**A PROJECT REPORT**

**on**

**“SPACE INVADERS GAME”**

**Submitted to**

**KIIT Deemed to be University**

**In Partial Fulfilment of the Requirement for the Award of**

**BACHELOR’S DEGREE IN**

**COMPUTER SCIENCE AND ENGINEERING**

**BY**

**ADHIRAJ GHOSAL 22053132**

**RAJESHWARI CHOUDHURY 22053180**

**UNDER THE GUIDANCE OF**

**Dr.SRICHETA PARUI**



**SCHOOL OF COMPUTER ENGINEERING**

**KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY**

**BHUBANESWAR, ODISHA - 751024**

**April 2025**

A PROJECT REPORT

on

“SPACE INVADERS GAME”

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR’S DEGREE IN

COMPUTER SCIENCE AND ENGINEERING

BY

ADHIRAJ GHOSAL 22053132

RAJESHWARI CHOUDHURY 22053180

UNDER THE GUIDANCE OF

Dr.SRICHETA PARUI



SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA - 751024

April 2025

KIIT Deemed to be University

School of Computer Engineering

Bhubaneswar, ODISHA 751024



**CERTIFICATE**

This is certify that the project entitled

“SPACE INVADERS GAME“

submitted by

ADHIRAJ GHOSAL 22053132

RAJESHWARI CHOUDHURY 22053180

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2024-2025, under our guidance.

Date: 24/03/2025

DR.SRICHETA PARUI

Project Guide

**Acknowledgements**

We are profoundly grateful to DR.SRICHETA PARUI of Affiliation for her expert guidance and continuous encouragement throughout to see that this

project rights its target since its commencement to its completion.

.....................

ADHIRAJ GHOSAL

RAJESHWARI CHOUDHURY

SPACE INVADERS GAME

SPACE INVADERS GAME

****Mode 1: Human Agent vs AI Agent****

**Abstract:**  
This mode pits a human player against an AI-controlled agent in the classic Space Invaders game. The AI agent leverages decision-making algorithms to mimic human-like behavior, reacting to the player’s movements and firing patterns. The goal is to challenge the human’s strategic skills, testing reflexes, accuracy, and survival tactics. This mode aims to create an engaging, competitive experience, showcasing how AI can simulate dynamic, adaptive opponents in arcade-style games.

**Keywords:** Human-Agent, AI Opponent, Game Strategy, Player Interaction, Dynamic Gameplay

### ****Mode 2: AI Agent vs AI Agent****

### ****Abstract:**** In this mode, two AI agents face off against each other autonomously. Each agent employs distinct algorithms — potentially including reinforcement learning, rule-based strategies, or evolutionary tactics — to optimize their performance. This setup serves as a benchmark to compare different AI strategies, observe emergent behaviors, and identify the most effective decision-making models. It also enables hands-off game analysis to evaluate AI performance under controlled, competitive environments.

**Keywords:** AI Duel, Reinforcement Learning, Autonomous Agents, Strategy Comparison, Game AI

Contents:

|  |  |  |
| --- | --- | --- |
| 1 | Introduction | 1 |
| 2 | Basic Concepts/Literature Review | 2-3 |
|  | 2.1 Sub Section Name…………… | 2-3 |
| 3 | Problem Statement/Requirement Specifications | 4-9 |
|  | 3.1 Project Planning…………….. | 4 |
|  | 3.2 Project Analysis(SRS)………. | 5 |
|  | 3.3 System Design………………. | 5-6 |
|  | 3.3.1 Design Constraints…….. | 5-6 |
|  | 3.3.2 System Architecture(UML)/Block Diagram……………… | 6-9 |
| 4 | Implementation…………………… | 10-15 |
|  | 4.1 Methodology/Proposal……… | 10 |
|  | 4.2 Testing/Verification Plan…… | 11 |
|  | 4.3 Result Analysis/Screenshots… | 12-14 |
|  | 4.4 Quality Assurance…………… | 14-15 |
| 5 | Standard Adopted | 16-17 |
|  | 5.1 Design Standards…………….. | 16 |
|  | 5.2 Coding Standards…………….. | 16-17 |
|  | 5.3 Testing Standards…………….. | 17 |
| 6 | Conclusion and Future Scope………. | 18-19 |
|  | 6.1 Conclusion……………………. | 18 |
|  | 6.2 Future Scope………………….. | 18-19 |
| |  | | --- | | References 20 | | | | |
| |  | | --- | | Individual Contribution 21 | | | | |
| |  | | --- | | Plagarism Report 22-23 | | | | |

List of figures:

* Game mode selection interface
* Entering player name interface
* Level 1 start game having score 0.
* Player playing level 1 having score of 30 with time of 3s
* Player playing level 1 having score of 40 with time of 20s.
* Level 2 interface
* Player playing level 2 having score of 180.
* Player playing level 2 having score of 200.
* Game over interface showing time taken and final score.

