**RACI Table for Pac-Man Game Development Project ( Waterfall Methodology )**

| **SDLC Phase & Task** | **Project Manager** | **Game Designer** | **Developer** | **QA Tester** | **IT Support** |
| --- | --- | --- | --- | --- | --- |
| **1. Requirements Gathering & Analysis** |  |  |  |  |  |
| 1.1 Define Game Requirements | R, A | C, I | I |  |  |
| 1.2 Conduct Feasibility Study | R, A | I | C |  | I |
| 1.3 Stakeholder Alignment | R, A | C | I |  |  |
| **2. System Design** |  |  |  |  |  |
| 2.1 Design Game Architecture | C | R, A | I |  | I |
| 2.2 Design User Interface (UI) | C, I | R, A |  |  |  |
| 2.3 Design Gameplay Flow | C | R, A | I |  |  |
| **3. Development** |  |  |  |  |  |
| 3.1 Setup Development Environment | A | I | R |  | C |
| 3.2 Code Game Mechanics | I | C | R, A |  |  |
| 3.3 Create Visual and Audio Assets | C | R, A | I |  |  |
| **4. Testing** |  |  |  |  |  |
| 4.1 Conduct Functional Testing | I | C | R | A |  |
| 4.2 Conduct Playability Testing | I | C |  | R, A |  |
| 4.3 Performance Testing | I | C |  | R, A |  |
| **5. Deployment** |  |  |  |  |  |
| 5.1 Prepare for Launch | R, A | I | C |  |  |
| 5.2 Execute Game Launch | R, A | I | C |  |  |
| 5.3 Monitor Initial Performance | R, A | I | C |  |  |
| **6. Maintenance & Support** |  |  |  |  |  |
| 6.1 Provide Ongoing Support | R | I | C |  | A |
| 6.2 Implement Updates and Enhancements | A | I | R | C |  |
| 6.3 Conduct Regular Testing and Maintenance | A | C |  | R | I |

**Explanation of the RACI Matrix**

Each phase and task from the SDLC (Software Development Life Cycle) is broken down with specific responsibilities:

**1. Requirements Gathering & Analysis**

* **Define Game Requirements**: The **Project Manager** (R, A) leads and approves the definition of requirements, while the **Game Designer** (C) provides the creative input necessary for gameplay elements.
* **Conduct Feasibility Study**: The **Project Manager** is responsible and approves the feasibility analysis. The **Developer** and **IT Support** provide insights on technical viability.
* **Stakeholder Alignment**: The **Project Manager** (A) arranges alignment, with the **Game Designer** (C) providing creative input.

**2. System Design**

* **Design Game Architecture**: The **Game Designer** (R, A) leads the design of the game structure, while the **Developer** (I) and **IT Support** (I) ensure the feasibility of technical components.
* **Design User Interface (UI)**: The **Game Designer** (R, A) designs the UI, with the **Project Manager** and **IT Support** involved for input and feedback.
* **Design Gameplay Flow**: The **Game Designer** takes the lead, supported by input from the **Project Manager** and **Developer** to ensure technical alignment.

**3. Development**

* **Setup Development Environment**: **Developer** (R) is responsible for setting up the environment, with oversight from **IT Support** (C).
* **Code Game Mechanics**: **Developer** (R, A) codes and tests mechanics, with input from the **Game Designer** on gameplay elements.
* **Create Visual and Audio Assets**: The **Game Designer** is responsible for visual/audio assets, ensuring all elements align with the game’s vision.

**4. Testing**

* **Conduct Functional Testing**: **QA Tester** (A) leads functional testing, with input from **Developer** to address any technical issues.
* **Conduct Playability Testing**: **QA Tester** (R, A) is responsible, with feedback from **Game Designer** for player experience.
* **Performance Testing**: **QA Tester** (R, A) is responsible for assessing performance, with **Developer** addressing performance issues identified.

**5. Deployment**

* **Prepare for Launch**: **Project Manager** (R, A) coordinates launch preparations, with **Developer** ensuring the game is production-ready.
* **Execute Game Launch**: **Project Manager** (R, A) oversees launch activities, with support from **Developer** on technical aspects.
* **Monitor Initial Performance**: **Project Manager** and **Developer** monitor performance and address user feedback.

