Playrion Technical Test

Prerequisite

Duration: you have a maximum of 48 hours to send us your test.
Unity version to use: 2022.3.1f1

Step 1: Git

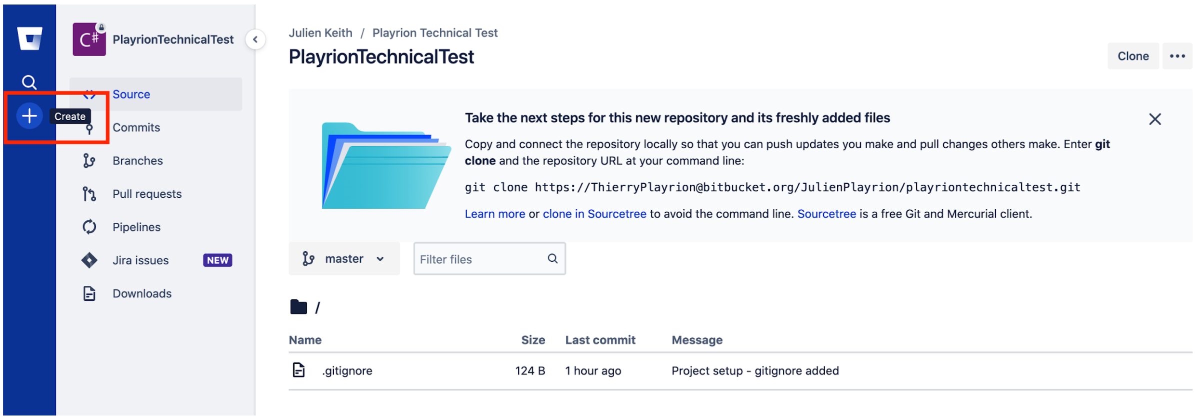
Estimated time: 20min

Statement
You need to fork the project from our git repository to a private git repository on your Bitbucket account.

Project Bitbucket link:
git@bitbucket.org:JulienPlayrion/playriontechnicaltest.git

You should have received a read-only invitation to this project on your Bitbucket-linked email address. If it's not the case, please contact us as you won't be able to go further with this test.

You are not evaluated on your Bitbucket knowledge, so here is how to fork on Bitbucket. In this example, your name is “Thierry Destribats”. Start by clicking on the top left “+” sign.



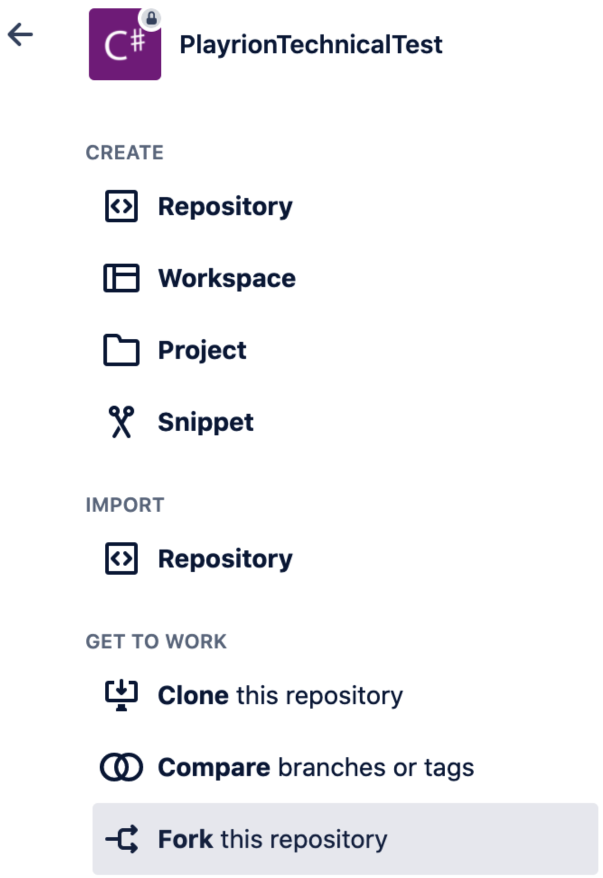
Take the next steps for this new repository and its freshly added files

