

← → 🔍 backgammon ⚙️

EXPLORER ...

prd.md X

structure > prd.md > # 五子棋遊戲 PRD (Product Requirements Document)

```

1 # 五子棋遊戲 PRD (Product Requirements Document)
2
3 ## 1. 專案概述
4
5 ### 1.1 專案名稱
6 五子棋 (Gomoku) 線上遊戲
7
8 ### 1.2 專案描述
9 開發一個支援多人線上對戰的五子棋遊戲，採用前後端分離架構，提供流暢的遊戲體驗。
10
11 ### 1.3 目標用戶
12 - 喜歡策略遊戲的玩家
13 - 想要休閒娛樂的用戶
14 - 希望與朋友線上對戰的玩家
15
16 ## 2. 功能需求
17
18 ### 2.1 核心功能
19
20 #### 2.1.1 遊戲核心邏輯
21 - **棋盤**: 15x15 標準棋盤
22 - **獲勝條件**: 橫、直、斜任意方向連續五子
23 - **禁手規則**: 支援標準五子棋禁手規則 (黑棋)
24 - **落子規則**: 輪流落子，不可悔棋
25 - **遊戲結束**: 分出勝負或和局
26
27 #### 2.1.2 用戶系統
28 - **訪客模式**: 不需註冊即可遊玩
29 - **用戶註冊**: 郵箱/用戶名註冊
30 - **用戶登入**: 帳號密碼登入
31 - **用戶資料**: 頭像、暱稱、戰績統計
32
33 #### 2.1.3 遊戲模式
34 - **人機對戰**: 與AI對戰 (多難度級別)
35 - **線上對戰**: 玩家間實時對戰
36 - **房間系統**: 創建/加入遊戲房間
37 - **觀戰模式**: 觀看他人對戰
38
39 ### 2.2 擴展功能
40
41 #### 2.2.1 社交功能
42 - **好友系統**: 添加好友、好友對戰
43 - **聊天系統**: 遊戲內文字聊天
44 - **戰績查詢**: 歷史對戰記錄
45 - **排行榜**: 玩家排名系統

```

... Claude Code X

* Welcome to Claude Code!

/help for help, /status for your current setup

cwd: /Users/yi-changchen/Workspace/backgammon

Tips for getting started:

1. Ask Claude to create a new app or clone a repository
2. Use Claude to help with file analysis, editing, bash commands and git
3. Be as specific as you would with another engineer for the best results
4. ✓ Run /terminal-setup to set up terminal integration

> 我想要製作一個五子棋遊戲，我想要採取前後端分離的設計。請先不要寫任何程式，請先將需求文件PRD寫到 @structure/prd.md

* Thinking...

The user wants me to create a Gomoku (五子棋) game with a front-end and back-end separation design. They specifically asked me NOT to write any code yet, but to first create a PRD (Product Requirements Document) and save it to @structure/prd.md.

Let me check if there's already a structure directory or if I need to create one. I should also check the current working directory structure first.

Since this involves creating a new file and the user specifically asked for a PRD document, I should use the TodoWrite tool to plan this task.

The user mentioned "@structure/prd.md" which suggests they want it in a structure directory. Let me first check if this directory exists, then create the PRD document.

- I'll help you create a PRD for a Gomoku game with front-end and back-end separation. Let me first plan this task and check the current directory structure.
- Update Todos
 - Check current directory structure
 - Create structure directory if needed
 - Write comprehensive PRD for Gomoku game

* Thinking...

Good, I've created the todo list. Now let me check the current directory structure to see if the structure directory exists.

- List(..)
 - Listed 2 paths (ctrl+r to expand)

* Thinking...

The directory appears to be empty. Let me update my todo to mark the first task as in progress and create the structure directory.

