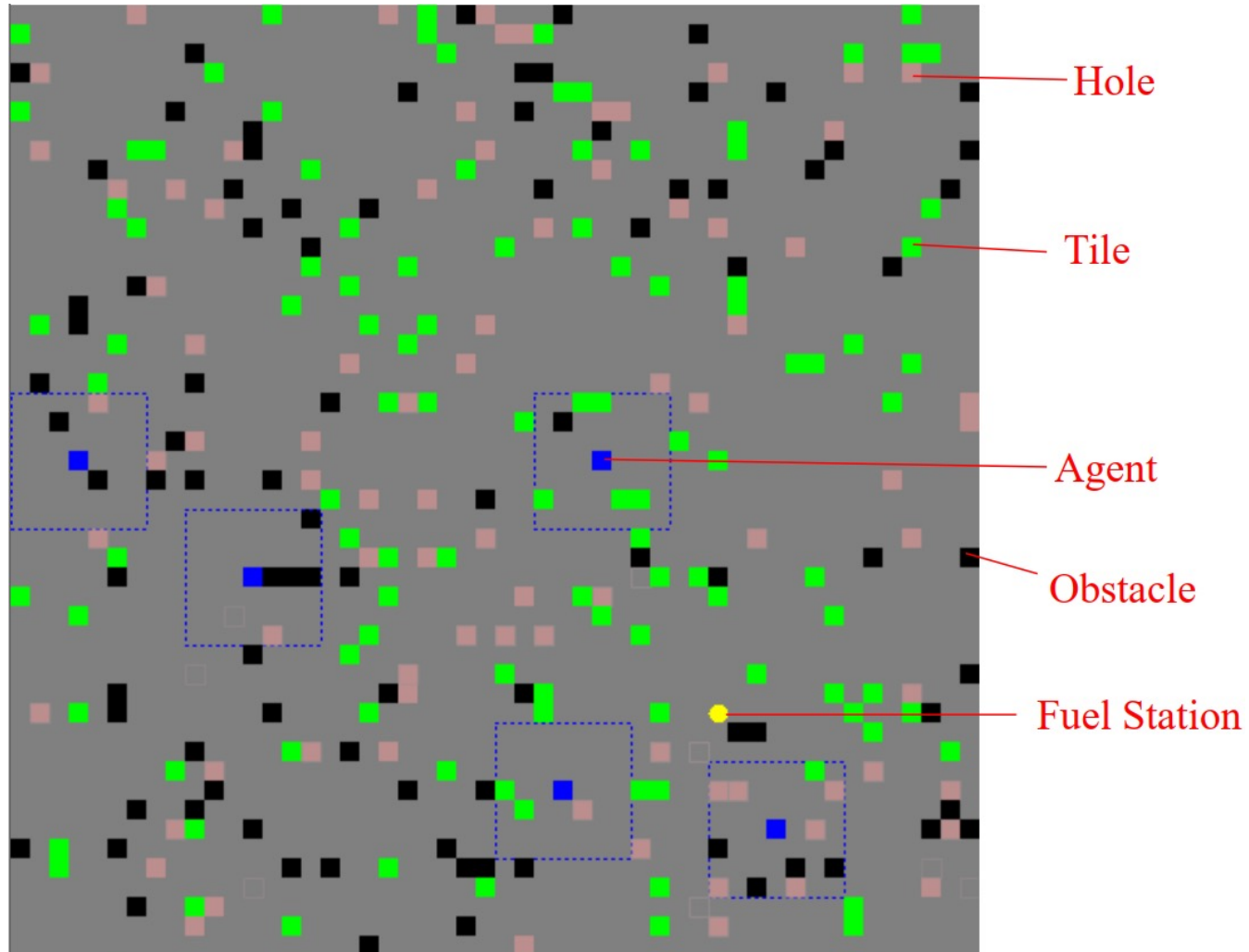

Build Intelligent Agents for Tileworld Environment

Chen Shuo (schen@ntu.edu.sg)

AI6125: Multi-agent System

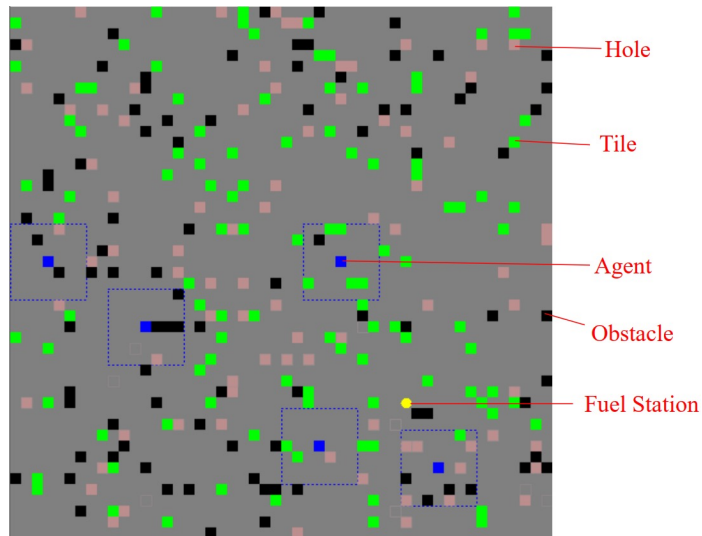
Tileworld



Tileworld

► Agent:

- Move up/down/left/right, consume one fuel
- Pickup tile (can carry 3 tiles at most)
- Drop, i.e. put tile in the hole → get 1 reward
- Refuel from fuel station due to limited fuel
- Has limited visibility



Obstacles/tiles/holes appear randomly and exist for a certain period

Fuel station has a fixed position that is randomly generated at the beginning. This position is unknown to agents.