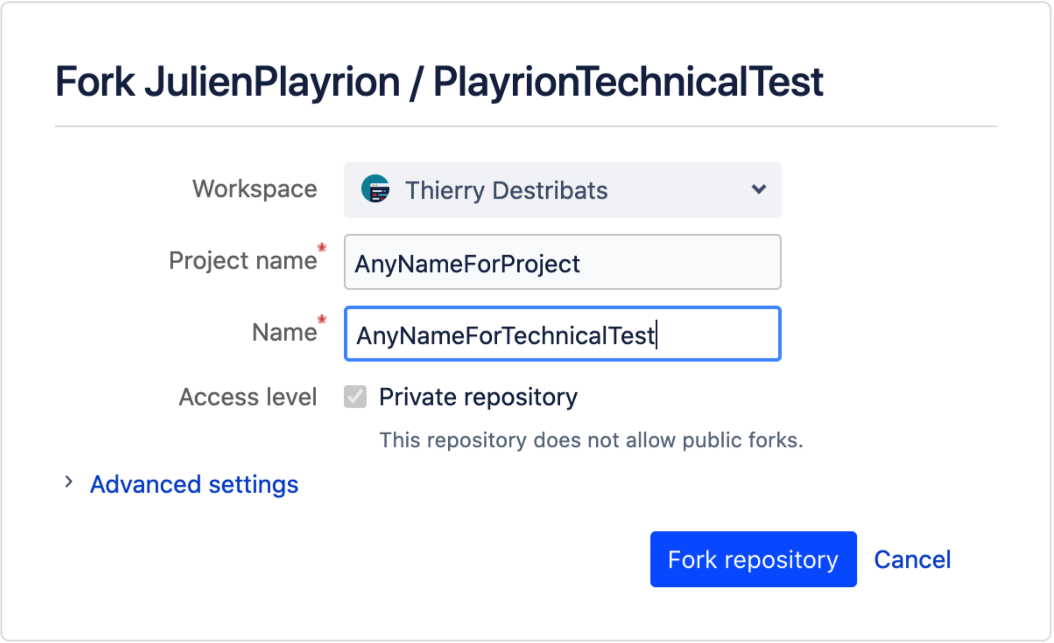
Copy and connect the repository locally so that you can push updates you make and pull changes others make. Enter **git clone** and the repository URL at your command line:

```
git clone https://ThierryPlayrion@bitbucket.org/JulienPlayrion/playriontechnicaltest.git
```

Learn more or clone in Sourcetree to avoid the command line. Sourcetree is a free Git and Mercurial client.

Name	Size	Last commit	Message
.gitignore	124 B	1 hour ago	Project setup - gitignore added




Fork JulienPlayrion / PlayrionTechnicalTest

Workspace: Thierry Destribats

Project name*: AnyNameForProject

Name*: AnyNameForTechnicalTest

Access level: ☑ Private repository

This repository does not allow public forks.

Then, clone the newly forked git repository to your computer. You will use this git repository to commit & push your test. Just use it the way you usually use a git repository.

By the end of the test, you will give us access to this git repository.

Step 2: Exercise 1

Estimated time: 2h

Constraint: don’t modify existing scripts.

Statement
The level is a train station. You need to code the AI of a late passenger who needs to board its train before the train leaves the station.

For this, the passenger needs to purchase a ticket at a vending machine and then go to the train.

Each vending machine has a different ticket treatment speed. Once the passenger reaches a machine, he needs to ask for a ticket, and the ticket will be delivered after some time. This time depends on the machine’s speed. You can’t modify vending machines' speeds.

The door between the station and the platform should not open if the passenger has no ticket. It should open if the passenger has a ticket.

The game starts at 15:01.
The train leaves at 15:05. We should see the train leaving the station.
Time must pass at actual speed, and needs to be displayed in the UI.

The passenger speed is 1.0f. You can’t modify this value.
If you want to set an acceleration to the passenger, then this acceleration is 2.6f. You can’t modify this value.

Once the train has left, a final (and basic) UI screen should appear, indicating if the passenger boarded its train or not.

Of course, you can’t trick the exercise and modify vending machines or walls’s positions.

You can use any tool or library you wish to.

Step 3: New git branch

Create a new git branch in order to complete the second exercise.

We will review Exercise 1 on your first branch, then Scene 2 on your second branch. Therefore you don’t need to worry if your Scene 1 code does not work on Scene 2 and vice versa. We won’t test this.

Step 3: Exercise 2

Estimated time: 3h

Constraint: don't modify existing scripts.

Statement:
You will work on the same scene as the previous exercise (scene 2 is not used within this test).