< X ⊗ 0 △ 0 🔍

gomoku

EXPLORER

GOMOKU

- structure
- prd.md

prd.md

五子棋遊戲 PRD (Product Requirements Document) > ## 7. 成功指標

```

1 # 五子棋遊戲 PRD (Product Requirements Document)
2
3 ## 1. 專案概述
4
5 ### 1.1 專案名稱
6 五子棋 (Gomoku) 簡易版
7
8 ### 1.2 專案描述
9 開發一個簡單的五子棋遊戲，採用前後端分離架構，後端負責遊戲邏輯控制，前端負責界面渲染。
10
11 ### 1.3 目標用戶
12 - 想要快速遊玩五子棋的用戶
13 - 希望與朋友面對面對戰的玩家
14
15 ## 2. 功能需求
16
17 ### 2.1 核心功能
18
19 #### 2.1.1 遊戲核心邏輯
20 - **棋盤**: 15x15 標準棋盤
21 - **獲勝條件**: 橫、直、斜任意方向連續五子
22 - **落子規則**: 輪流落子，黑棋先手
23 - **遊戲結束**: 分出勝負或棋盤下滿和局
24
25 #### 2.1.2 遊戲模式
26 - **本地對戰**: 兩人輪流在同一設備上對戰，遊戲狀態由後端管理
27
28 ## 3. 技術架構
29
30 ### 3.1 前端技術棧
31 - **框架**: HTML5 + CSS3 + JavaScript (原生)
32 - **棋盤渲染**: HTML5 Canvas
33 - **HTTP請求**: Fetch API
34 - **部署**: 靜態網頁託管
35
36 ### 3.2 後端技術棧
37 - **語言**: Python 3.8+
38 - **框架**: FastAPI
39 - **遊戲邏輯**: 純Python實現
40 - **數據存儲**: 內存存儲 (簡易版)
41 - **部署**: Uvicorn
42
43 ### 3.3 通信架構
44 - **協議**: HTTP/HTTPS

```

Claude Code

Python

> /clear
 _ (no content)

> give me a simple git commit message based on `git diff`

- I'll run git diff to see the changes and create a simple commit message based on them.
- Bash(git diff)
 _ (No content)
- The git diff shows no changes. Let me check git status to see the current state:
- Bash(git status)
 _ On branch main
 Changes to be committed:
 (use "git restore --staged <file>..." to unstage)
 ... +1 line (ctrl+r to expand)
- Let me check the staged changes:
- Bash(git diff --cached)
 _ diff --git a/structure/prd.md b/structure/prd.md
 index 3d24414..6be2ef9 100644
 --- a/structure/prd.md
 ... +337 lines (ctrl+r to expand)
- Based on the git diff, this appears to be a simplification of the Gomoku game PRD, changing from a complex online multiplayer version to a simpler local version.

Simple commit message:

Update PRD to simplify Gomoku game scope

Simplify project from online multiplayer to local game

> give me a simple single git commit message based on `git diff`

- Simplify Gomoku game to local version

> @structure/prd.md : 這個五子棋的功能太過齊全，我想要盡可能的簡化

► auto-accept edits on (shift+tab to cycle)

OUTLINE

TIMELINE

main ⊞ ⊗ 0 ▲ 0

gomoku

EXPLORER

GOMOKU

structure

prd.md

prd.md

prd.md > # 五子棋遊戲 PRD (Product Requirements Document) > ## 7. 成功指標

```

1 # 五子棋遊戲 PRD (Product Requirements Document)
2
3 ## 1. 專案概述
4
5 ### 1.1 專案名稱
6 五子棋 (Gomoku) 簡易版
7
8 ### 1.2 專案描述
9 開發一個簡單的五子棋遊戲，採用前後端分離架構，後端負責遊戲邏輯控制，前端負責界面渲染。
10
11 ### 1.3 目標用戶
12 - 想要快速遊玩五子棋的用戶
13 - 希望與朋友面對面戰的玩家
14
15 ## 2. 功能需求
16
17 ### 2.1 核心功能
18
19 #### 2.1.1 遊戲核心邏輯
20 - **棋盤**: 15x15 標準棋盤
21 - **獲勝條件**: 橫、直、斜任意方向連續五子
22 - **落子規則**: 輪流落子，黑棋先手
23 - **遊戲結束**: 分出勝負或棋盤下滿和局
24
25 #### 2.1.2 遊戲模式
26 - **本地對戰**: 兩人輪流在同一設備上對戰，遊戲狀態由後端管理
27
28 ## 3. 技術架構
29
30 ### 3.1 前端技術棧
31 - **框架**: HTML5 + CSS3 + JavaScript (原生)
32 - **棋盤渲染**: HTML5 Canvas
33 - **HTTP請求**: Fetch API
34 - **部署**: 靜態網頁託管
35
36 ### 3.2 後端技術棧
37 - **語言**: Python 3.8+
38 - **框架**: FastAPI
39 - **遊戲邏輯**: 純Python實現
40 - **數據存儲**: 內存存儲 (簡易版)
41 - **部署**: Uvicorn
42
43 ### 3.3 通信架構
44 - **協議**: HTTP/HTTPS