Agent Implementation

- ▶ Objective: get as many rewards as possible within a preset steps of simulation.
- ▶ In every step, an agent goes through the cycle:
Sense-Communicate (optional)-Plan-Act

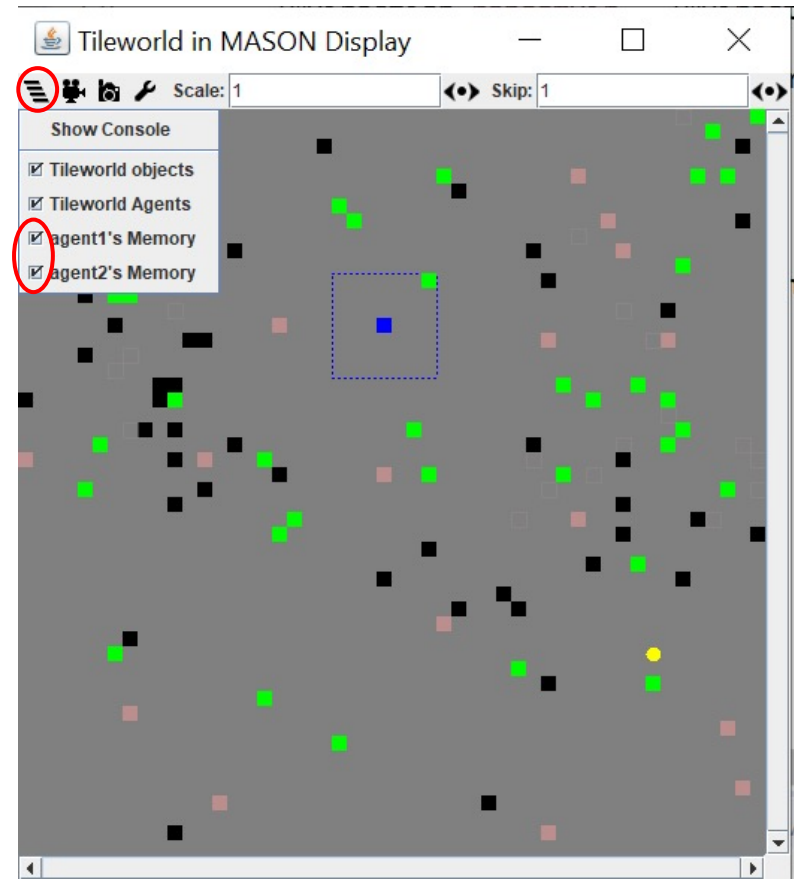
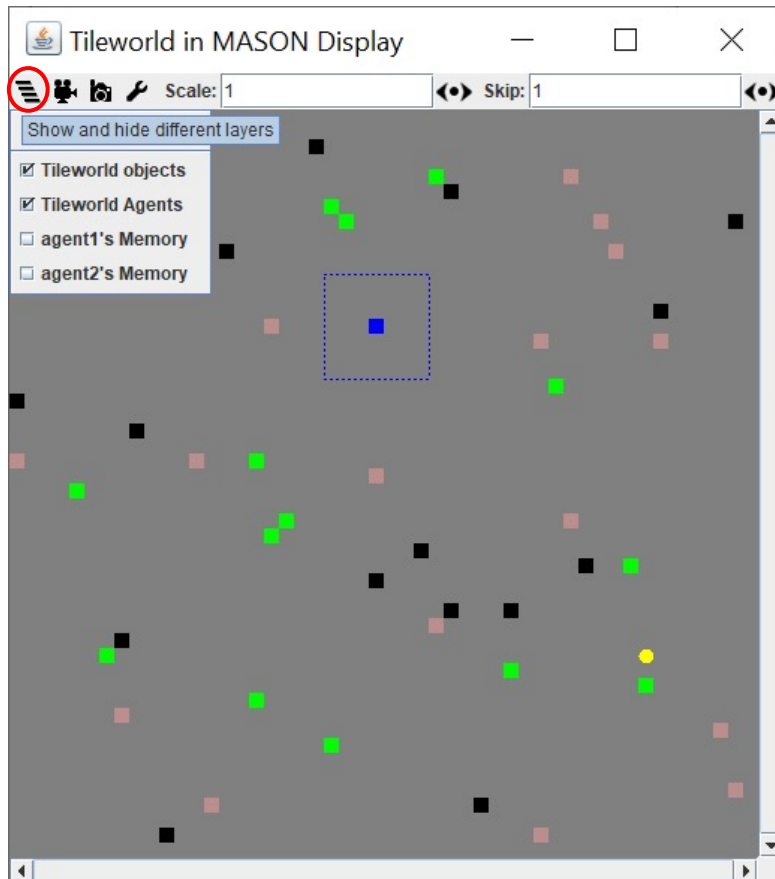
Agent Implementation

▶ Three modules to be implemented:

- Planning module: after sensing/communication, the agent's memory is updated. The agent plans its action accordingly.
- Memory module (optional): agents store environmental information in their memory. By default, we use `TWAgentWorkingMemory` module. You may extend “`TWAgentWorkingMemory`” class and implement your own module.
- Communication module (optional): in every step, each agent can broadcast a message encoded by “`Message`” class. You can extend “`Message`” class and use your own design to encode information.

