CSSE2025 Assignment 2 – AI Section

This document outlines the AI (Achievements Integration) component of the assignment, detailing the design decisions and implementation based on the provided specifications and Javadoc expectations. It also explains how SOLID principles were followed throughout the refactoring and enhancement of the codebase.

1. Achievement Integration Overview

The Achievement system was introduced in Assignment 2 to track player progress and provide motivational feedback. Three standard achievements were added: 'Survivor', 'Enemy Exterminator', and 'Sharp Shooter'. These are tracked every tick and rendered in the Game Over screen.

2. Component Implementations

2.1 GameController.java

- Added field: AchievementManager achievementManager
- Constructor updated to receive and store AchievementManager instance
- refreshAchievements(tick) method implemented
- showGameOverWindow() updated to show achievement progress
- PlayerStatsTracker used to track shots fired/hit

2.2 AchievementManager.java

- Implements functionality to add, update, and list achievements
- Logs newly mastered achievements through AchievementFile
- Ensures no duplicate achievements are registered

2.3 GameAchievement.java

- Implements Achievement interface
- Tracks progress between 0.0 to 1.0
- Returns tier label based on progress (Novice, Expert, Master)

2.4 PlayerStatsTracker.java

- Tracks number of shots fired and hit
- Calculates accuracy and survival time

2.5 FileHandler.java

- Implements AchievementFile interface
- Handles file IO for saving achievement logs

3. SOLID Principles Applied

Single Responsibility: Each class has one clear purpose.
Open/Closed: Classes like GameAchievement and GameController are open for extension (via interface) but closed for modification by separating achievement logic.
Liskov Substitution: Achievement interface is cleanly implemented by GameAchievement.
Interface Segregation: Small, focused interfaces like Achievement and AchievementFile improve flexibility.
Dependency Inversion: GameController depends on AchievementManager (abstraction), not directly on file logic.