**6. Maintenance & Support**

* **Provide Ongoing Support**: **IT Support** (A) provides ongoing technical support, while the **Project Manager** oversees issue resolution.
* **Implement Updates and Enhancements**: **Developer** (R) leads updates, with **QA Tester** involved for validation.
* **Conduct Regular Testing and Maintenance**: **QA Tester** (R) leads testing for stability, with **Developer** making necessary adjustments and **IT Support** assisting as needed.

**Work Breakdown Structure (WBS) for Pac-Man Game Development Project ( Waterfall Methodology )**

| **WBS Code** | **Task/Activity** | **Resources (Team, Hardware & Software)** | **Dependencies** | **Effort (Hours)** | **Man-Days** |
| --- | --- | --- | --- | --- | --- |
| **1** | Requirements Gathering & Analysis | Project Manager, Game Designer, Developer, IT Support  Hardware: Laptops, Meeting Room Setup  Software: Requirements Management Tool (e.g., JIRA, Confluence) | None | 60 | 7.5 |
| **1.1** | Define Game Requirements | Project Manager, Game Designer  Hardware: Laptops  Software: Documentation Tool (e.g., Google Docs) | None | 20 | 2.5 |
| **1.2** | Conduct Feasibility Study | Project Manager, Developer, IT Support  Hardware: Workstations  Software: Feasibility Analysis Tool, Database Analysis Tool | 1.1 | 20 | 2.5 |
| **1.3** | Stakeholder Alignment | Project Manager, Game Designer  Hardware: Meeting Room, Projector  Software: Presentation Tool (e.g., PowerPoint) | 1.1, 1.2 | 20 | 2.5 |
| **2** | System Design | Game Designer, Developer, IT Support  Hardware: Workstations  Software: UML Design Tool (e.g., Lucidchart), IDE (e.g., Unity) | 1.3 | 80 | 10 |
| **2.1** | Design Game Architecture | Game Designer, Developer  Hardware: Workstations  Software: UML Tool, Game Engine (e.g., Unity) | 1.3 | 30 | 3.75 |
| **2.2** | Design User Interface (UI) | Game Designer  Hardware: Graphics Tablet, Workstation  Software: UI Design Tool (e.g., Adobe XD, Figma) | 2.1 | 30 | 3.75 |
| **2.3** | Design Gameplay Flow | Game Designer, Developer  Hardware: Workstations  Software: Game Engine, UML Tool | 2.1 | 20 | 2.5 |
| **3** | Development | Developer, Game Designer, IT Support  Hardware: High-Performance Workstations  Software: Game Engine, Version Control (e.g., Git) | 2.3 | 200 | 25 |
| **3.1** | Setup Development Environment | Developer, IT Support  Hardware: Workstations, Server Access  Software: IDE, Version Control (e.g., GitHub) | 2.3 | 40 | 5 |
| **3.2** | Code Game Mechanics | Developer, Game Designer  Hardware: High-Performance Workstations  Software: Game Engine, IDE (e.g., Unity), Debugging Tool | 3.1 | 100 | 12.5 |
| **3.3** | Create Visual and Audio Assets | Game Designer  Hardware: Graphics Tablet, Audio Recording Equipment  Software: Adobe Photoshop, Audio Editing Tool (e.g., Audacity) | 3.1 | 60 | 7.5 |
| **4** | Testing | QA Tester, Developer  Hardware: Test Devices (PC, Mobile)  Software: Testing Tools (e.g., Selenium), Game Engine | 3 | 120 | 15 |
| **4.1** | Conduct Functional Testing | QA Tester, Developer  Hardware: Test Devices  Software: Test Management Tool (e.g., TestRail) | 3.3 | 40 | 5 |
| **4.2** | Conduct Playability Testing | QA Tester  Hardware: Test Devices  Software: Playability Test Tool | 4.1 | 40 | 5 |
| **4.3** | Performance Testing | QA Tester  Hardware: Test Server, High-Performance PC  Software: Performance Testing Tool (e.g., JMeter) | 4.2 | 40 | 5 |
| **5** | Deployment | Project Manager, Developer, IT Support  Hardware: Production Server, Networking Equipment  Software: Deployment Tool (e.g., Jenkins) | 4 | 40 | 5 |
| **5.1** | Prepare for Launch | Project Manager, Developer  Hardware: Server Access  Software: Deployment Tool, Database Management System | 4 | 10 | 1.25 |
| **5.2** | Execute Game Launch | Project Manager, Developer  Hardware: Server, Test Devices  Software: Monitoring Tool (e.g., New Relic) | 5.1 | 20 | 2.5 |
| **5.3** | Monitor Initial Performance | Project Manager, Developer  Hardware: Server, Monitoring Dashboard  Software: Application Performance Monitoring Tool | 5.2 | 10 | 1.25 |
| **6** | Maintenance & Support | IT Support, Developer, QA Tester  Hardware: Server, Workstations  Software: Issue Tracking Tool (e.g., JIRA), IDE | 5.3 | 120 | 15 |
| **6.1** | Provide Ongoing Support | IT Support  Hardware: Workstations  Software: Helpdesk System (e.g., ServiceNow) | 5.3 | 40 | 5 |
| **6.2** | Implement Updates and Enhancements | Developer, QA Tester  Hardware: Workstations  Software: IDE, Version Control | 6.1 | 60 | 7.5 |
| **6.3** | Conduct Regular Testing and Maintenance | QA Tester, IT Support  Hardware: Test Devices  Software: Test Automation Tool (e.g., Selenium) | 6.2 | 20 | 2.5 |