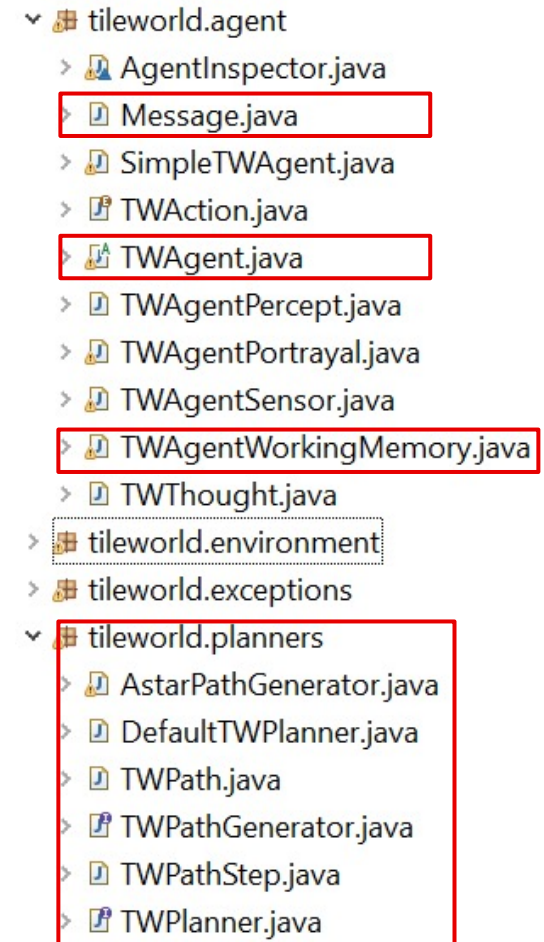
Memory is critical for planning

- Use GUI to check memory for implementation/debug



Agent Implementation

- ▶ To implement your own agent, you should
 - Extend “TWAgent” class
 - Create your own planner/message/memory
 - Override “communicate()”, “think()” and “act()” methods in “TWAgent” class



Agent Implementation

▶ You should NOT

- Override other methods of “TWAgent” class
- Modify “environment” package of the simulator
- Call “increaseReward()” method of “TWEnvironment” class in any methods other than the “putTileInHole()” method of “TWAgent” class

Assessment

- ▶ Report (maximum 2000 words and 8 pages including images/plots)
- ▶ Team based 15~20 minute presentation
- ▶ Competition
 - Each team has 3~4 agents
 - Your agent class will be used in “TWEnvironment” class:

```
//Now we create some agents
Int2D pos = this.generateRandomLocation();
createAgent(new SimpleTWAgent("agent1", pos.getX(), pos.getY(), this, Parameters.defaultFuelLevel));
pos = this.generateRandomLocation();
createAgent(new SimpleTWAgent("agent2", pos.getX(), pos.getY(), this, Parameters.defaultFuelLevel));
```

Competition

- **Configuration One:**

- Environment Size: 50×50 cells
- Average Object Creation Rate: $\text{NormalDistribution}(\mu=0.2, \sigma=0.05)$
- Lifetime: 100

- **Configuration Two:**

- Environment Size: 80×80 cells
- Average Object Creation Rate: $\text{NormalDistribution}(\mu=2, \sigma=0.5)$
- Lifetime: 30

- **Configuration Three:**

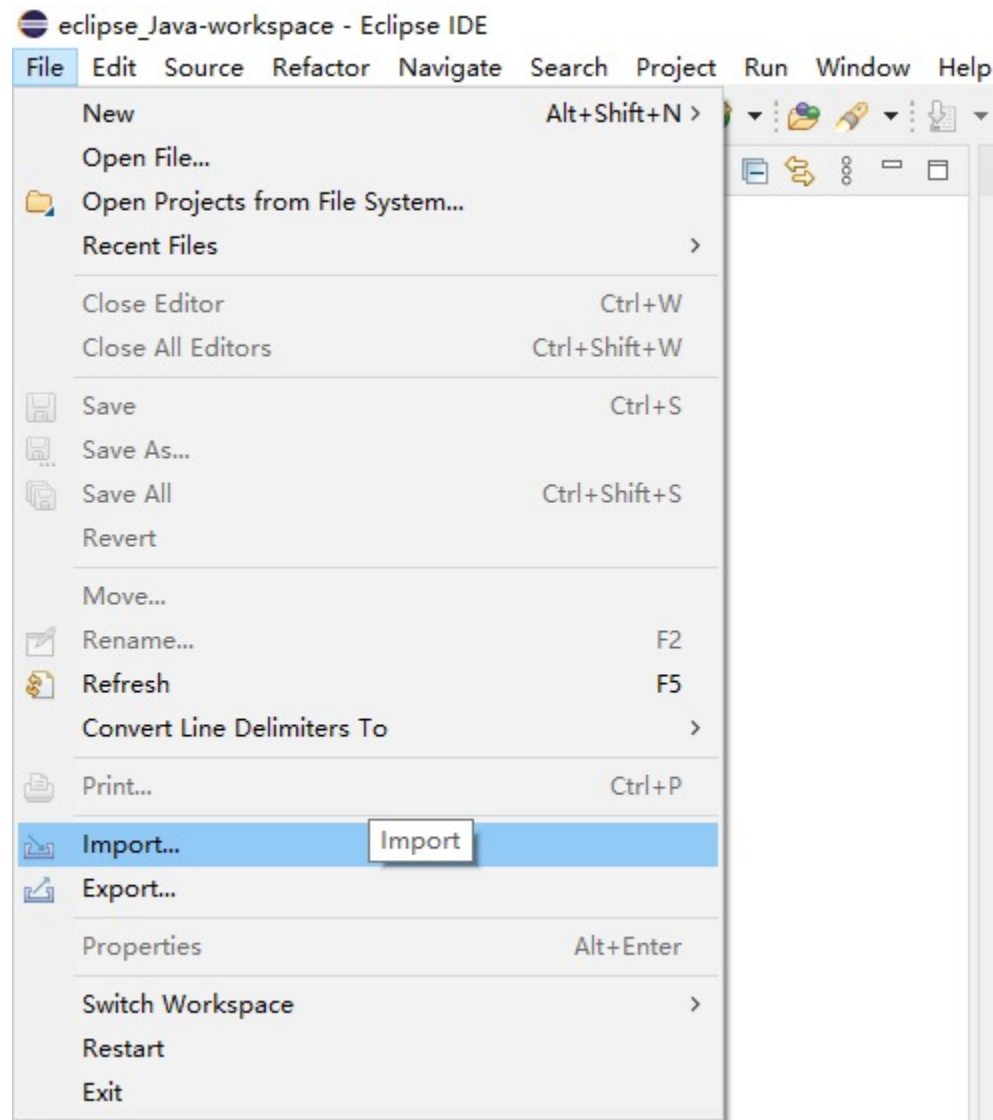
- Environment Size: $? \times ?$ cells
- Average Object Creation Rate: $?(\mu=?, \sigma=?)$
- Lifetime: ?