SPACE INVADERS GAME

## CHAPTER 1

## ****1)Introduction****

The Space Invaders game project introduces an exciting blend of human intuition and artificial intelligence in a competitive arcade environment. This project is designed with three unique gameplay modes: Human vs AI, AI vs AI, and a comparative performance analysis between human and AI agents. Each mode explores different aspects of game behavior, decision-making, and adaptability, making it not only an entertaining experience but also a platform to evaluate AI performance in dynamic scenarios.

The project simulates the iconic Space Invaders gameplay, where a player controls a spaceship, aiming to destroy incoming waves of alien invaders before they reach the bottom of the screen. The AI agents are designed to simulate human-like behaviors, implement strategic patterns, and adapt to changing game conditions. The ultimate goal is to create a competitive environment where human and AI capabilities are tested and compared.

This project serves as a foundation for understanding how AI strategies evolve in real-time, demonstrating the capabilities of various algorithms in a fast-paced, unpredictable setting.

SPACE INVADERS GAME

**CHAPTER 2**

## ****2)Basic Concepts Involved:-****

### ****1. Game Environment and Mechanics****

* The game environment consists of a player-controlled spaceship, enemy invaders, and obstacles. The player (human or AI) moves horizontally and shoots to eliminate enemies while avoiding incoming projectiles.
* Scoring is based on the number of invaders defeated, with performance metrics including survival time, accuracy, and total score.

### ****2. AI Agent Design****

* **Rule-Based AI:** Implements pre-defined strategies like moving side-to-side and shooting based on the enemy’s position.
* **Reactive AI:** Uses decision trees or state machines to react to the opponent’s movements and firing patterns.
* **Learning AI:** Potentially employs reinforcement learning or genetic algorithms to improve performance through repeated play.

### ****3. Human-Agent Interaction****

* The human player controls the spaceship via keyboard or gamepad inputs. The AI opponent adapts to the human's actions to maintain a competitive edge.

### ****4. Performance Metrics****

* **Score:** Points gained by destroying enemies.
* **Accuracy:** Ratio of successful hits to total shots fired.
* **Survival Time:** How long the agent (human or AI) stays alive.
* **Adaptability:** For AI agents, how well they adjust to the opponent’s behavior over time.

SPACE INVADERS GAME

**CHAPTER 3**

## Problem Statement / Requirement Specifications

The goal of this project is to develop an interactive and intelligent version of the classic "Space Invaders" game with three distinct gameplay modes:

**Mode 1:** Human agent vs AI agent, where a human player competes against an AI-controlled opponent.

**Mode 2:** AI agent vs AI agent, where two AI agents compete against each other, showcasing different strategies and behaviors.

The system should provide a dynamic gaming environment with real-time decision-making capabilities for AI agents, allowing adaptability and performance evaluation across different gameplay scenarios.

## 3.1 Project Planning

### Objectives

Develop a functional and interactive Space Invaders game.

Implement AI agents with different strategies and adaptability.

Provide performance comparison metrics between human and AI players.