```

Claude Code

Python

> /clear
_ (no content)

> give me a simple git commit message based on `git diff`

- I'll run git diff to see the changes and create a simple commit message based on them.
- Bash(git diff)
_ (No content)
- The git diff shows no changes. Let me check git status to see the current state:
- Bash(git status)
_ On branch main
Changes to be committed:
(use "git restore --staged <file>..." to unstage)
... +1 line (ctrl+r to expand)
- Let me check the staged changes:
- Bash(git diff --cached)
_ diff --git a/structure/prd.md b/structure/prd.md
index 3d24414..6be2ef9 100644
--- a/structure/prd.md
... +337 lines (ctrl+r to expand)
- Based on the git diff, this appears to be a simplification of the Gomoku game PRD, changing from a complex online multiplayer version to a simpler local version.

Simple commit message:

Update PRD to simplify Gomoku game scope

Simplify project from online multiplayer to local game

> give me a simple single git commit message based on `git diff`

- Simplify Gomoku game to local version

> @structure/prd.md :
我想要前後端分離，用python當作後端，遊戲的控制要放在後端執行

► auto-accept edits on (shift+tab to cycle)

OUTLINE

TIMELINE

main ⊞ ⊗ 0 ▲ 0

gomoku

EXPLORER

GOMOKU

structure

context_diagram.plantuml

prd.md

context_diagram.plantuml

```

structure > context_diagram.plantuml > {} context_diagram
1 @startuml context_diagram
2 !include https://raw.githubusercontent.com/plantuml-stdlib
3
4 title Gomoku Game – System Context Diagram
5
6 Person(player1, "Player 1", "玩家1 – 黑子先手")
7 Person(player2, "Player 2", "玩家2 – 白子後手")
8
9 System(gomoku_system, "Gomoku Game System", "五子棋遊戲系統\r\n本地對戰")
10
11 System_Ext(web_browser, "Web Browser", "網頁瀏覽器\r\n渲染遊戲界")
12
13 Rel(player1, web_browser, "使用", "點擊棋盤落子")
14 Rel(player2, web_browser, "使用", "點擊棋盤落子")
15 Rel(web_browser, gomoku_system, "通信", "HTTP/JSON API\r\n遊戲狀態同步")
16
17 note right of gomoku_system
18 **系統功能:**
19 • 15x15標準棋盤
20 • 輪流落子邏輯
21 • 勝負判斷算法
22 • 遊戲狀態管理
23 • RESTful API服務
24 end note
25
26 note left of web_browser
27 **前端功能:**
28 • HTML5 Canvas棋盤渲染
29 • 用戶點擊事件處理
30 • 遊戲狀態顯示

```

PROBLEMS OUTPUT TERMINAL PORTS

> TERMINAL

1. Run /init to create a CLAUDE.md file with instructions for Claude
2. Use Claude to help with file analysis, editing, bash commands and git
3. Be as specific as you would with another engineer for the best results
4. ✓ Run /terminal-setup to set up terminal integration

> 使用C4架構，基於 @structure/prd.md 來設計Context Diagram，並存於 @structure/context_diagram.plantuml

PlantUML Preview

Gomoku Game – System Context Diagram

```

graph TD
    Player1["Player 1<br>玩家1 – 黑子先手"] -- "使用<br>[點擊棋盤落子]" --> WebBrowser["Web Browser<br>網頁瀏覽器渲染遊戲界面並處理用戶互動"]
    Player2["Player 2<br>玩家2 – 白子後手"] -- "使用<br>[點擊棋盤落子]" --> WebBrowser
    WebBrowser -- "通信<br>[HTTP/JSON API<br>遊戲狀態同步]" --> GomokuSystem["Gomoku Game System<br>五子棋遊戲系統<br>提供15x15棋盤的本地對戰遊戲"]

```

前端功能:

- HTML5 Canvas棋盤渲染
- 用戶點擊事件處理
 - 遊戲狀態顯示
 - 控制按鈕界面

系統功能:

- 15x15標準棋盤
- 輪流落子邏輯
- 勝負判斷算法
- 遊戲狀態管理
- RESTful API服務

CLAUDE CODE USAGE MONITOR

[pro | Europe/Warsaw]

Token Usage: 0.0%

Tokens: 0 / ~44,000 (0 left)

Burn Rate: 0.0 tokens/min

02:21:03 No active session | Ctrl+C to exit

OUTLINE

TIMELINE

main* 0 △ 0

gomoku

EXPLORER

GOMOKU

- structure
- _deprecated
- c4
- system-gomoku_game_system
- container_diagram.plantuml**
- context_diagram.plantuml

prd.md

PROBLEMS

OUTPUT

TERMINAL

specifications:

- Web Application: HTML5/CSS3/JavaScript frontend with Canvas rendering
- Game API Server: Python/FastAPI backend with Uvicorn deployment
- Game Logic Engine: Pure Python game logic implementation
- Game State Store: In-memory storage for game state

The diagram now reflects the actual tech stack (Python/FastAPI) and API endpoints specified in the PRD.

> 使用C4架構，基於 @structure/c4 和 @structure/prd.md 來為System設計其下各自的Container Diagrams，並存於 @structure/c4/system-*/*.plantuml 下

► auto-accept edits on (shift+tab to cycle) ○ IDE disconnected

PlantUML Preview

```

structure > c4 > system-gomoku_game_system > container_diagram
1 @startuml container_diagram
2 !include https://raw.githubusercontent.com/plantuml-stdlib/C4-PlantUML/master
3
4 title Gomoku Game System - Container Diagram
5
6 Person(player1, "Player 1", "玩家1 - 黑子先手")
7 Person(player2, "Player 2", "玩家2 - 白子後手")
8
9 System_Boundary(gomoku_system, "Gomoku Game System") {
10     Container(web_app, "Web Application", "HTML5/CSS3/JavaScript", "前端靜態網頁")
11
12     Container(game_api, "Game API Server", "Python/FastAPI", "後端API服務器\n- RESTful API端點\n- 遊戲邏輯控制\n- 狀態管理服務\n- Uvicorn部署")
13
14     Container(game_engine, "Game Logic Engine", "Python", "核心遊戲邏輯引擎\n- 游戲狀態計算\n- 规則判斷\n- 落子處理")
15
16     ContainerDb(game_state, "Game State Store", "Memory Storage", "內存遊戲狀態儲存")
17 }
18
19 System_Ext(web_browser, "Web Browser", "網頁瀏覽器")
20
21 Rel(player1, web_browser, "使用", "點擊棋盤落子")
22 Rel(player2, web_browser, "使用", "點擊棋盤落子")
23 Rel(web_browser, web_app, "載入", "HTTP/HTTPS\n靜態資源")
24 Rel(web_app, game_api, "API調用", "HTTP/JSON\nRESTful請求")
25 Rel(game_api, game_engine, "調用", "函數調用\n遊戲邏輯處理")

```

Gomoku Game System [system]

Web Application [HTML5/CSS3/JavaScript]

前端靜態網頁應用
- HTML5 Canvas棋盤渲染
- 用戶互動事件處理
- Fetch API後端通信
- 遊戲狀態視覺化

API調用 [HTTP/JSON RESTful請求]

Game API Server [Python/FastAPI]

後端API服務器
- RESTful API端點
- 遊戲邏輯控制
- 狀態管理服務
- Uvicorn部署

Game Logic Engine [Python]

調用 [函數調用 遊戲邏輯處理]

Claude Code

CLAUDE CODE USAGE MONITOR

- Token Usage: 4.4%
- Time to Reset: 3h 51m
- Tokens: 1,955 / ~44,000 (42,045 left)
- Burn Rate: 28.6 tokens/min
- Predicted End: 03:37
- Token Reset: 07:00