- ▶ Each experiment runs 5000 steps
- ▶ Each agent has 500 fuel at the beginning
- ▶ Ten experiments for each configuration (average reward)

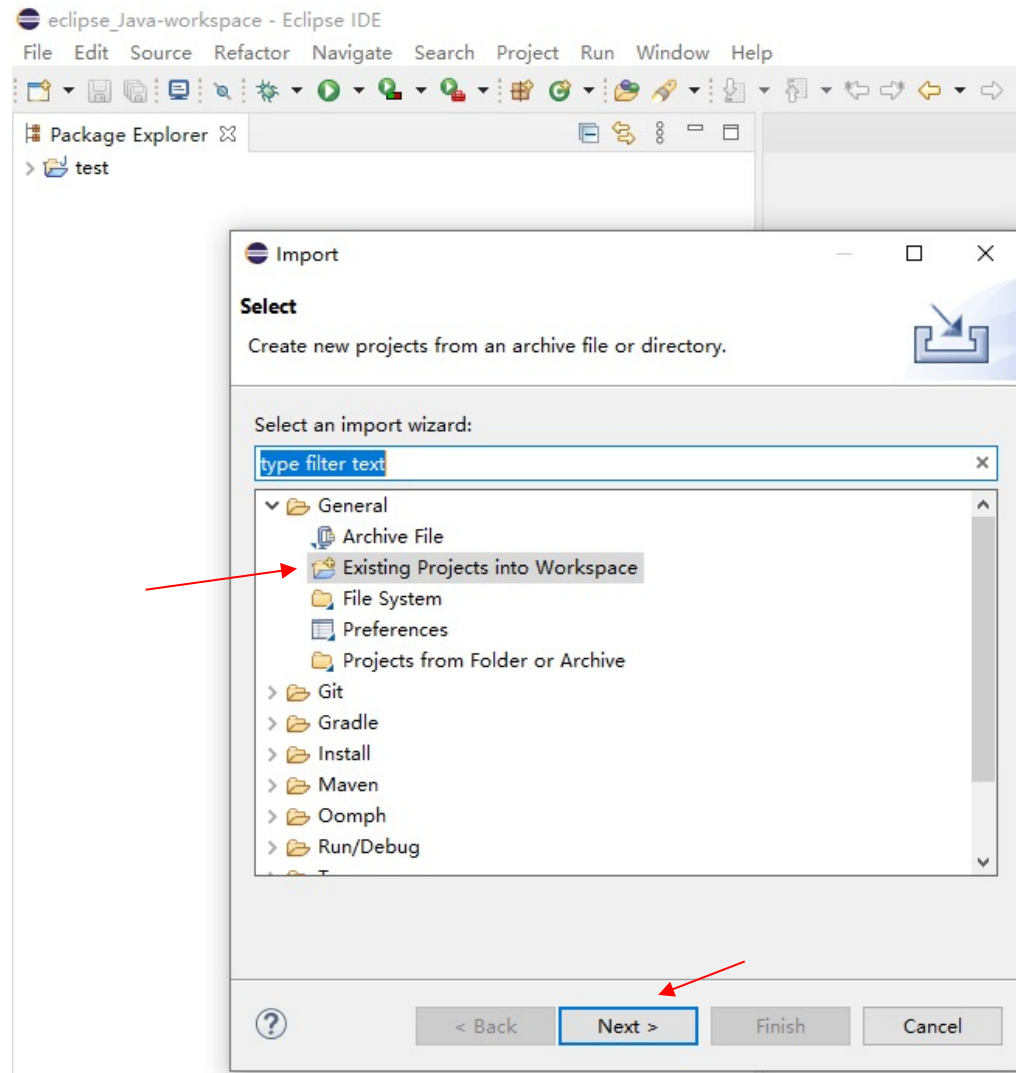
Installation of Tileworld

- ▶ Read group-project.pdf file in NTULearn for more details
- ▶ Download source file (Tileworld.zip) from NTULearn
- ▶ Install Java JDK and the JDK version should be 1.8
 - <https://www.oracle.com/java/technologies/javase/javase8u211-later-archive-downloads.html>
- ▶ Install the Java3D library (version 1.5)
 - Library file link: <https://www.oracle.com/java/technologies/javase/java-3d.html>
 - Installation guide: <https://download.java.net/media/java3d/builds/release/1.5.1/README-download.html>
- ▶ Download the MASON_14.jar file from NTULearn and use it as an external library for compiling and running
- ▶ Download Eclipse IDE for Java Developers from:
<https://www.eclipse.org/downloads/packages/release/2021-06/r>