Now the game needs to be partially controlled by a backend.

When the game starts, vending machines' speed must be updated accordingly to values set in your backend. The values will be (in hierarchical scene order): 0.1, 0.6, 0.9, 0.2, 0.25, 0.25, 0.25, 0.4.

Also, your passenger can now have 3 types of behaviors : lazy, normal, hurried. This behavior is set when the game starts, and the value is set within the backend.
If the passenger is lazy, its speed will be slowed down by half.
If the passenger is normal, then nothing changes.
If the passenger is hurried, its speed will be multiplied by 1.5.

Finally, you must provide a backend interface where we can freeze / unfreeze the passenger's walk at any given time.
This feature has no impact on the passenger's machine choice.

Of course, you still can't trick the exercise by editing vending machines or walls's positions.

You can use any tool or library you wish to.

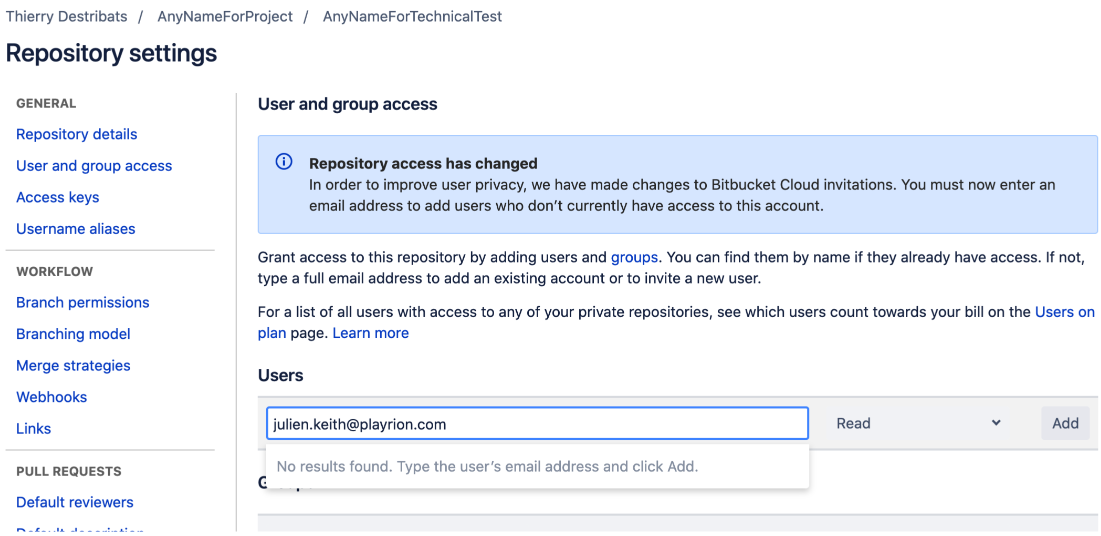
Step 4: Explain us how to run your test

Your project must contain a "Readme.md" or "Readme.txt" file that indicates step by step how we can run your backend environment, and how to use it.

If you used any third party tool that needs credentials, then give us the credentials so we can properly review your exercises.

Step 5: Send us the test

Once you have completed the test, add julien.keith@playrion.com to the users of your git repository. This will allow us to review the test.


Thierry Destribats / AnyNameForProject / AnyNameForTechnicalTest

Repository settings

User and group access

Repository access has changed
In order to improve user privacy, we have made changes to Bitbucket Cloud invitations. You must now enter an email address to add users who don't currently have access to this account.

Grant access to this repository by adding users and groups. You can find them by name if they already have access. If not, type a full email address to add an existing account or to invite a new user.

For a list of all users with access to any of your private repositories, see which users count towards your bill on the Users on plan page. Learn more

Users

julien.keith@playrion.com

No results found. Type the user's email address and click Add.

Please, be sure that you have a maximum of 4 users on your account (if you are on Bitbucket free plan, which grants a maximum of 5 users) so we can access your test.

You can view all your users here: https://bitbucket.org/account/user/<username>/plan-users/ with <username> being your actual username.

Also, send a quick mail to julien.keith@playrion.com, indicating us that you completed the test and gave us the access, and the name of your 2 branches and their meaning (example: “My branch named “Scene1” is the one that you need to checkout in order to review the first scene”).

The subject of the mail should be “[Firstname Lastname] Playrion Technical Test”.