Smooth sailing... | Ctrl+C to exit

gomoku

EXPLORER

GOMOKU

- structure
- _deprecated
- c4
- system-gomoku_game_system
- container-game_api
- component_diagram.plantuml
- container-game_engine
- component_diagram.plantuml
- container-game_state
- component_diagram.plantuml
- container-web_app
- component_diagram.plantuml
- container_diagram.plantuml
- context_diagram.plantuml

prd.md

container_diagram.plantuml

```

structure > c4 > system-gomoku_game
1 @startuml container_c4
2 !include https://raw.githubusercontent.com/...
3
4 title Gomoku Game Sys
5
6 Person(player1, "Play") --> Container(web_app)
7 Person(player2, "Play") --> Container(game_api)
8
9 System_Boundary(gomoku) --- Container(web_app)
10 Container(web_app) --- Container(game_api)
11
12 Container(game_api) --- Container(game_engine)
13 Container(game_engine) --- ContainerDb(game_db)
14
15 }
16
17
18 System_Ext(web_browser)
19
20 Rel(player1, web_browser)
21 Rel(player2, web_browser)
22 Rel(web_browser, web_app)
23 Rel(web_app, game_api)
24 Rel(game_api, game_engine)
25

```

PROBLEMS OUTPUT TERMINAL PORTS

TERMINAL

3. Game Logic Engine Container
(container-game_engine/component_diagram.plantuml)
- 包含遊戲邏輯的7個組件：遊戲管理器、棋盤管理器、落子驗證器、勝負判斷器、遊戲規則、玩家管理器和遊戲狀態計算器

4. Game State Store Container
(container-game_state/component_diagram.plantuml)
- 包含狀態存儲的7個組件：內存存儲、遊戲會話存儲、棋盤狀態存儲、玩家狀態存儲、遊戲元數據存儲、快取管理器和數據序列化器

每個組件圖都詳細說明了組件間的關係、職責分工，並包含了基於PR需求的具體實現細節。

> 使用C4架構，基於 @structure/c4 和 @structure/prd.md 來為Container設計其下各自的Component Diagrams，並存於 @structure/c4/system-*/container-*/component_diagram.plantuml

► auto-accept edits on (shift+tab to cycle) ○ IDE disconnected

PlantUML Preview

component_diagram.plantuml

Response Models [Python/Pydantic]
Game Controller [Python]
Validation Service [Python]
Game Service [Python]

服務層功能:
• 遊戲會話生命週期管理
• 玩家回合輪換控制
• 遊戲狀態持久化
• 業務邏輯封裝
• 事務處理

component

Claude Code

CLAUDE CODE USAGE MONITOR

Token Usage: 4.5%
Time to Reset: 3h 30m
Tokens: 1,990 / ~44,000 (42,010 left)
Burn Rate: 22.2 tokens/min
Predicted End: 11:01
Token Reset: 07:00
03:29:37 Smooth sailing... | Ctrl+C to exit

gomoku

EXPLORER

GOMOKU

- > architecture
- > ENV
- > **src**
- 23 .gitignore

game_service.py U .gitignore X

```

.gitignore
195 .pypirc
196
197 # Cursor
198 # Cursor is an AI-powered code editor. `.cursorignore` specifies files/directories to
199 # exclude from AI features like autocomplete and code analysis. Recommended for sensitive data
200 # refer to https://docs.cursor.com/context/ignore-files
201 .cursorignore
202 .cursorindexingignore
203
204 # Marimo
205 marimo/_static/
206 marimo/_lsp/
207 __marimo__/
208
209 # Streamlit
210 .streamlit/secrets.toml
211
212 # virtualenv
213 ENV/
214

```

PROBLEMS OUTPUT TERMINAL PORTS

TERMINAL

```

- Reset game: POST
http://localhost:8000/api/game/{game_id}/reset

The frontend is also served at the root URL.

> based on @architecture , to write down the codes in @src/ In .gitignore

```

Claude Code

CLAUDE CODE USAGE MONITOR

Token Usage:	15.9%
Time to Reset:	2h 19m
Tokens:	6,995 / ~44,000 (37,005 left)
Burn Rate:	43.5 tokens/min
Predicted End:	18:52
Token Reset:	07:00