**RACI Matrix with Specific Development Team Roles:**

| **Phase/Task** | **Product Owner (PO)** | **Scrum Master (SM)** | **Development Team (Dev)** | **QA** |
| --- | --- | --- | --- | --- |
| **Sprint 0: Initial Setup** |  |  |  |  |
| Define User Stories | A | C | R (Game Developer) | I |
| Set Up Dev Environment | C | A | R (DevOps) | I |
| **Sprint 1: Core Game Mechanics** |  |  |  |  |
| Develop Basic Movement | C | I | R (Game Developer) | R |
| Code and Test Collision Mechanics | C | I | R (Game Developer) | R |
| Code and Test Scoring System | C | I | R (Game Developer) | R |
| Sprint Review & Retrospective | A | R | C | C |
| **Sprint 2: Advanced Gameplay** |  |  |  |  |
| Develop AI for Ghosts | C | I | R (Game Developer) | R |
| Code and Test Power-Up Functionality | C | I | R (Game Developer) | R |
| Implement Sound and Visual Feedback | C | I | R (Game Developer) | R |
| Sprint Review & Retrospective | A | R | C | C |
| **Sprint 3: UI and Level Design** |  |  |  |  |
| Develop Main Menu | C | I | R (Game Designer/UI Designer) | R |
| Code and Test Level 1 Layout | C | I | R (Game Designer/Game Developer) | R |
| Review and Adjust UI Elements | A | I | R (Game Designer/UI Designer) | C |
| Sprint Review & Retrospective | A | R | C | C |
| **Sprint 4: Final Integration & Testing** |  |  |  |  |
| Full Game Integration | C | I | R (Game Developer) | R |
| Conduct Regression Testing | C | I | R (Game Developer) | A |
| Conduct User Acceptance Testing | C | I | R (Game Developer) | A |
| Sprint Review & Retrospective | A | R | C | C |
| **Release and Post-Launch Support** |  |  |  |  |
| Prepare Game for Production | A | I | R (Game Developer/DevOps) | C |
| Monitor Game Post-Launch | C | I | R (Game Developer/QA) | A |

 **Product Owner (PO)** is **Accountable (A)** for the overall vision and user stories. They also ensure that the product backlog is well-defined, prioritized, and aligned with the business objectives.

 **Scrum Master (SM)** is **Responsible (R)** for facilitating Scrum processes and removing obstacles. They ensure that Scrum ceremonies (such as the Sprint Review and Retrospective) are conducted efficiently and serve the project’s objectives.

