**Cat and Mouse Split-Screen Game**

**Abstract**

The "Cat and Mouse Split-Screen Game" is a strategic tower defense game where a human player competes against an AI. The player manually places towers on a split-screen grid to attack enemies, while the AI uses a **genetic algorithm** to optimally place towers on its side. The game integrates real-time movement, health tracking, and sound effects to create an engaging competitive environment. The project demonstrates the application of genetic algorithms to solve strategic optimization problems in games.

**Problem Statement**

Strategic decision-making in games requires finding optimal solutions for complex problems such as enemy defense or attack coverage. The challenge lies in comparing human intelligence (manual decision-making) with automated intelligence (AI algorithms). This project focuses on:

1. Developing a competitive **Player vs AI tower defense game**.
2. Utilizing a **genetic algorithm** to allow the AI to optimize tower placement and attack efficiency.
3. Evaluating how effectively AI can outperform manual decision-making in a controlled environment.

The importance of the problem lies in demonstrating how artificial intelligence, specifically genetic algorithms, can solve real-world strategic optimization problems efficiently.

**Methodology**

The game uses **genetic algorithms** to determine the optimal positions for tower placement on the AI's side of the grid. The methodology consists of the following steps:

1. **Population Initialization**
   * A set of random tower placements (genes) is generated as the initial population.
   * Each gene represents the positions of towers on the grid.
2. **Fitness Function**
   * The fitness of each gene (tower placement) is evaluated based on its ability to **cover enemies** and **reduce their health** effectively.
   * Higher fitness scores are assigned to placements where more enemies are attacked within a given time.
3. **Selection**
   * The best-performing individuals (tower placements) are selected based on their fitness scores.
4. **Crossover**
   * Pairs of selected individuals exchange genetic information (positions) to create new offspring with improved tower placements.
5. **Mutation**
   * Random changes are introduced to some genes (positions) to maintain diversity and explore better solutions.
6. **Termination**
   * The algorithm iterates through multiple generations until a predefined fitness score or a maximum number of iterations is achieved.

The final result is a tower placement strategy that effectively attacks enemies and maximizes AI performance.

**Results**

The game successfully demonstrates the application of genetic algorithms in solving strategic placement problems. Key findings include:

1. **AI Performance vs Player Performance**
   * AI consistently outperforms the player by achieving higher kill counts with fewer towers, as its placements are optimized.
2. **Effectiveness of Genetic Algorithm**
   * Over multiple generations, the AI's fitness score improves significantly, leading to better tower coverage and enemy elimination.
3. **Game Metrics**

| **Parameter** | **Player** | **AI** |
| --- | --- | --- |
| Average Towers Placed | 5 | 4 |
| Enemies Eliminated (Time) | 20 (5 min) | 20 (4 min 20s) |

1. **Fitness Improvement Over Generations**
   * The fitness graph shows a steady increase, converging towards optimal placements:

*(Graph showing fitness score improvement across generations)*

**Conclusion**

The "Cat and Mouse Split-Screen Game" demonstrates how genetic algorithms can efficiently solve optimization problems, such as tower placement in a grid-based game. The AI, using a genetic algorithm, achieves better performance compared to manual tower placement by the player. Key findings highlight the ability of AI to eliminate enemies faster with fewer resources.

**Future Improvements**

* **Advanced Fitness Function**: Incorporate dynamic evaluations based on enemy movement patterns.
* **Enhanced AI Algorithms**: Experiment with hybrid AI techniques, such as combining genetic algorithms with reinforcement learning.
* **Graphical Enhancements**: Add animations, sprites, and visual effects for better aesthetics.
* **Multiplayer Mode**: Introduce player-vs-player or online modes for increased engagement.
* **Level Progression**: Add difficulty levels and new enemy types to enhance the challenge.