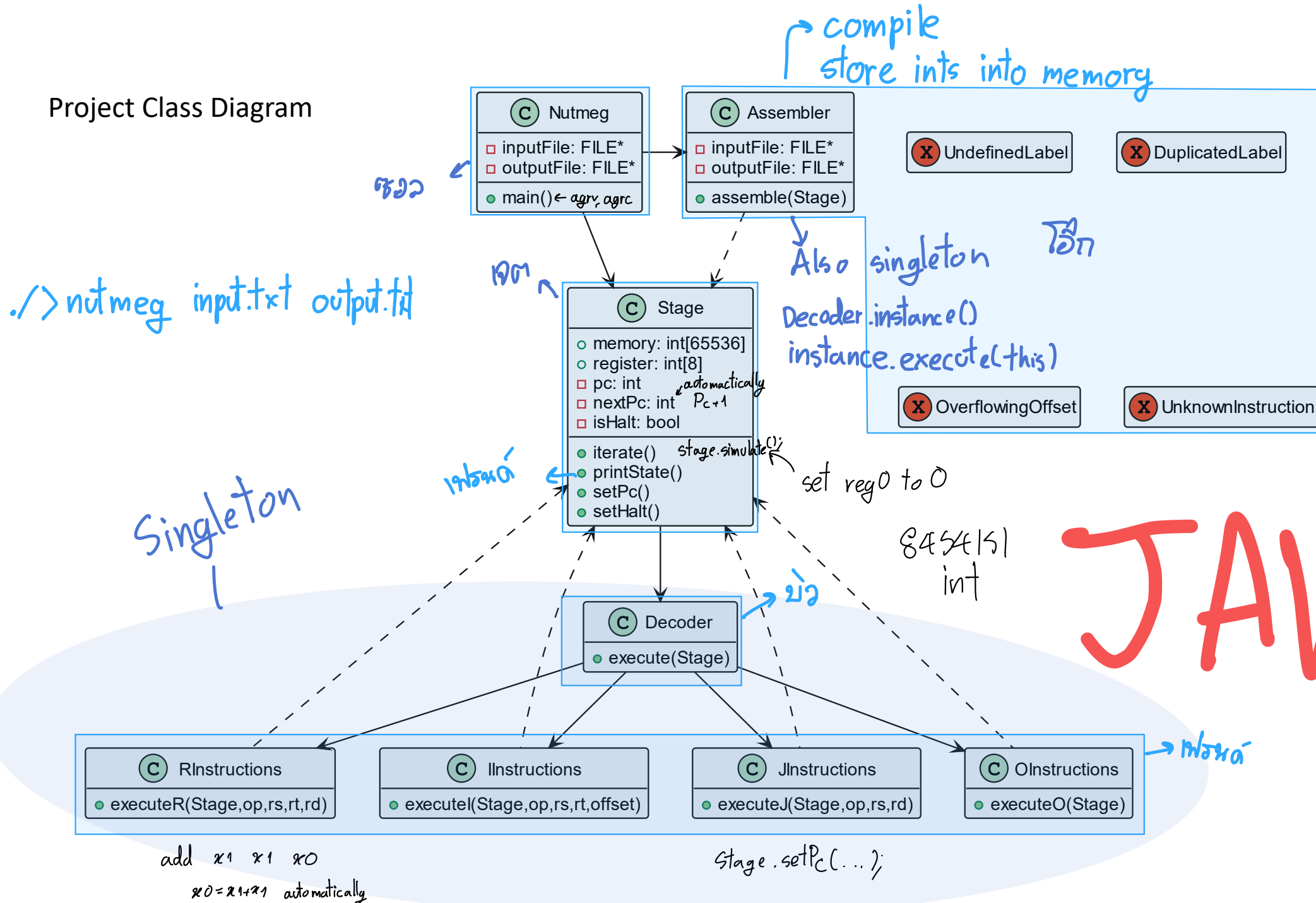
