Game Development Phases and Time Allocation

Phase	Description	Time (%)
Planning & Game Design	Playing similar games (e.g., Starcraft, Galactic Frontier) to understand core mechanics, analyzing gameplay, developing storyline, concepts, documentation, and wireframes.	%15
Prototyping	Developing basic gameplay mechanics, core elements (units, resources, buildings), feasibility validation, and refining concepts based on feedback.	%15
Core Gameplay Development	Implementing primary mechanics (combat systems, unit interactions, movement algorithms, resources, basic AI behaviors, rules, and objectives).	%25
Advanced Feature Development	Enhancing gameplay with complex AI, special abilities, multiplayer integration, optimized pathfinding, and refined UI/UX.	%20
Visual & Audio Integration	Integrating graphics, animations, VFX, audio assets (music, SFX) for immersion.	%10
Testing & Debugging	Extensive testing, playtesting, bug fixing, gameplay tuning, and performance optimization.	%10
Deployment & Launch Preparation	Platform-specific optimization, final build creation, marketing materials preparation, and official game release.	%5

Market research is the first thing I do when I have a game idea in mind. By determining what other games are performing well or poorly, this research aids in my understanding of the competition. I can better understand what works and what doesn't by playing similar well-known games and learning about their features and mechanics. I identify precisely what is required to develop the game after finishing the research. I put together knowledgeable team that can handle every facet of the development process based on these specifications.

The prototyping stage comes next, during which I create a scaled-down version of the game to test fundamental concepts and gameplay elements. Essential elements like buildings, units, resources, and basic actions are established during this phase. Through prototyping, I can assess the viability of the game concept and make quick adjustments in response to early feedback. Before beginning full-scale development, this helps address possible problems early and enhance the overall design. My attention turns to creating the main systems that enable the game to be played and enjoyed during the core gameplay development phase. Important components are put into place, including win/lose conditions, resource gathering, unit movements, combat mechanics, and building construction. In order for players to engage with the game world in a meaningful way, I also create fundamental AI behaviors. The goal is to produce a solid and captivating gameplay experience that stays true to the RTS game's original intent.

After that, I move on to the advanced feature development stage, adding more intricate and well-designed systems to improve the core gameplay. During this stage, intricate AI strategies are created, unique unit abilities are added, pathfinding algorithms are optimized, and multiplayer functionality is added if needed. The game is dynamic, difficult, and unique in the RTS genre because a lot of effort is put into enhancing the user interface and overall experience.

In-depth analyses are carried out to find and address bugs, performance concerns, and gameplay issues during the testing and debugging stage. Stability and fluid gameplay are ensured by techniques like device compatibility checks, user playtesting, and unit testing. I can fix crashes, balance the game, and improve the overall player experience by collecting and evaluating feedback.

Lastly, the deployment and launch preparation phase entails making the final build, making sure the game works well on all supported systems, and optimizing it for different mobile devices and platforms. A thorough launch strategy is also planned, store listings are created, and marketing materials are created. This stage aims to guarantee the game's successful launch, optimal exposure, and a favorable initial impression among players.

I will gather player feedback following the launch and utilize it to make the game better. This entails resolving any bugs that were overlooked during testing, modifying the balance of gameplay, and potentially incorporating new features in response to player feedback. Regular updates enhance user satisfaction, foster a vibrant and devoted player base, and keep the game interesting.

Early Gameplay Development Plan

First 5 Mechanics/Functionalities to Develop

- Unit Spawning Allows players to produce units, forming the base of their army.
- Unit Movement Enables players to select and move units across the map.
- Resource Collection Units gather resources to start building the economy.
- Building Placement Lets players construct buildings and strategically position structures.
- Combat Mechanic Units can engage enemies, take and deal damage.

Questions Game Designers Should Answer Before Development

Unit Spawning:

- Which buildings can produce which units?
- What is the production time?
- What is the resource cost per unit?
- Will there be a production queue system?

Unit Movement:

- Will units be selected by click or drag?
- Will movement be grid-based or freeform?
- Should units avoid collisions with each other?

Resource Collection:

- What types of resources exist (e.g., minerals, energy)?
- Which units are able to gather resources?
- Are resources limited or renewable?
- What is the gathering time once a unit reaches a resource?

Building Placement:

- Where can buildings be placed?
- Is there a construction time?
- Should there be collision checks or placement restrictions?
- Can players cancel a building placement?

Combat Mechanic:

- What is the attack range?
- How does the damage system work? (e.g., cooldowns, critical hits?)
- How will enemies be detected? (automatic or command-based?)
- Will units have health, armor, or other defensive stats?