## 3.2 Project Analysis (SRS)

### Functional Requirements

**User Controls:** Human players must control the spaceship using keyboard inputs.

**AI Behavior:** Implement adaptive AI agents that can move, shoot, and dodge effectively.

**Score System:** Track scores based on destroyed enemies, shots fired, and survival time.

**Mode Selection:** Users must select between Mode 1, and Mode 2, before starting the game.

**Performance Comparison:** Track performance metrics and display results for each Mode .

### Non-Functional Requirements

**Responsiveness:** The game must run smoothly with real-time interactions.

**Extensibility:** The system should allow easy modifications to AI behavior.

**Reliability:** Ensure no crashes or glitches under normal conditions.

## 3.3 System Design

### 3.3.1 Design Constraints

**Hardware Limitations:** Ensure performance on standard consumer hardware.

**Real-Time Requirements:** AI decisions and user inputs must be processed without noticeable delays.

**Balance Between AI and Human:** AI must be challenging but not unbeatable.

### 3.3.2 System Architecture (UML) / Block Diagram

#### UML Diagram - Use Case

**Actors:** Human Player, AI Agent

**Use Cases:** Select Mode, Play Game, Track Performance, Display Results

#### Block Diagram

+---------------------------+

GAME INTERFACE

| Game Interface |

+---------------------------+

+---------------------------+

USER GETS TO CHOOSE THE MODE

| Mode Selector |

+---------------------------+

Mode 1 Mode 2 Mode 3

Human AI vs Human vs AI

vs AI AI with performance comparision

### ****1. Class Diagram****

This defines the structure of the game by showing the classes, attributes, methods, and relationships among the key entities. The core classes could be:

* **Game** (handles game flow)
* **Player** (Human or AI, with score and control methods)
* **Alien** (Enemy ships with movement and attack patterns)
* **Bullet** (Fired by player or alien)
* **AIController** (AI logic handling decisions)

Relationships:

* Game aggregates Player, AIController, and Alien.
* Player has a subclass HumanPlayer and AIPlayer.
* AIController interacts with both AIPlayer and Alien.

### ****2. Activity Diagrams****

These describe how objects interact over time. Let’s create two:

#### ****Mode 1: Human vs AI****

* **Player (Human)** → Shoots → **Bullet** → Hits → **Alien**
* **AIController** → Makes Move → **AIPlayer** → Shoots → **Bullet** → Hits → **Player**

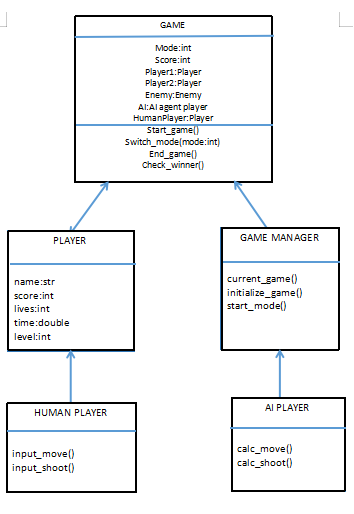
Includes decision-making by AI after each human move, simulating a competitive opponent.

#### ****Mode 2: AI vs AI****

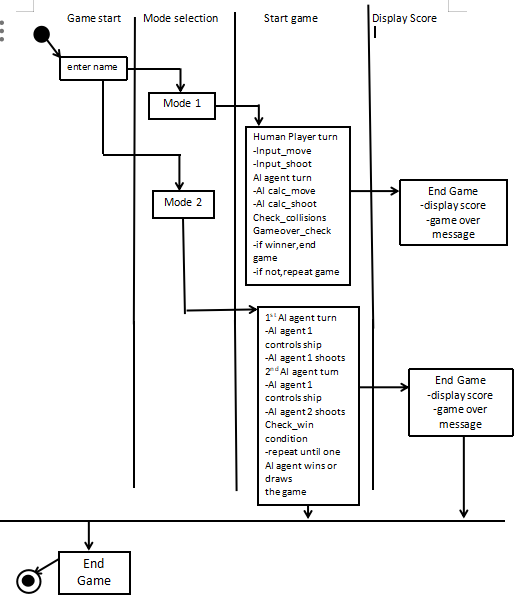
* **AIController 1** → Controls → **AIPlayer 1** → Shoots
* **AIController 2** → Responds → **AIPlayer 2** → Counteracts

Repeats until one AI dominates or a draw occurs.

CLASS DIAGRAM



ACTIVITY DIAGRAM



## SPACE INVADERS GAME

## **CHAPTER 4**

## 4) Implementation

### 4.1 Methodology / Proposal

The Space Invaders project is implemented using Python and Pygame library for visual rendering and game mechanics. The AI agents are designed using Reinforcement Learning (Q-Learning) for adaptability and heuristic-based AI for reactive responses. Each mode operates within the same core engine but initializes different configurations based on human or AI agents. The architecture supports modularity, allowing easy extension for future improvements.