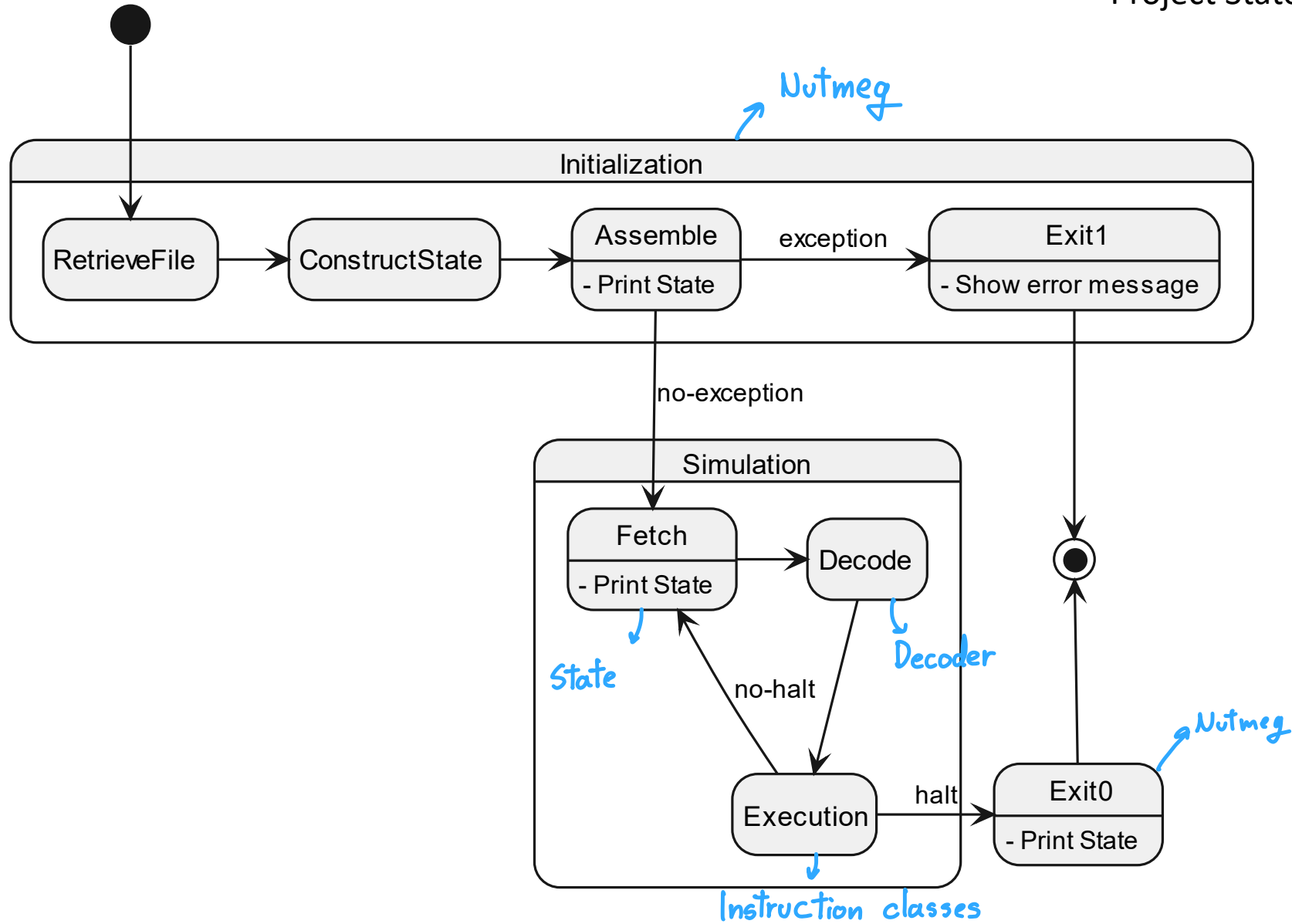