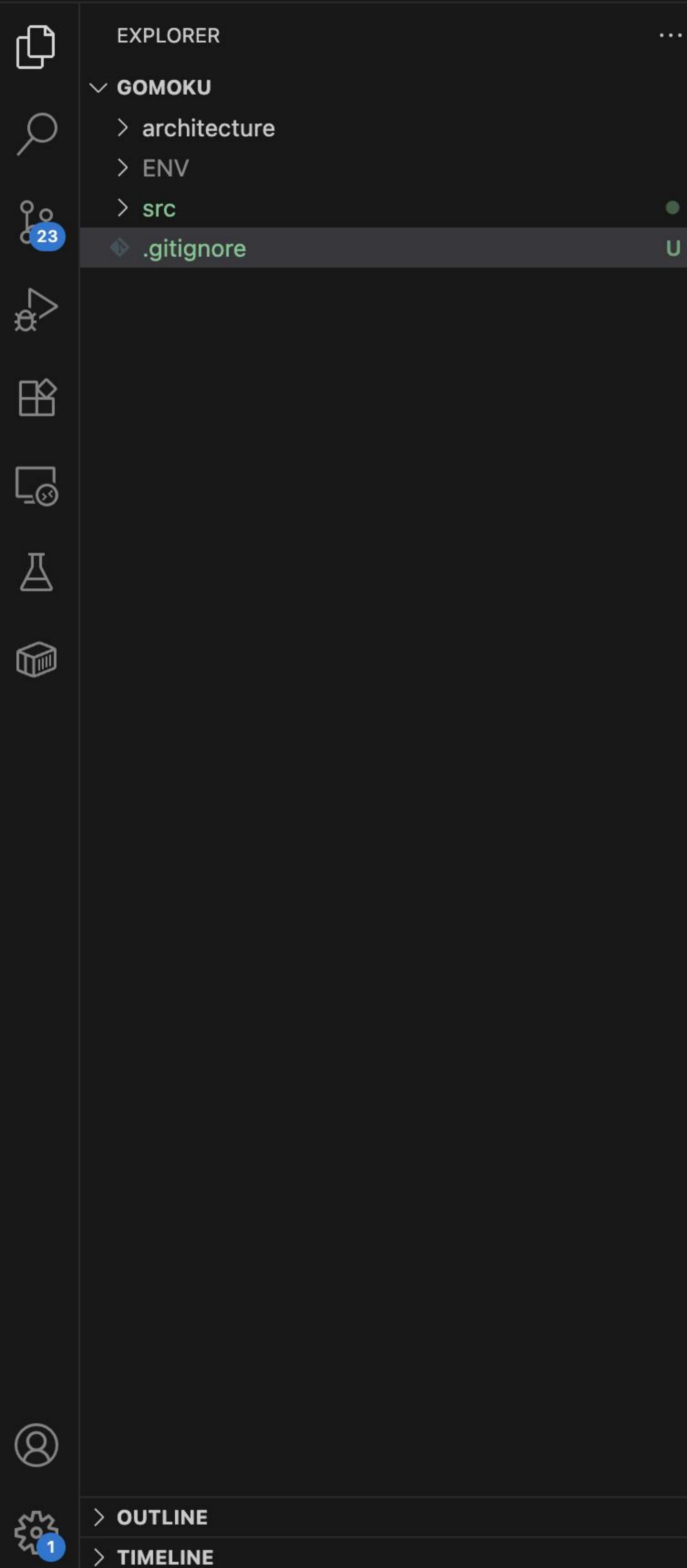
Smooth sailing... | Ctrl+C to exit

OUTLINE

TIMELINE

main* 0 △ 0

Ln 214, Col 1 Spaces: 4 UTF-8 LF {} Ignore



game_service.py U .gitignore X

⋮ .gitignore

```
195 .pypirc
196
197 # Cursor
198 # Cursor is an AI-powered code editor. `cursorignore` specifies files/directories to
199 # exclude from AI features like autocomplete and code analysis. Recommended for sensitive data
200 # refer to https://docs.cursor.com/context/ignore-files
201 .cursorignore
202 .cursorindexingignore
203
204 # Marimo
205 marimo/_static/
206 marimo/_lsp/
207 __marimo__/
208
209 # Streamlit
210 .streamlit/secrets.toml
211
212 # virtualenv
213 ENV/
214
```

PROBLEMS OUTPUT TERMINAL PORTS ⋮ ^ X

> ⚡ TERMINAL

```
- Reset game: POST
http://localhost:8000/api/game/{game_id}/reset

The frontend is also served at the root URL.

> To fix the issues when I run `cd src/backend/; python3 main.py`
```

