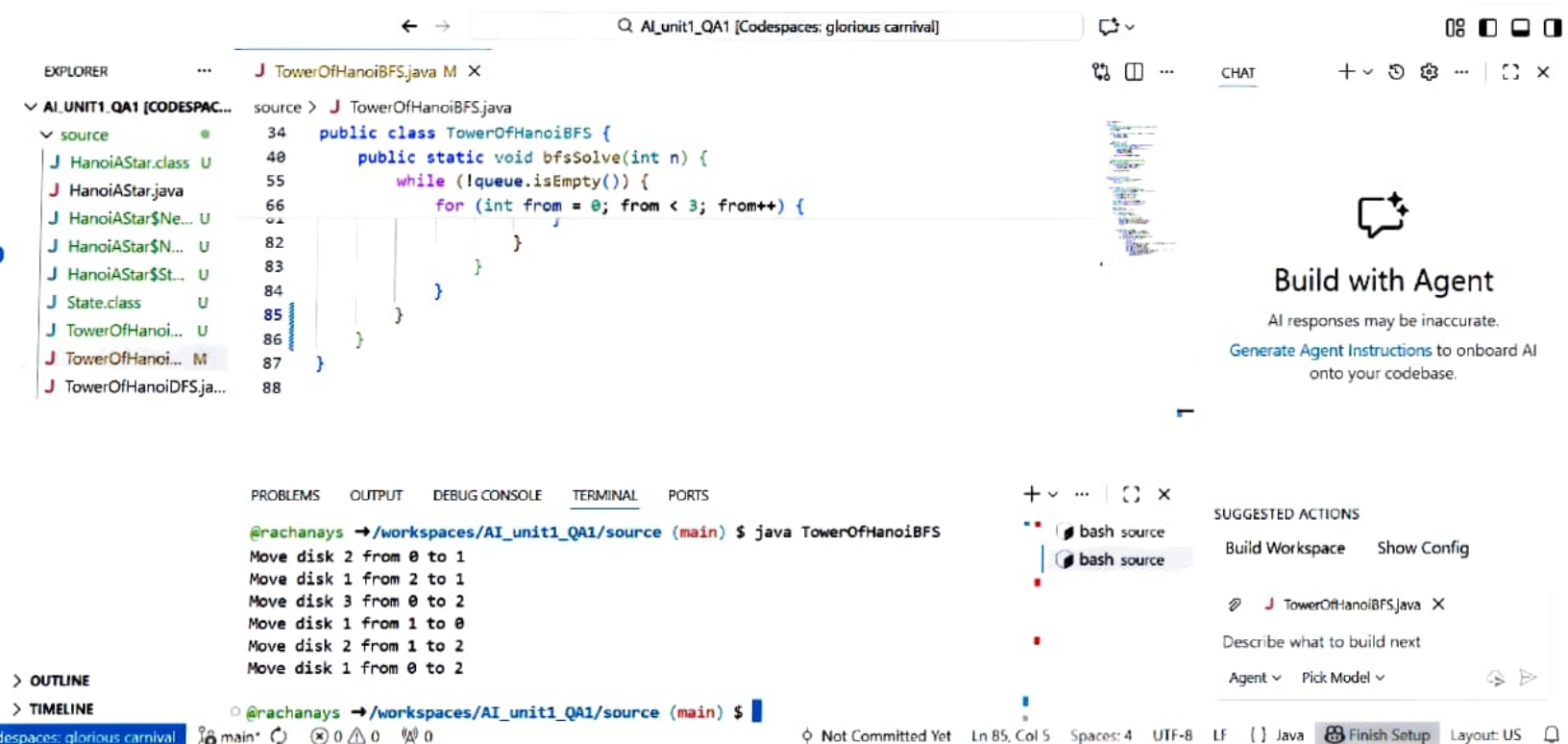
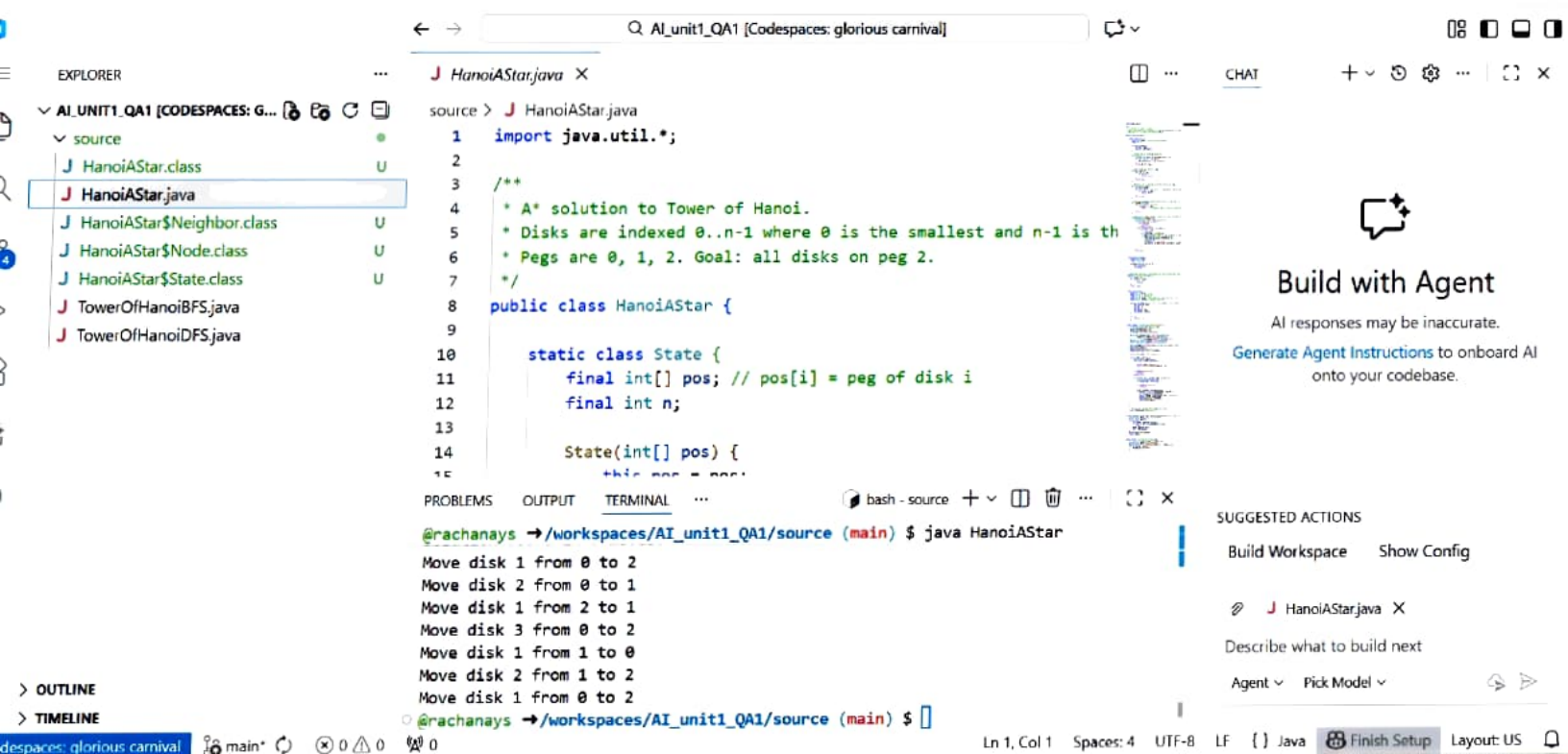

AI_unit1_QA1 [Codespaces: glorious carnival]
EXPLORER
TowerOfHanoiBFS.java M
AI_UNIT1_QA1 [CODESPAC...
source
HanoiAStar.class U
HanoiAStar.java
HanoiAStar$Ne... U
HanoiAStar$N... U
HanoiAStar$St... U
State.class U
TowerOfHanoi... U
TowerOfHanoi... M
TowerOfHanoiDFS.ja...