Project Class Diagram



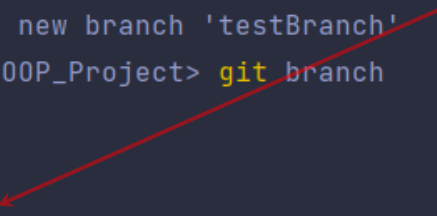
Project State Diagram



Pull Request

1. Before any coding, create a branch.

```
PS E:\261200\00P_Project> git branch
main
* master
PS E:\261200\00P_Project> git checkout -b testBranch
Switched to a new branch 'testBranch'
PS E:\261200\00P_Project> git branch
main
master
* testBranch
```



2. After coding, commit as customary.

```
PS E:\261200\00P_Project> git commit -m "demonstration message"
[testBranch b5ce50f] demonstration message
3 files changed, 113 deletions(-)
delete mode 100644 src/main/out/production/main/orchestrator/plan1.txt
delete mode 100644 src/main/out/production/main/orchestrator/plan2.txt
delete mode 100644 src/main/out/production/main/orchestrator/plan3.txt
```

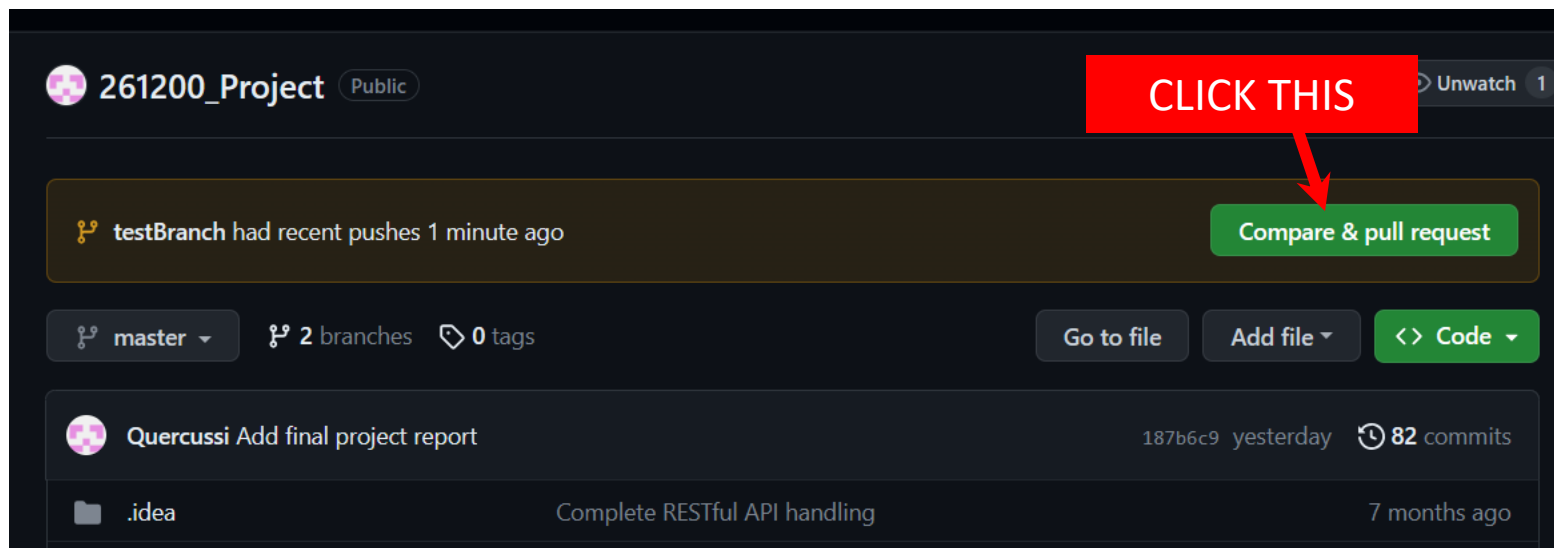
Pull Request (continue)

3. When ready to “push”, do as follows.

Do this once.
For next pushes, use
"git push" like usual

```
PS E:\261200\00P_Project> git push --set-upstream origin testBranch
Enumerating objects: 793, done.
Delta compression using up to 8 threads
Compressing objects: 100% (742/742), done.
Writing objects: 100% (793/793), 259.76 KiB | 565.00 KiB/s, done.
Total 793 (delta 430), reused 0 (delta 0), pack-reused 0
```

Then the repository will show that a branch was pushed.



Pull Request (continue)

4. Create pull request


↕

base: master

←

compare: testBranch










✓ Able to merge. These branches can be automatically merged.




Test branch

Write

Preview

H B I       @   

This is a test.

Attach files by dragging & dropping, selecting or pasting them. 

Create pull request

▼

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Pull Request (continue)

5. Merge pull request (**DON'T DO THIS TO YOUR OWN PULL REQUEST**)

The screenshot shows a GitHub Pull Request for 'Test branch #28'. At the top, it says 'Quercussi wants to merge 2 commits into master from testBranch'. Below this, there are tabs for 'Conversation' (0), 'Commits' (2), 'Checks' (0), and 'Files changed' (6). The 'Conversation' tab is active, showing a comment from 'Quercussi' that says 'This is a test.' Below the comment, it shows that 'Quercussi added 2 commits 17 minutes ago'. The commits are 'demonstration message' (b5ce50f) and 'demonstration message 2' (6c24b3b). At the bottom, there is a green box with a checkmark icon and the text 'This branch has no conflicts with the base branch. Merging can be performed automatically.' Below this, there is a green button labeled 'Merge pull request' with a dropdown arrow. A red arrow points to this button. To the right of the button, it says 'You can also [open this in GitHub Desktop](#) or view [command line instructions](#).'

Test branch #28

Open Quercussi wants to merge 2 commits into master from testBranch

Conversation 0 Commits 2 Checks 0 Files changed 6

Quercussi commented 1 minute ago

This is a test.

Quercussi added 2 commits 17 minutes ago

- demonstration message b5ce50f
- demonstration message 2 6c24b3b

Add more commits by pushing to the testBranch branch on Quercussi/261200_Project

✓ This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

In case of using C++

For every class, create both header file (.h) and implementation file (.cpp). For example,

```
1  #pragma once
2
3  class VertexBuffer {
4  private:
5      unsigned int m_rendererID;
6  public:
7      VertexBuffer(const void* data, unsigned int& size);
8      ~VertexBuffer();
9
10     void bind() const;
11     void unbind() const;
12 };
```

Header file
(VertexBuffer.h)

```
1  #include "VertexBuffer.h"
2  #include "Renderer.h"
3
4  VertexBuffer::VertexBuffer(const void* data, unsigned int& size) {
5      GLCall(glGenBuffers(1, &m_rendererID));
6      GLCall(glBindBuffer(GL_ARRAY_BUFFER, m_rendererID));
7      GLCall(glBufferData(GL_ARRAY_BUFFER, size, data, GL_DYNAMIC_DRAW));
8  }
9
10 VertexBuffer::~VertexBuffer() {
11     GLCall(glDeleteBuffers(1, &m_rendererID));
12 }
13
14 void VertexBuffer::bind() const {
15     GLCall(glBindBuffer(GL_ARRAY_BUFFER, m_rendererID));
16 }
17
18 void VertexBuffer::unbind() const {
19     GLCall(glBindBuffer(GL_ARRAY_BUFFER, 0));
20 }
```

Implementation file
(VertexBuffer.cpp)

First deadline:

Wednesday

27/9/2023