 **Development Team (Dev)** is **Responsible (R)** for delivering the increments during each sprint. They write code, design game mechanics, implement features, and ensure all development tasks are completed according to the Sprint Goal.

 **Game Developer** (Programmer):

* **Responsibility**: Develops the core game logic, mechanics (such as movement, collision detection, and AI), and features (such as scoring system, power-ups, etc.). They also code the backend systems (e.g., game physics, level design).
* **Role in RACI**: *Responsible (R)* for most of the tasks related to coding, designing, and implementing features. For example, tasks like coding the AI for ghosts, developing the player movement, and integrating sound effects.

 **Game Designer**:

* **Responsibility**: Designs the game flow, user interface (UI), user experience (UX), and overall aesthetics of the game. They ensure the game mechanics are fun, intuitive, and engaging. The game designer also helps develop game levels, backgrounds, and character designs.
* **Role in RACI**: *Consulted (C)* on tasks such as designing the main menu, user interface elements, level design, and visual feedback. While they may not directly implement code, their design decisions are essential to guide development.

 **Quality Assurance (QA)**:

* **Responsibility**: Though listed separately in the RACI matrix, QA is considered part of the Development Team in Scrum. QA professionals are involved early in the sprint to ensure testing criteria are defined, automation tools are set up, and they provide continuous feedback on game stability and usability.
* **Role in RACI**: While QA is usually *Responsible (R)* for testing tasks, they work closely with developers to ensure that the tests are aligned with the development process. They might also be *Consulted (C)* for certain design decisions related to functionality and user experience.

 **Technical Writer**:

* **Responsibility**: In some Scrum teams, especially for game development, a technical writer might be included to create documentation for the game, such as user manuals, game instructions, and API documentation if applicable.
* **Role in RACI**: *Consulted (C)* on tasks such as documenting the game logic, rules, and in-game instructions.

 **DevOps/Systems Engineer**:

* **Responsibility**: Supports the team by managing the development environment, ensuring that the integration and continuous deployment systems are functioning properly. They may handle deployment pipelines, server maintenance, and environment setup for testing and production.
* **Role in RACI**: *Consulted (C)* or *Responsible (R)* for setting up the environments and ensuring the game is integrated into deployment pipelines effectively.

 **UI/UX Designer**:

* **Responsibility**: Works on the visual design elements of the game, including creating icons, buttons, and optimizing user experience. They ensure that the game interface is both functional and visually appealing.
* **Role in RACI**: *Consulted (C)* for tasks like designing game UI elements, visual feedback, and ensuring a user-friendly experience.

 **Quality Assurance (QA)** is **Responsible (R)** for ensuring the quality of the product at every stage of development. They write test cases, run automated and manual tests, and provide feedback to developers for continuous improvement.