⋮ In .gitignore

Claude Code + - X

CLAUDE CODE USAGE MONITOR

[pro | Europe/Warsaw]

Token Usage:	15.9%
Time to Reset:	2h 18m
Tokens:	6,995 / ~44,000 (37,005 left)
Burn Rate:	43.3 tokens/min
Predicted End:	18:55
Token Reset:	07:00

04:41:25 Smooth sailing... | Ctrl+C to exit

Ln 214, Col 1 Spaces: 4 UTF-8 LF {} Ignore 🌐

gomoku

EXPLORER

GOMOKU

- > architecture
- > ENV
- src
 - backend
 - > __pycache__
 - > game_api
 - > game_engine
 - > models
 - __init__.py
 - main.py
 - requirements.txt
- frontend
 - > css
 - > js
 - JS api-client.js
 - JS board-renderer.js
 - JS game-state-manager.js
 - JS game-ui.js
 - JS input-handler.js
 - JS main.js
 - <> index.html
- README.md
- <> .gitignore

PROBLEMS OUTPUT TERMINAL PORTS

TERMINAL

```
... +1189 lines (ctrl+r to expand)
● Bash(git log --oneline -5)
└ 6e9cae7 Add comprehensive .gitignore file
  ec1c76b Move architecture files from structure/ to architecture/
  directory
  ... +6 lines (ctrl+r to expand)

● Add complete Gomoku game implementation with FastAPI backend
  and HTML5 frontend
```

? for shortcuts

Claude Code Usage Monitor

- Token Usage: 16.4%
- Time to Reset: 2h 10m
- Tokens: 7,233 / ~44,000 (36,767 left)
- Burn Rate: 42.7 tokens/min
- Predicted End: 19:11
- Token Reset: 07:00