source > TowerOfHanoiBFS.java
34 public class TowerOfHanoiBFS {
40 public static void bfsSolve(int n) {
55 while (!queue.isEmpty()) {
66 for (int from = 0; from < 3; from++) {
82 }
83 }
84 }
85 }
86 }
87 }
88
CHAT
Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
@rachanays →/workspaces/AI_unit1_QA1/source (main) $ java TowerOfHanoiBFS
Move disk 2 from 0 to 1
Move disk 1 from 2 to 1
Move disk 3 from 0 to 2
Move disk 1 from 1 to 0
Move disk 2 from 1 to 2
Move disk 1 from 0 to 2
@rachanays →/workspaces/AI_unit1_QA1/source (main) $
bash source
bash source
SUGGESTED ACTIONS
Build Workspace
Show Config
TowerOfHanoiBFS.java
Describe what to build next
Agent
Pick Model
OUTLINE
TIMELINE
main*
Not Committed Yet
Ln 85, Col 5
Spaces: 4
UTF-8
LF
Java
Finish Setup
Layout: US


TowerOfHanoiDFS.java - AI_un
glorious-carnival-wvgv57pwqg7cv64q.github.dev
Install
AI_unit1_QA1 [Codespaces: glorious carnival]
EXPLORER
TowerOfHanoiBFS.java M
TowerOfHanoiDFS.java M
CHAT
AI_UNIT1_QA1 [CODESPAC...
source
HanoiAStar.class U
HanoiAStar.java
HanoiAStar$Ne... U
HanoiAStar$N... U
HanoiAStar$St... U
State.class U
TowerOfHanoi... U
TowerOfHanoi... M
TowerOfHanoi... U
TowerOfHanoi... M
source > TowerOfHanoiDFS.java
1 public class TowerOfHanoiDFS {
19 }
20
21 public static void main(String[] args)
22 {
23 int n = 3; // number of disks
24 System.out.println("Tower of Hanoi solution using DFS for " + n + " disk
25 solveHanoi(n, 'A', 'B', 'C'); // A=source, B=auxiliary, C=destination
26 }
27 }
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
@rachanays →/workspaces/AI_unit1_QA1/source (main) $ javac ./TowerOfHanoiDFS.java
4 errors
@rachanays →/workspaces/AI_unit1_QA1/source (main) $ javac ./TowerOfHanoiDFS.java
@rachanays →/workspaces/AI_unit1_QA1/source (main) $ java TowerOfHanoiDFS
Tower of Hanoi solution using DFS for 3 disks:
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
@rachanays →/workspaces/AI_unit1_QA1/source (main) $
bash source
bash source
bash source
Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
SUGGESTED ACTIONS
Build Workspace
Show Config
TowerOfHanoiDFS.java
Describe what to build next
Agent
Pick Model
OUTLINE
TIMELINE
Codespaces: glorious carnival
main*
Ln 26, Col 5
Spaces: 4
UTF-8
LF
Java
Finish Setup
Layout: US


AI_unit1_QA1 [Codespaces: glorious carnival]
EXPLORER
AI_UNIT1_QA1 [CODESPACES: G...
source
HanoiAStar.class
HanoiAStar.java
HanoiAStar$Neighbor.class
HanoiAStar$Node.class
HanoiAStar$State.class
TowerOfHanoiBFS.java
TowerOfHanoiDFS.java
HanoiAStar.java
source > HanoiAStar.java
import java.util.*;
/**
* A* solution to Tower of Hanoi.
* Disks are indexed 0..n-1 where 0 is the smallest and n-1 is th
* Pegs are 0, 1, 2. Goal: all disks on peg 2.
*/
public class HanoiAStar {
static class State {
final int[] pos; // pos[i] = peg of disk i
final int n;
State(int[] pos) {
PROBLEMS
OUTPUT
TERMINAL
bash - source
@rachanays →/workspaces/AI_unit1_QA1/source (main) $ java HanoiAStar
Move disk 1 from 0 to 2
Move disk 2 from 0 to 1
Move disk 1 from 2 to 1
Move disk 3 from 0 to 2
Move disk 1 from 1 to 0
Move disk 2 from 1 to 2
Move disk 1 from 0 to 2
@rachanays →/workspaces/AI_unit1_QA1/source (main) $
OUTLINE
TIMELINE
main*
Ln 1, Col 1
Spaces: 4
UTF-8
LF
Java
Finish Setup
Layout: US
CHAT
Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
SUGGESTED ACTIONS
Build Workspace
Show Config
HanoiAStar.java
Describe what to build next
Agent
Pick Model