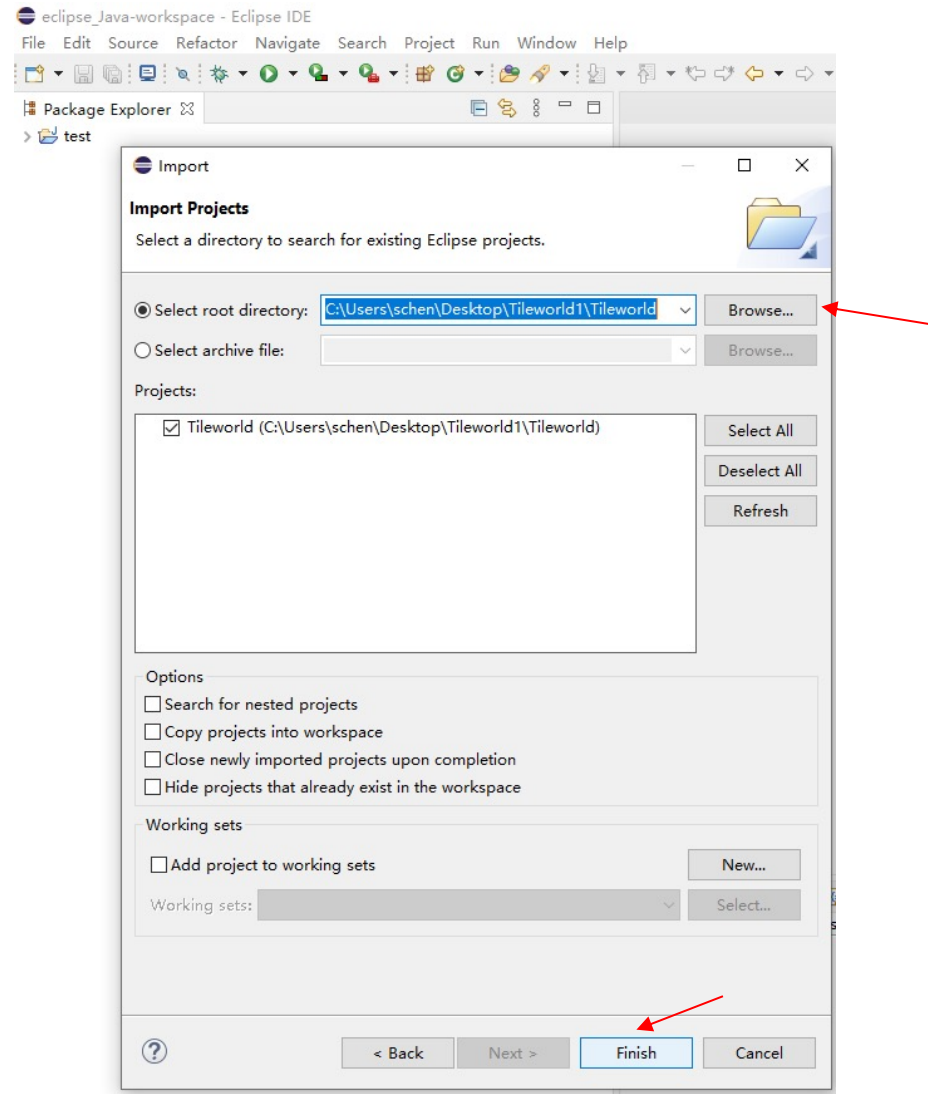
Import Project



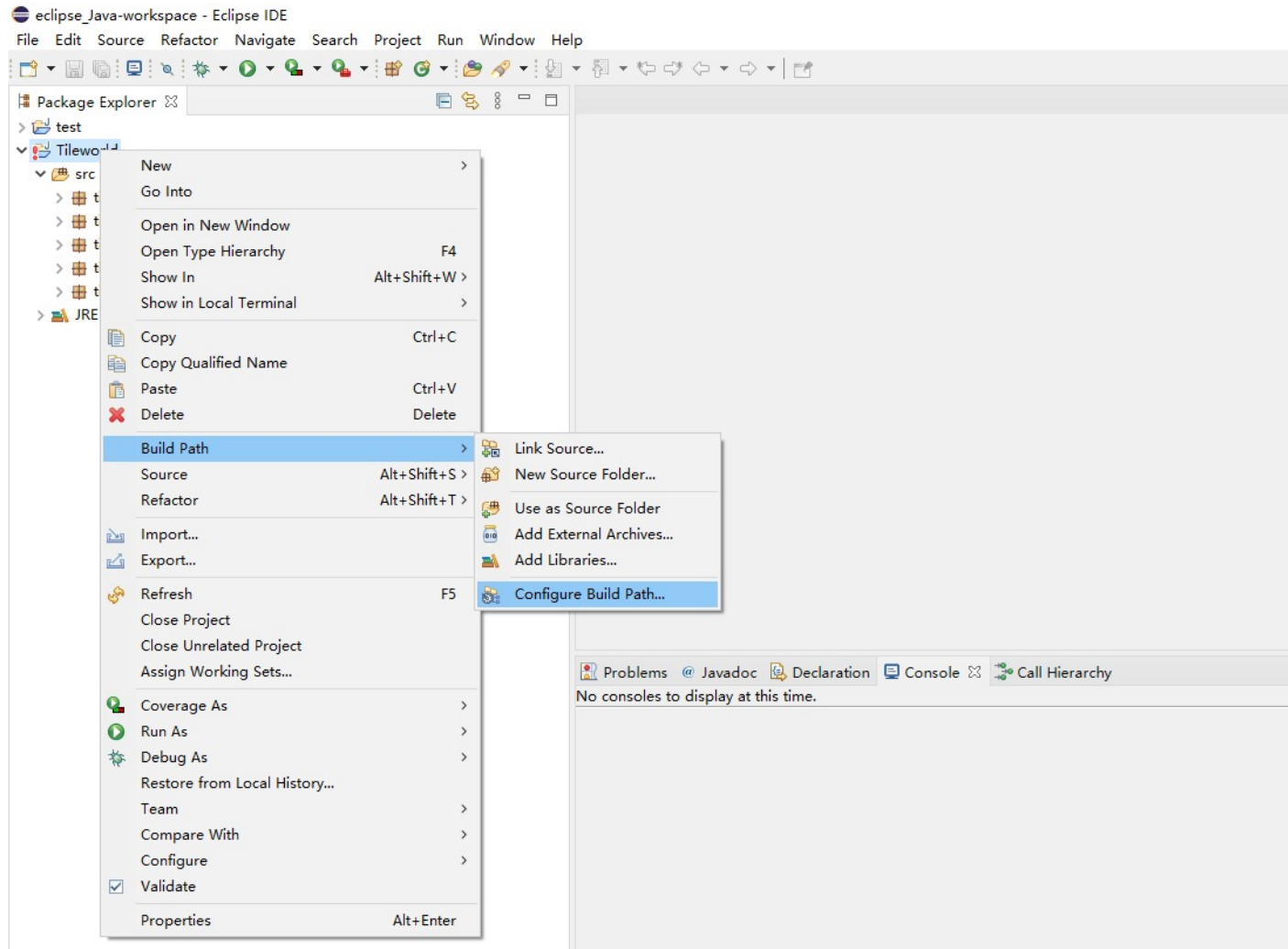
Import Project