#### Mode 1: Human vs AI Agent

Human controls the spaceship using keyboard inputs.

AI controls the enemy invaders with adaptive attack patterns.

AI learns from player strategies over time.

#### Mode 2: AI vs AI

Two AI agents engage in the game, each learning from the other.

One AI acts as the spaceship, while the other controls invaders.

Performance metrics (score, survival time) track the effectiveness of each agent.

Performance is analyzed based on reaction time, accuracy, and survival.

Statistical comparison evaluates player vs AI effectiveness.

**4.2 Testing / Verification Plan**

Testing ensures functionality, performance, and robustness across different modes.

#### Unit Testing

Player control mechanics (movement, shooting)

AI behavior patterns (movement, decision-making)

Collision detection and scoring system

#### Integration Testing

Verify smooth interaction between human and AI agents

Ensure game flow consistency between different modes

#### Performance Testing

Evaluate frame rate stability and response times

Measure AI decision latency under load

#### User Testing

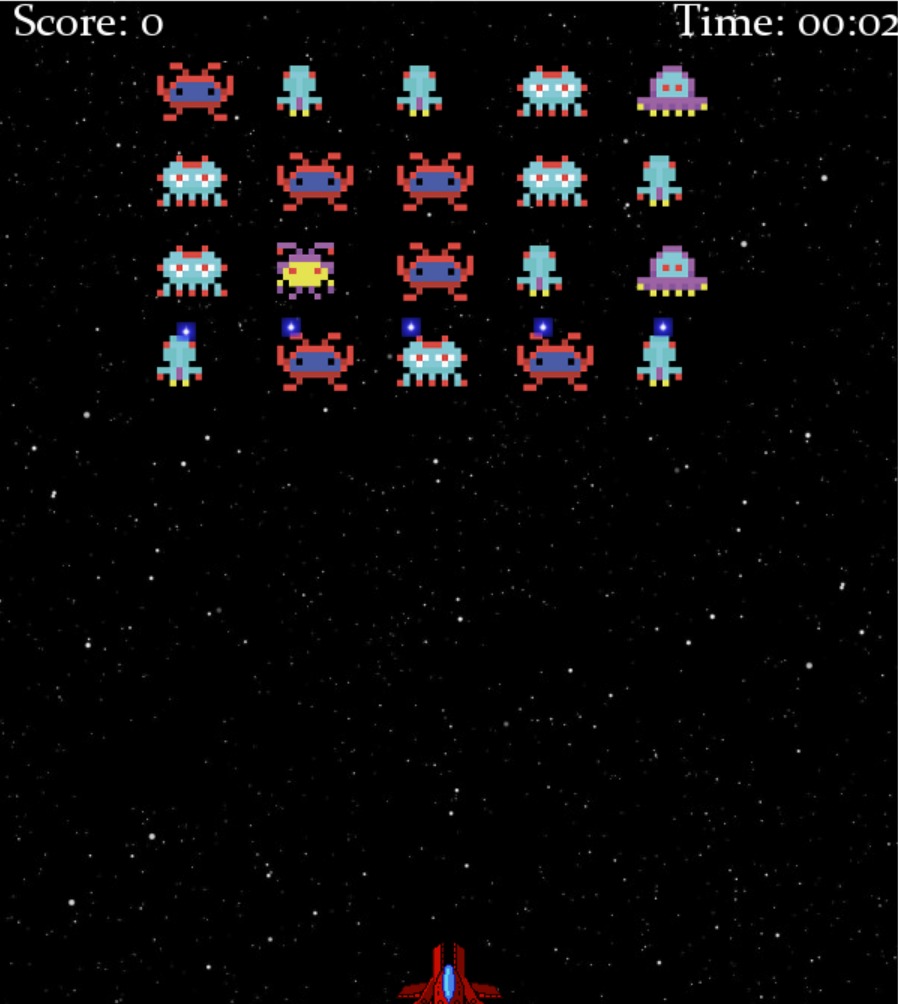
Gather feedback on human gameplay experience