**Work Breakdown Structure (WBS) for Pac-Man Game Development Project (Scrum Testing Methodology)**

| **WBS Code** | **Task/Activity** | **Resources (Team, Hardware & Software)** | **Dependencies** | **Effort (Hours)** | **Man-Days** |
| --- | --- | --- | --- | --- | --- |
| **1** | Sprint 0: Initial Setup | Product Owner, Scrum Master, Project Manager, DevOps  Hardware: Laptops, Meeting Room Setup  Software: Scrum Board (e.g., JIRA), Version Control | None | 40 | 5 |
| **1.1** | Define User Stories | Product Owner, Scrum Master  Hardware: Laptops  Software: JIRA, Documentation Tool | None | 20 | 2.5 |
| **1.2** | Set Up Dev Environment | DevOps, Developer  Hardware: Workstations  Software: IDE, GitHub | None | 20 | 2.5 |
| **2** | Sprint 1: Core Game Mechanics | Scrum Team (Product Owner, Developer, QA)  Hardware: Workstations  Software: Game Engine, Testing Framework (e.g., Selenium) | 1.2 | 160 | 20 |
| **2.1** | Develop and Test Basic Movement | Developer, QA  Hardware: High-Performance Workstations  Software: IDE, Testing Framework | 1.1 | 40 | 5 |
| **2.2** | Code and Test Collision Mechanics | Developer, QA  Hardware: Workstations  Software: Game Engine, Testing Framework | 2.1 | 40 | 5 |
| **2.3** | Code and Test Scoring System | Developer, QA  Hardware: Workstations  Software: Game Engine, JIRA | 2.2 | 40 | 5 |
| **2.4** | Sprint Review & Retrospective | Scrum Team  Hardware: Meeting Room  Software: Scrum Board, Documentation Tool | 2.3 | 40 | 5 |
| **3** | Sprint 2: Advanced Gameplay Elements | Scrum Team (Product Owner, Developer, QA)  Hardware: High-Performance Workstations  Software: Game Engine, Automation Testing Tool | 2 | 160 | 20 |
| **3.1** | Develop and Test AI for Ghosts | Developer, QA  Hardware: Workstations  Software: Game Engine, Testing Framework | 2 | 60 | 7.5 |
| **3.2** | Code and Test Power-Up Functionality | Developer, QA  Hardware: Workstations  Software: IDE, Automation Tool | 3.1 | 60 | 7.5 |
| **3.3** | Implement Sound and Visual Feedback | Developer, QA  Hardware: Audio Equipment  Software: Adobe Photoshop, Audio Tool | 3.2 | 40 | 5 |
| **3.4** | Sprint Review & Retrospective | Scrum Team  Hardware: Meeting Room  Software: Scrum Board, Documentation Tool | 3.3 | 40 | 5 |
| **4** | Sprint 3: UI and Level Design | Scrum Team (Product Owner, Developer, Game Designer, QA)  Hardware: Graphics Tablets  Software: UI Design Tool, Game Engine | 3 | 160 | 20 |
| **4.1** | Develop and Test Main Menu | Developer, QA  Hardware: Workstations  Software: Game Engine, UI Testing Tool | 3.4 | 40 | 5 |
| **4.2** | Code and Test Level 1 Layout | Game Designer, Developer  Hardware: Graphics Tablet  Software: Game Engine, Testing Framework | 4.1 | 80 | 10 |
| **4.3** | Review and Adjust UI Elements | Scrum Team  Hardware: Meeting Room  Software: UI Design Tool | 4.2 | 40 | 5 |
| **5** | Sprint 4: Final Integration and Testing | Scrum Team (Product Owner, Developer, QA)  Hardware: Test Devices  Software: Game Engine, Automated Test Tool | 4 | 160 | 20 |
| **5.1** | Full Game Integration | Developer  Hardware: High-Performance Workstations  Software: IDE, Version Control | 4.3 | 80 | 10 |
| **5.2** | Conduct Regression Testing | QA Tester  Hardware: Test Devices  Software: Automated Testing Framework | 5.1 | 40 | 5 |
| **5.3** | Conduct User Acceptance Testing | Scrum Team  Hardware: Test Devices  Software: Game Engine, UAT Tool | 5.2 | 40 | 5 |
| **6** | Release and Post-Launch Support | Scrum Team (Product Owner, Developer, IT Support)  Hardware: Production Server  Software: Monitoring Tool | 5 | 80 | 10 |
| **6.1** | Prepare Game for Production | Scrum Team  Hardware: Server Access  Software: Deployment Tool (e.g., Jenkins) | 5.3 | 40 | 5 |
| **6.2** | Monitor Game Post-Launch | IT Support  Hardware: Monitoring Dashboard  Software: Application Performance Monitoring Tool | 6.1 | 40 | 5 |

 **Integration of Scrum Phases**: Each sprint focuses on a specific aspect of the game, from core mechanics to UI, integrating testing within each sprint. Each sprint concludes with a Sprint Review and Retrospective to assess functionality and gather feedback.

 **Testing Embedded in Each Sprint**: Instead of a single, isolated testing phase, testing is embedded within each sprint, involving QA testers and Scrum team members to verify completed features, ensure functionality, and facilitate early detection of issues.

 **Continuous Feedback and Adjustment**: Each sprint provides opportunities for review and realignment with project goals, ensuring that the game evolves based on testing feedback, stakeholder input, and user needs.

 **Post-Launch Support and Monitoring**: The final phase includes production readiness, launch, and ongoing monitoring to address any performance issues or user feedback post-release.