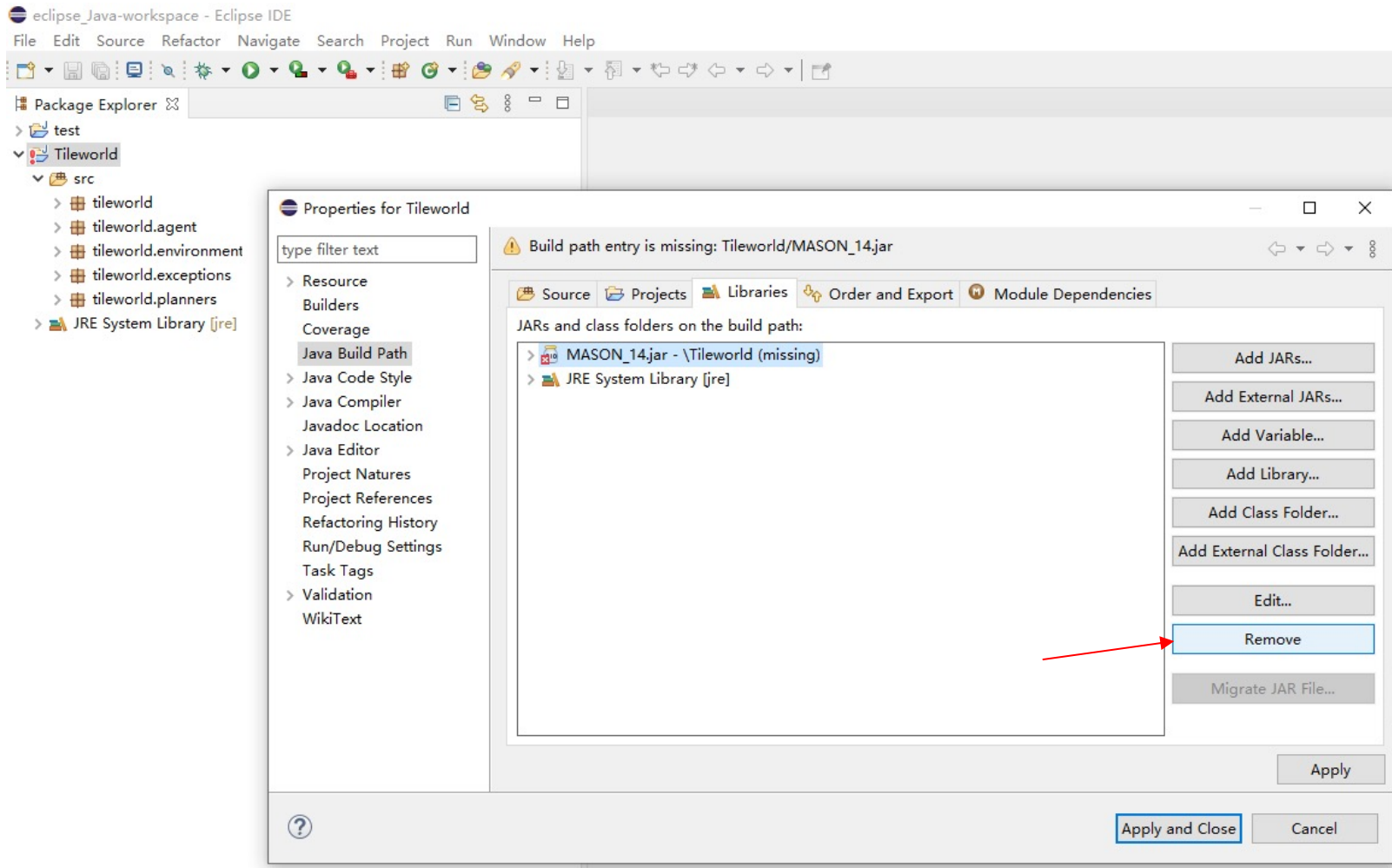
Import Project



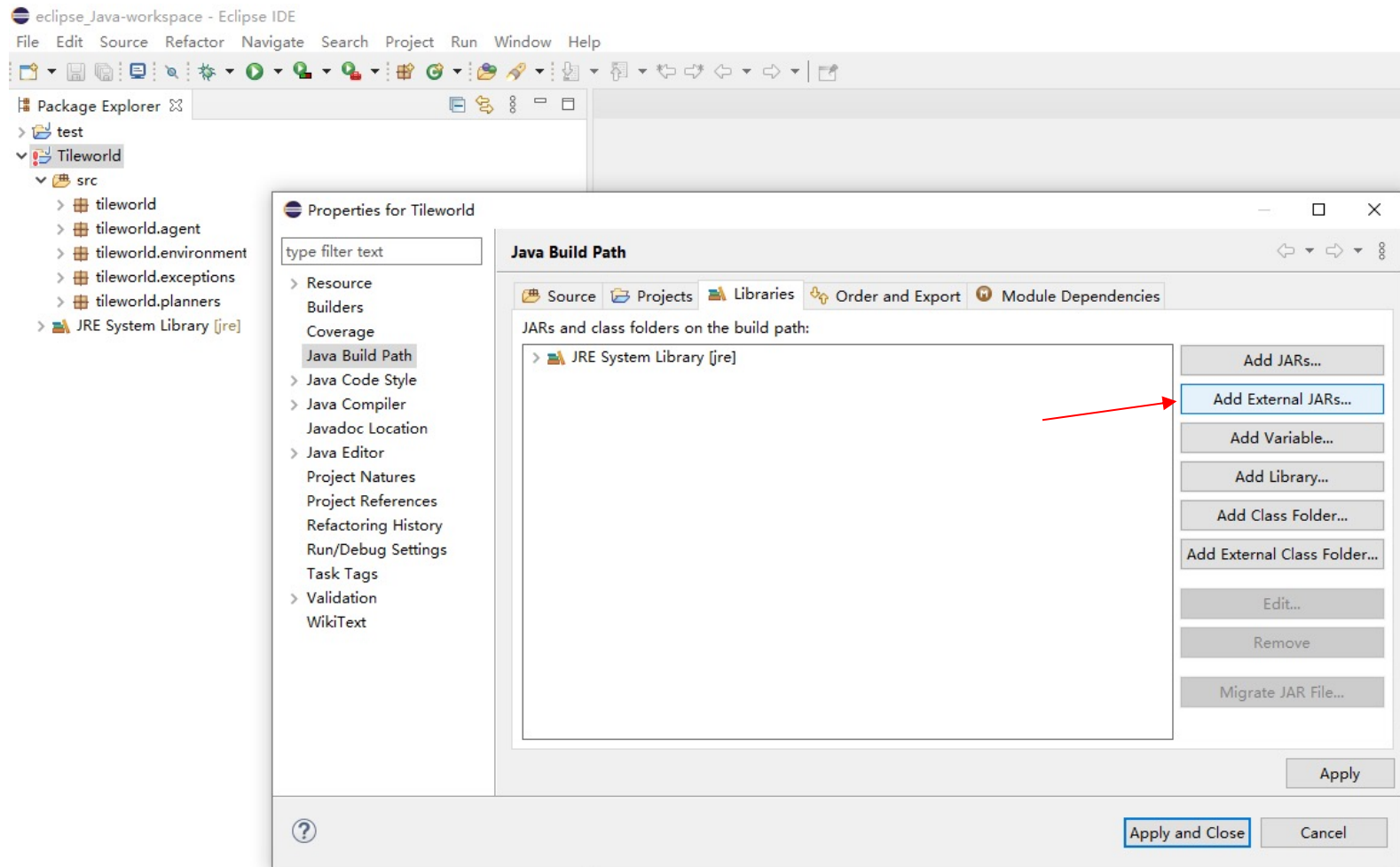
Configuration of MASON External Library