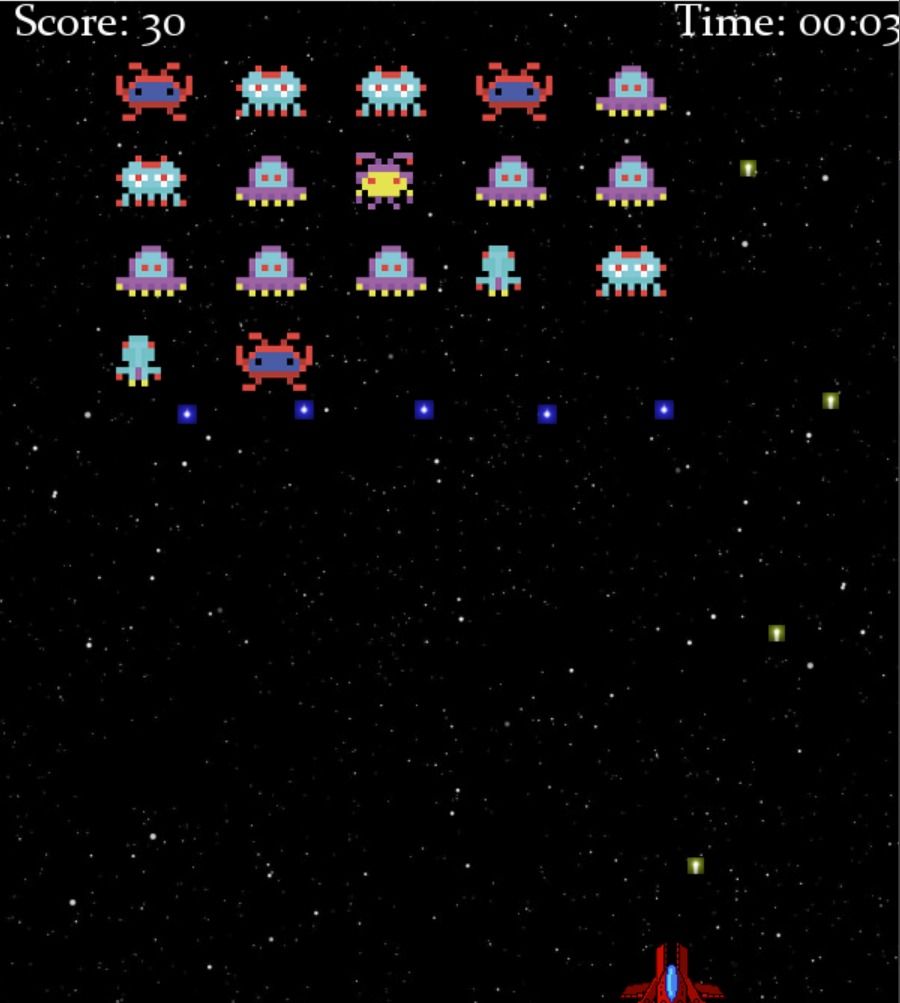
Compare human vs AI performance intuitively

