

Claude AI

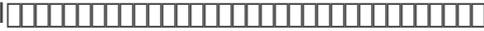
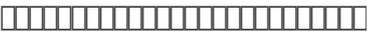
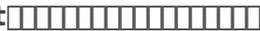
<https://claude.ai/new>

Claude AI Anthropic AI Claude 2023 ChatGPT

- [Claude AI](#)
- [Claude AI](#)

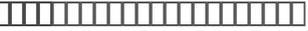
Claude AI



1. **Claude**  AI 
2. **Claude**  Claude 3 Haiku  Claude 3 Sonnet  Claude 3 Opus 
 - **Claude 3 Haiku** 
 - **Claude 3 Sonnet** 
 - **Claude 3 Opus** 
3. **Claude**  
4. **Claude**  



Claude AI 

- 
- 
- 

2. 矩形

javascript

```
class VirtualMap {
  constructor() {
    this.areas = [
      { name: 'A', x: 0, y: 0 },
      { name: 'B', x: 100, y: 100 },
      { name: 'C', x: 200, y: 50 },
      // ...
    ];
  }

  render() {
    // Canvas WebGL
  }

  movePlayer(x, y) {
    //
  }
}
```

3. 测验系统

javascript

```
class QuizSystem {
  constructor() {
    this.questions = [
      {
        question: '...',
        options: ['...', '...'],
        answer: 0
      },
      // ...
    ];
  }

  askQuestion() {
    //
  }

  checkAnswer(questionId, answerId) {
    //
  }
}
```

4. 任务

javascript

```
class TaskSystem {
  constructor() {
    this.tasks = [
      { id: 1, name: 'Task 1', completed: false },
      { id: 2, name: 'Task 2', completed: false },
      // ...
    ];
  }
}
```

```

}

assignTask() {
  // []
}

completeTask(taskId) {
  // []
}
}

```

5. []

javascript

```

class EcoSystem {
  constructor() {
    this.factors = {
      temperature: 25,
      humidity: 60,
      pollution: 10
    };
    this.flora = [];
    this.fauna = [];
  }

  update() {
    // []
  }

  addOrganism(type, species) {
    // []
  }

  removeOrganism(type, species) {
    // []
  }
}

```

6. []

javascript

```

class AchievementSystem {
  constructor() {
    this.achievements = [
      { id: 1, name: '100 unlocked', description: '100 unlocked', blocked: false },
      { id: 2, name: '50 unlocked', description: '50 unlocked', blocked: false },
      // ...
    ];
  }

  checkAchievements(player) {
    // []
  }
}

```

7. []

javascript

```
class MultiplayerSystem {
  constructor() {
    this.players = [];
    this.teams = [];
    this.projects = [];
  }

  createTeam(players) {
    // TODO
  }

  startProject(team, projectType) {
    // TODO
  }

  updateProjectProgress(projectId, progress) {
    // TODO
  }
}
```

TODO:

1. 使用 React/Vue.js 构建 UI
2. 使用 Three.js 实现 3D 渲染
3. 使用 Socket.io 实现实时通信
4. 使用 Express.js, MongoDB 搭建后端

如何设计数据库 schema, 如何设计 API, 如何设计 UI?

如何设计数据库 schema, 如何设计 API?