   |
| Security & Compliance   |  | None   |   |
| Packages                | Approve No approval required; you can still approve 🕜                                  | Lock merge request Edit                      |   |
| ል Snippets              | Test summary contained 2 failed test results out of 430 total tests     Collap         | apse 1 participant                           |   |
| Settings                | (f) test found 2 failed test results out of 430 total tests                            | 小  |   |
|                         | X Xine testSrund   | Notifications                                |   |
|                         | × New testClassic  | Reference: alame60/freecoll2                 |   |
|                         | () Merge   |  |   |
| ≪ Collapse sidebar      | 1 commit and 1 merge commit will be added to master. Modify merge commit               |  |   |
|                         | np, aut a CODEOWNERS to automatically aut approvers based of the paths and the types.  |  |   |
| Source branch           | CI_alash325 v  |  |   |
| Target branch           | master   |  |   |
|                         | Delete source branch when merge request is accepted.                                   |  |   |
|                         |  |  |   |
| Contribution            | Allow commits from members who can merge to the target branch. About this feature      |  |   |
| Submit merge request    |  | Cancel                                       |   |

• Click on Merge to merge the proposed changes to your repository

### 8. Fixing the test cases

As mentioned, the reason for the test job failure is due to the sound mixer is not available in our computer system. However, for the GitLab runner to execute and publish html page for the test, you are expected to fix those tests. In this lab, we will fix just by removing those tests. Instructions are given below.

- 1. Check out to the original repository i.e., git checkout master
- 2. Pulling changes from a remote repository to retrieve new work done by other people and combines your local changes with changes made by others

- 3. Create a new branch called "Fix\_tests" and checkout your new branch
- 4. Go to **AllTests.java** in ./test/src/net/sf/freecol/common/AllTests.java and comment the following line of code:

// suite.addTest(net.sf.freecol.common.sound.AllTests.suite());

5. Push your changes (See **Step 3 in Task 4**)

After pushing your changes successfully, GitLab CI automatically starts to execute the jobs you've set and you can view the status of the Pipelines.

| <b>ILU</b> Projects ~ Go | ups v | Activity  | Milestones S     | nippets 🛆 ~    | <b>b</b> (1   | 🛛 🛩 🛛 Search or Jun | ip to: C           | ) n c 0-            | · 🕸 -   |
|--------------------------|-------|-----------|------------------|----------------|---|---------------------|--------------------|---------------------|---------|
| F freecol                | Ĩ     | Alachew M | lengist - Treeco | · Pipelines    |   |                     |                    |                     |         |
| Project.                 |       | All 2     | Pending (0       | Running (g Fin | ished 🗐 Branches Tags   |                     | Run Pipeline       | Clear Runner Caches | Cl Lint |
| C Repository             |       | Status    | Pipeline         | Triggerer      | Commit  | Stages              |                    |                     |         |
| D Issues                 |       | (O passa  | e4107            | 赤              | Paaster ~ 43fe39fc  | 000                 | 0.00               | 108.28              | φ.*     |
| n Merge Requests         |       |           |                  | 1150           | · Jourio test not required  | 1.12.2.2.           |                    | Es alto             |         |
| ∉ CI/CD                  |       | ⊙ failed  | #4106            | *              | Peaster ↔ 920e5942<br>∰ Add new file  | <b>8</b>            | e 00<br>= 58 minut | 107.26<br>Is ago    | • • ¢   |
| Pipelines                |       |           |                  |                | 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - |                     |                    |                     |         |
| « Collapse sidebar       |       |           |                  |                |   |                     |                    |                     |         |

After successful Pipelines you can view your clickable HTML pages by going to your projects **Settings->Pages**.

You can view your HTML report by clicking the pages your found.

| ← → C △ ① Not secure                  | alash325.gitlab-   | pages.liu.se/freecol/ |        |         |         |        |              |         |  | ☆              | Incognito 👼              |  |
|---------------------------------------|--|-----------------------|--------|---------|---------|--------|--------------|---------|--|----------------|--------------------------|--|
| <u>Home</u>                           | Unit Test Re   | esults.               |        |         |         |        |              |         |  | Designed for u | use with <u>JUnit</u> an |  |
| Packages<br>net.sf.freecol<br>Summary |  |                       |        |         |         |        |              |         |  |                |                          |  |
|                                       | Tests  | Failures              | Errors | Skipped |         | 5      | Success rate |         |  | Time           |                          |  |
|                                       | 428  | Q                     | Q      | Q       | 100.00% |        |              |         |  | 15.264         |                          |  |
| Classes                               | Note: failures are anticipated and checked for with assertions while errors are unanticipated. |                       |        |         |         |        |              |         |  |                |                          |  |
| AllTests                              | Packages   |                       |        |         |         |        |              |         |  |                |                          |  |
|                                       | Name   |                       |        |         | Tests   | Errors | Failures     | Skipped | Time(s)  | Time Stamp     | Host                     |  |
|                                       | net.sf.freecol   |                       |        |         | 428     | 0      | 0            | 0       | 15.264 2019-09- runner-<br>15T13:27:18 rrKWF53<br>project-<br>2138-<br>concurre<br>0 |                |                          |  |
|                                       |  |                       |        |         |         |        |              |         |  |                |                          |  |

## 9. Merge test fix into the master branch

1. Create a merge request (See Step 5 in Task 4)

2. Merge the requested changes from your original repository (See Step 5 Task 4)



## Examination

When you are done with all tasks and ready to demonstrate your solutions, contact your assistant during a lab occasion. Be prepared to show the workflows described in Task 4 and answers to questions. The lab assistants may ask you to send in these answers for later evaluations.