04:49:33 Smooth sailing... | Ctrl+C to exit

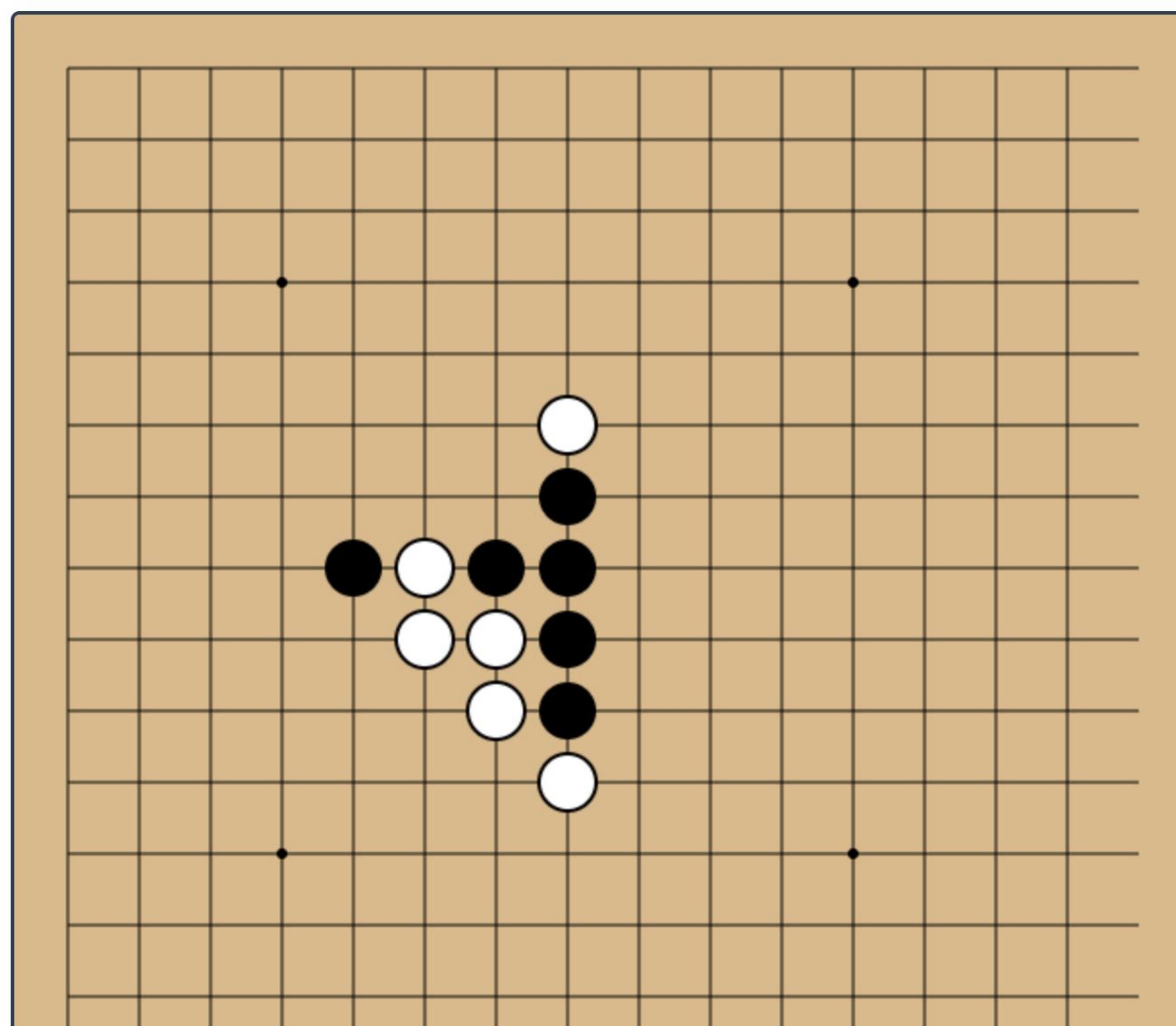
Ln 1, Col 1 Spaces: 4 UTF-8 LF {} HTML

當前玩家: 黑子

遊戲進行中

新遊戲

重新開始



gomoku

EXPLORER

GOMOKU

_vibe/tasks

20250707-01_add_scoreboard.md

architecture

c4

system-gomoku_game_system

> container-game_api

> container-game_engine

> container-game_state

container-web_app

component_diagram.plantuml

container_diagram.plantuml

context_diagram.plantuml

prd.md

ENV

src

.gitignore

.gitignore

20250707-01_add_scoreboard.md

_vibe > tasks > 20250707-01_add_scoreboard.md > abc ## task

1 ## task

2

3 我想要在前端介面增加記分板，紀錄雙方過去的得分狀況

PROBLEMS OUTPUT TERMINAL PORTS

> TERMINAL

> based on @_vibe/tasks/20250707-01_add_scoreboard.md , to redesign the architecture in @architecture

In 20250707-01_add_scoreboard.md

1h 37m

Tokens: 12,499 / ~44,000 (31,501 left)

Burn Rate: 61.7 tokens/min

Predicted End: 13:53

Token Reset: 07:00

05:22:36 Smooth sailing... | Ctrl+C to exit

Yi-Chang Chen (now) Ln 3, Col 26 Spaces: 4 UTF-8 LF {} Markdown

gomoku

EXPLORER

GOMOKU

- _vibe/tasks
 - 20250707-01_add_scoreboard.md
- architecture
 - c4
 - system-gomoku_game_system
 - > container-game_api
 - > container-game_engine
 - > container-game_state
 - > container-web_app
 - component_diagram.plantuml
 - container_diagram.plantuml
 - context_diagram.plantuml
- prd.md
- ENV
- src
- .gitignore

PROBLEMS OUTPUT TERMINAL PORTS

TERMINAL

> based on @_vibe/tasks/20250707-01_add_scoreboard.md and the re-designed architecture @architecture , to update the code in @src/

In 20250707-01_add_scoreboard.md

Claude Code 1h 37m

Tokens: 12,499 / ~44,000 (31,501 left)
Burn Rate: 61.6 tokens/min

Predicted End: 13:54
Token Reset: 07:00

05:22:58 Smooth sailing... | Ctrl+C to exit

Yi-Chang Chen (now) Ln 3, Col 26 Spaces: 4 UTF-8 LF {} Markdown

五子棋遊戲

記分板

重置記分板

黑子 (先手)

勝利 1 失敗 1

平手 0 總局數 2

勝率 50%

VS

白子 (後手)

勝利 1 失敗 1

平手 0 總局數 2

勝率 50%

當前玩家: 白子

白子獲勝!

新遊戲

重新開始