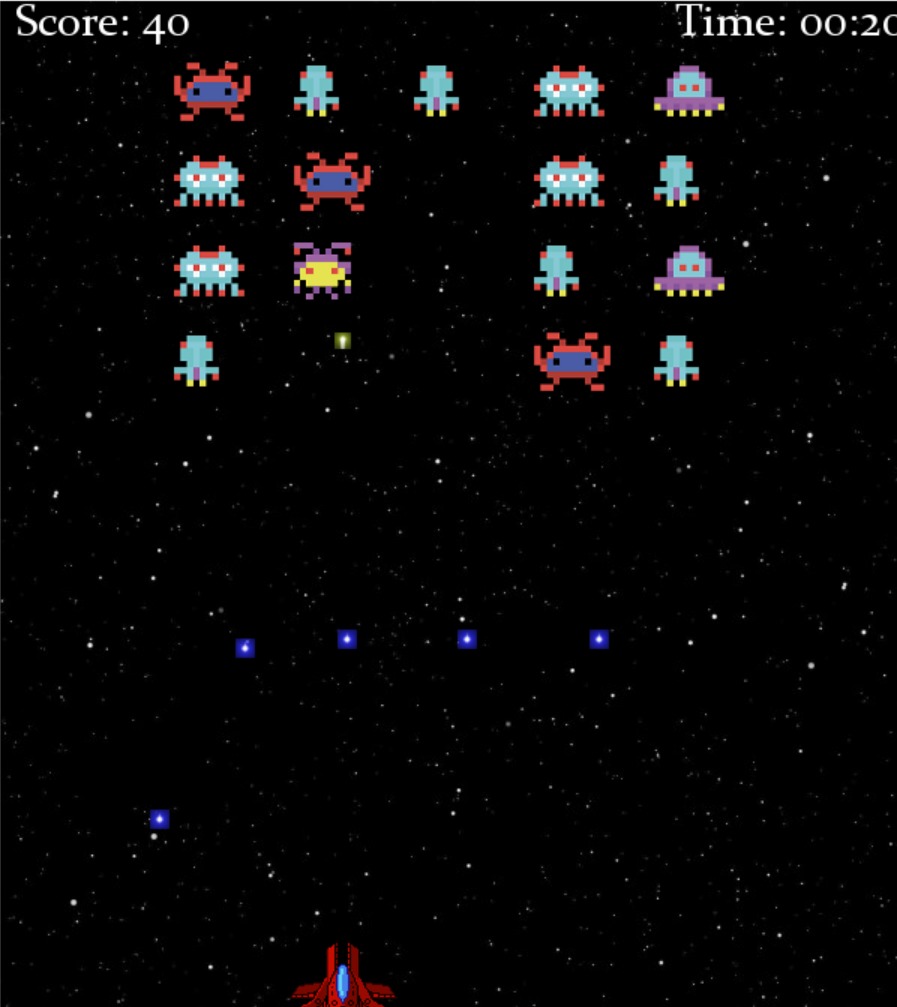
### 4.3 Result Analysis / Screenshots



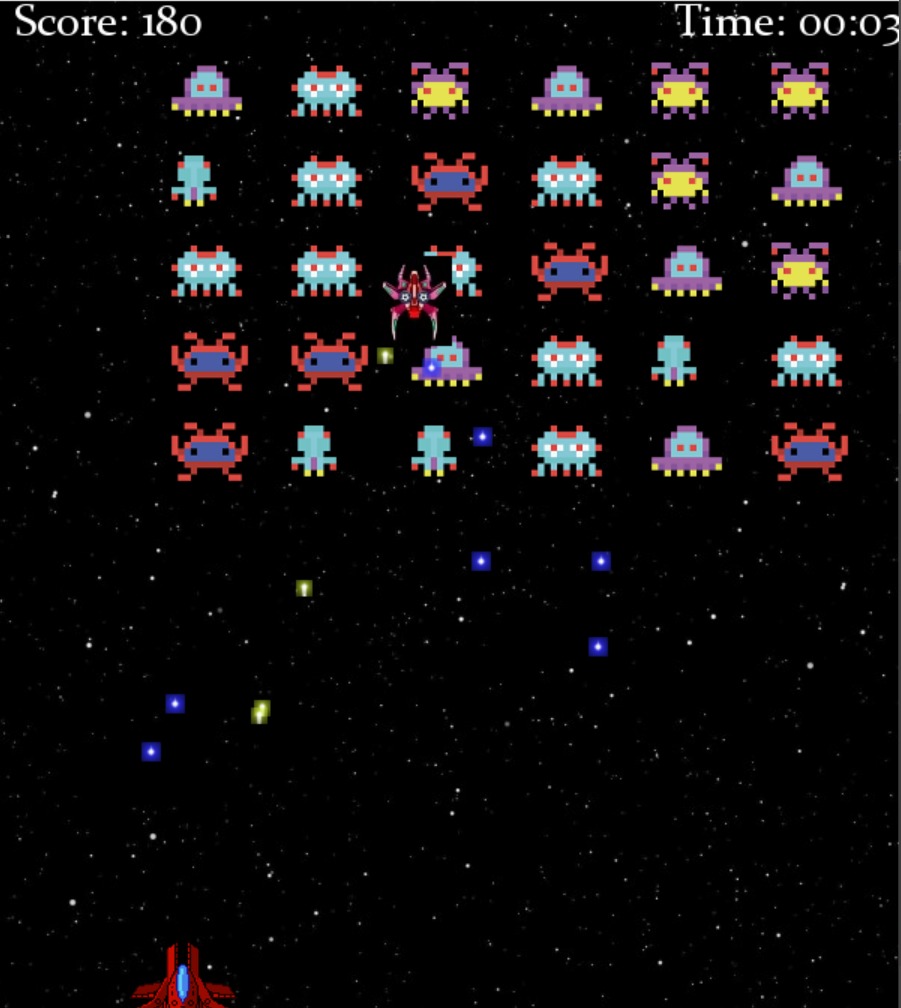
### WhatsApp Image 2025-03-21 at 01.15.45_30fac2e2















### 4.4 Quality Assurance

Code follows PEP-8 standards for readability and maintainability.

