Project Approach and Lessons Learned

Garbage Collector Game - Multiplayer Trash Sorting Game

Project Approach

Development Model: Waterfall Model

- Linear, sequential approach with distinct phases
- Requirements → Design → Implementation → Testing → Deployment
- Clear documentation at each phase before moving forward

Technology Stack:

- **Engine:** Godot Engine (GDScript)
- **Architecture:** Client-Server Model
- **Networking:** Godot's built-in ENetMultiplayerPeer
- **Platform:** PC (Windows/Linux/Mac)

Development Phases:

1. **Requirements Analysis** - Defined functional and non-functional requirements
2. **System Design** - Created architectural documents and system interfaces
3. **Implementation** - Coded game components with role-based task distribution
4. **Integration** - Combined multiplayer, UI, collision, and audio systems
5. **Testing** - Verified network synchronization and gameplay mechanics

Key Lessons Learned

Technical Insights:

- **Multiplayer Complexity:** Network synchronization requires careful RPC implementation
- **Godot Engine:** Built-in multiplayer features simplify networking but need proper understanding
- **Simple Architecture:** Host-client model works well for small-scale multiplayer games

Team Management:

- **Clear Role Division:** Assigning specific components to team members improved efficiency
- **Documentation First:** Having architectural documents helped coordinate development
- **Regular Integration:** Frequent code merging prevented major conflicts

Development Challenges:

- **Network Debugging:** Multiplayer bugs are harder to debug than single-player issues
- **State Synchronization:** Ensuring both players see identical game states
- **Performance Balance:** Managing trash spawning to prevent lag

Best Practices Discovered:

- Keep networking code separate from game logic
- Use immediate visual feedback for better user experience
- Test multiplayer features with actual network connections, not just local testing
- Simple control schemes work better for multiplayer games

Project Management:

- Waterfall model worked well for educational project with clear requirements
- Having 5 team members required good communication and task coordination
- Documentation helped team members understand each other's components

Conclusion

The project successfully delivered a functional multiplayer game while teaching important software engineering concepts including network programming, team collaboration, and system architecture design.