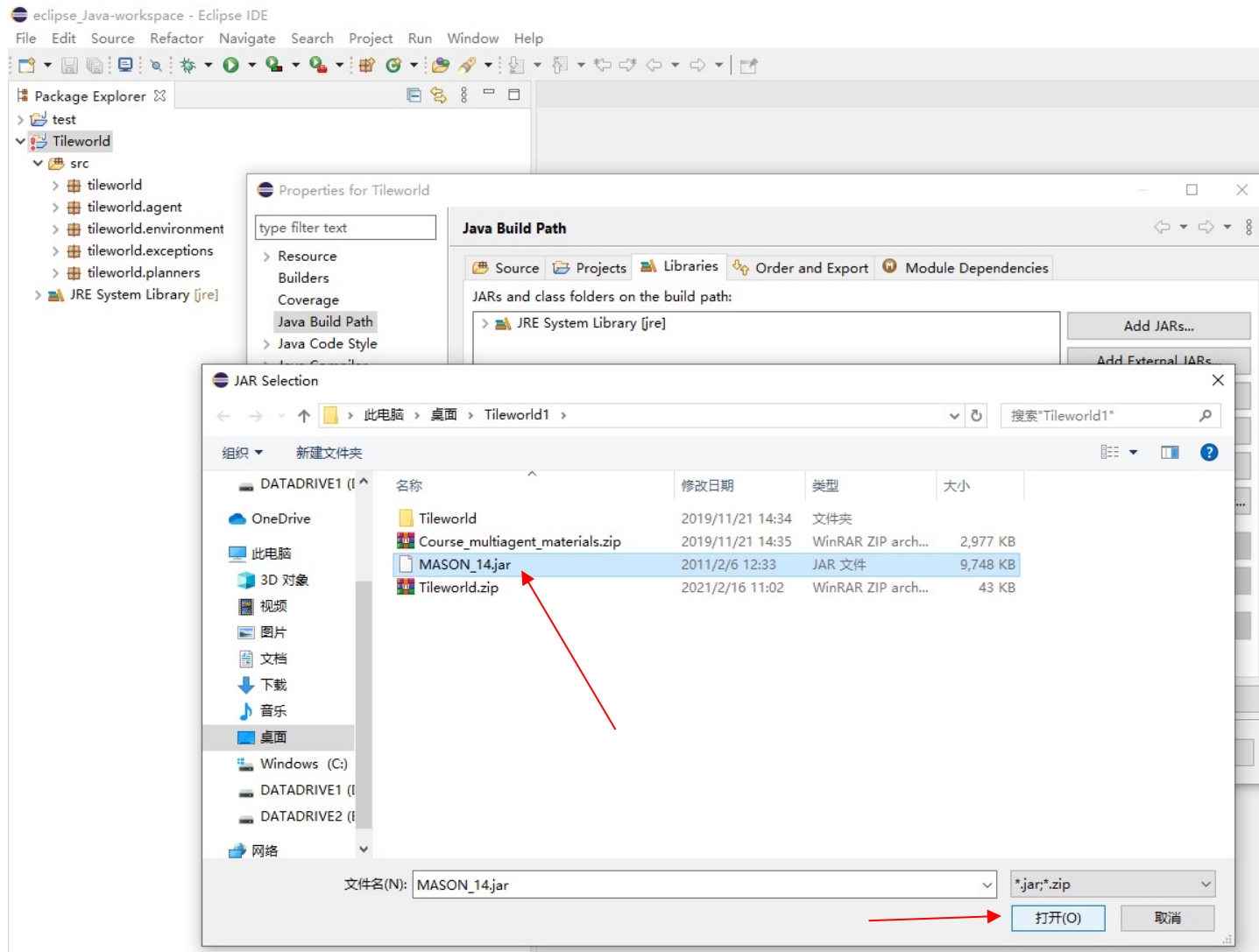
Configuration of MASON External Library



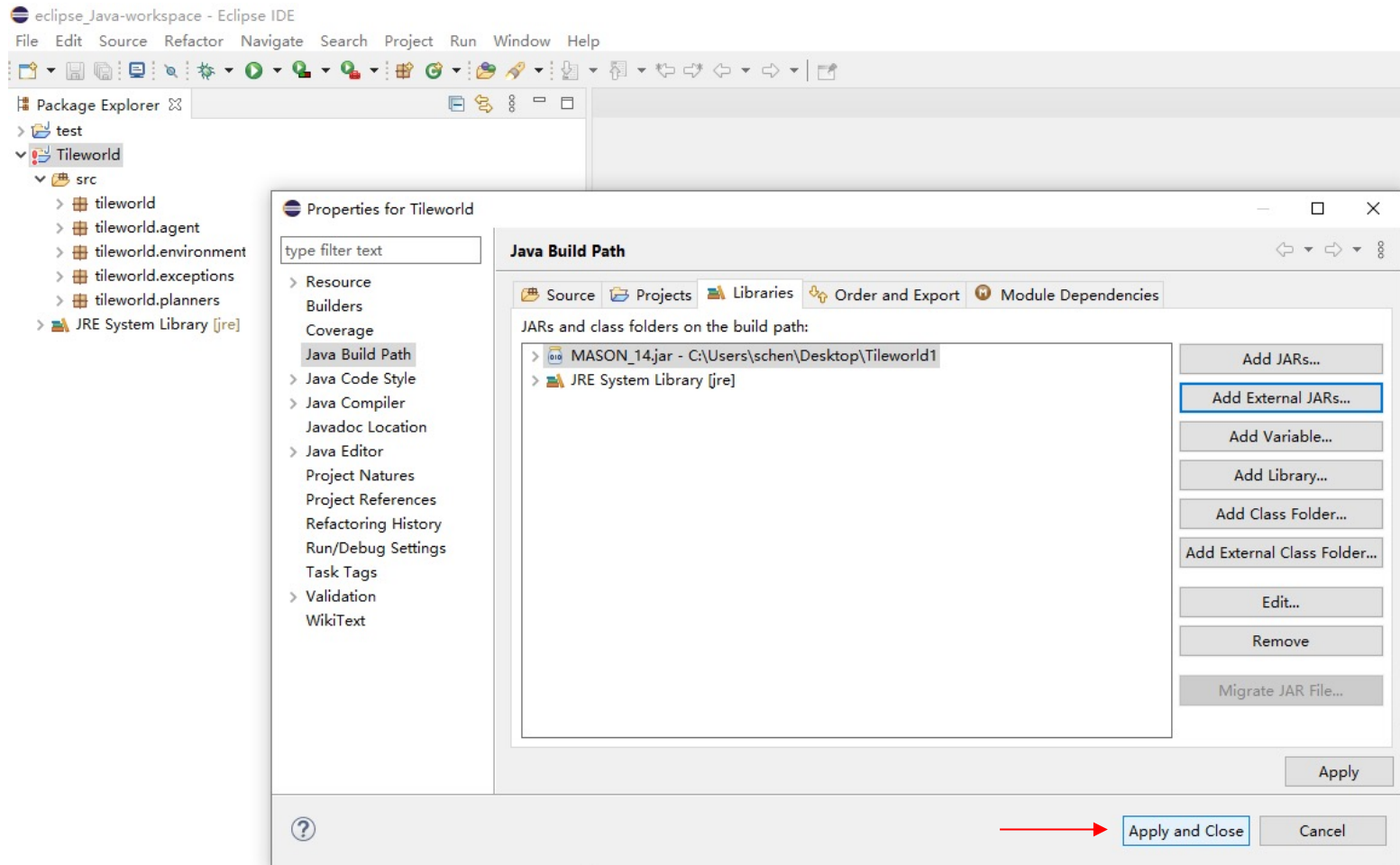
Configuration of MASON External Library



Configuration of MASON External Library



Configuration of MASON External Library



Run Experiment