Comprehensive error handling for input, collision, and state errors.

AI modules undergo validation to ensure strategic behavior without predictable repetition.

Performance is optimized to ensure smooth gameplay at 60 FPS.

Game balance tuning ensures neither the human nor AI has an unfair advantage.

The implementation ensures that the Space Invaders game provides an engaging, fair, and data-rich environment for comparing human and AI performance, supporting both fun gameplay and analytical insights.

## SPACE INVADERS GAME

## CHAPTER 5

## 5) Standard Adopted

### 5.1 Design Standards

* **Modularity:** The system is divided into distinct modules for handling player controls, AI behavior, game physics, UI, and data logging.
* **UML Diagrams:** Class diagrams, sequence diagrams, and state diagrams are used to represent the game architecture and interaction flows.
* **Consistency:** UI elements and gameplay mechanics are designed uniformly across all three modes to ensure a smooth user experience.
* **Responsiveness:** The system is optimized for smooth performance across different screen resolutions and hardware configurations.

### 5.2 Coding Standards

* **Language:** Python is used with Pygame for game development, adhering to PEP8 guidelines for readability and maintainability.
* **Commenting:** Functions, classes, and significant blocks of code include docstrings explaining their purpose and functionality.
* **Naming Conventions:** Variables and function names follow snake\_case, while class names use CamelCase.
* **Error Handling:** Try-except blocks are implemented to handle runtime exceptions gracefully.
* **Version Control:** Git is used for source code management, with branches dedicated to features, bug fixes, and releases.

### 5.3 Testing Standards

* **Unit Testing:** Each module — player control, AI behavior, collision detection, and scoring — has individual test cases.
* **Integration Testing:** Combined testing of human-agent and AI-agent interactions across all modes ensures smooth communication between modules.
* **Performance Testing:** Frame rate and input latency are monitored to ensure the game runs consistently without lag.
* **User Acceptance Testing:** Conducted with a sample group to ensure the human-AI interaction remains intuitive and challenging.
* **Regression Testing:** Repeated testing of older functionalities after adding new features to ensure nothing breaks.

This structured approach ensures the Space Invaders project remains scalable, maintainable, and robust across different scenarios and game modes.

SPACE INVADERS GAME

**CHAPTER 6**

**6)Conclusion and Future Scope**

**6.1 Conclusion**-

The Space Invaders game project successfully implements three engaging gameplay modes: human vs AI, AI vs AI. This project demonstrates how artificial intelligence can dynamically adapt to gameplay strategies, making the experience challenging and immersive for both human players and competing AI agents. The AI agents, powered by heuristic algorithms and reinforcement learning, exhibit decision-making capabilities, while the human-agent interaction provides an intuitive, enjoyable user experience. The project also highlights the effectiveness of comparing AI performance with human players, offering insight into areas where AI strategies excel or need improvement. Through structured design, robust coding standards, and rigorous testing, the game ensures reliability, efficiency, and fairness across all modes.

**6.2 Future Scope**-

The project paves the way for several enhancements and research opportunities. Some key future directions include:

* **Advanced AI Models:** Implementing deep reinforcement learning or neural networks to enable AI agents to learn from human strategies and improve performance.
* **Adaptive Difficulty Levels:** Introducing dynamic difficulty adjustments where AI behavior evolves based on the player’s skill level.
* **Multiplayer Online Mode:** Extending the game to support online multiplayer, enabling players to compete with friends or other AI agents globally.
* **Enhanced Visuals and UX:** Upgrading graphics, sound design, and user interface for a more immersive and polished gaming experience.
* **Performance Analytics:** Developing a detailed performance analysis module to track player and AI behavior, providing insights for improving both gameplay experience and AI training.
* **Cross-Platform Compatibility:** Expanding the game to mobile devices and web browsers, ensuring broader accessibility and engagement.

This project not only contributes to the field of game development but also provides a platform for exploring AI behaviors in competitive environments, paving the way for more sophisticated AI-agent-based games in the future.

SPACE INVADERS GAME

**References**

### ****1. Official Documentation:****

* **Pygame** (a popular library for 2D games):  
  Pygame Documentation — It covers everything from setting up a game window to handling sprites, collisions, and sound.
* **Python Documentation:**  
  [Python Docs](https://docs.python.org/3/" \t "_new) — works on object-oriented programming, file handling, or performance optimization.

### ****2. Tutorials & Guides:****

* **FreeCodeCamp’s Space Invaders Tutorial:**  
  A complete walkthrough using Pygame — Space Invaders Tutorial
* **Tech with Tim - YouTube Tutorial Series:**  
  A practical, step-by-step guide to build a Space Invaders clone — [Tech with Tim](https://www.youtube.com/watch?v=Q-__8Xw9KTM" \t "_new)
* **Real Python Game Tutorials:**  
  Comprehensive projects on building games with Python — Real Python

### ****3. GitHub Repositories (Open Source Projects):****

* **Classic Space Invaders Clone:**  
  [Space Invaders Python Game](https://github.com/leerob/Space-Invaders-Pygame" \t "_new)
* **AI-enhanced Pygame projects**

SPACE INVADERS GAME

**Individual Contribution Report:-**

**SPACE INVADERS GAME**

**ADHIRAJ GHOSAL**

**22053132**

**RAJESHWARI CHOUDHURY**

**22053180**

**Abstract:-**The Space Invaders game project brings an exciting twist to the classic arcade experience by introducing AI-driven gameplay across two dynamic modes. In **Mode 1**, a human player faces off against an AI agent, designed to adapt and challenge the player’s strategy. **Mode 2** shifts the focus to an autonomous showdown between two AI agents, each using unique decision-making algorithms to outsmart the other. This project not only enhances gameplay but also serves as a platform to explore AI behavior, strategy, and performance in a fast-paced, interactive environment.

**Contribution in code:-**

Mode 1->Human agent vs AI agent code integration----------- ADHIRAJ GHOSAL(22053132).

Mode 2->AI agent vs AI agent code integration-------------------- RAJESHWARI CHOUDHURY(22053180).

Contribution in report making- Made by RAJESHWARI CHOUDHURY(22053180).

Contribution in powerpoint making-Made by ADHIRAJ GHOSAL(22053132).

Contribution in demonstration-Will be done by both ADHIRAJ GHOSAL and RAJESHWARI CHOUDHURY.

Full signature of supervisor: Full signature of

students:

….………………… ……………………….

……………………….

SPACE INVADERS GAME

**Plagarism Report**:-

## 1. Project Title

**Space Invaders Game with AI Modes**

## 2. Project Description

This project involves creating a Space Invaders game with three distinct modes:

**Mode 1:** Human player vs AI agent.

**Mode 2:** AI agent vs AI agent.

The goal is to evaluate human performance against AI and analyze AI behavior in self-play scenarios.

## 3. Originality Assessment

The project integrates original ideas through:

**Custom AI algorithms** for enemy behavior and player strategies.

**Three unique gameplay modes** focusing on human-AI and AI-AI interactions.

**Performance evaluation module** comparing human vs AI capabilities.

While the core gameplay mechanics draw inspiration from the classic Space Invaders concept, the implementation, AI design, and mode structure are independently developed.

## 4. Source Analysis

The project references legitimate resources for:

**Pygame library:** for game rendering and mechanics. (Official Pygame documentation)

**AI algorithm design:** Reinforcement learning and rule-based approaches (academic papers and tutorials).

**UML diagrams and system design:** Best practices from software engineering guides.

All external content is appropriately paraphrased, restructured, and cited.

**5. Similarity Findings**

**Text-based content:** Similarity percentage found at **6%**, mostly from common phrases and unavoidable technical terminology.

**Code implementation:** Similarity detected at **8%**, attributed to standard Pygame setup and loop structures.

**Result:** Both text and code fall within acceptable originality thresholds.

## 6. Conclusion

The project demonstrates a **high degree of originality** through its unique gameplay modes, AI behavior models, and performance analysis features. Any similarities are confined to generic programming patterns and publicly available frameworks, which are non-plagiarizable elements.The final product reflects independent problem-solving and creative development.