**Pucket: An Innovative Twist on a Classic Game**

**Project Information**

* **Submitted By**: Hamaiz Siddiqui (22k-4682), Abdul Wadood (22k-4764), Zeerak Shahzad (22k-4692)
* **Course**: AI
* **Instructor**: Ms. Mehak Mazhar
* **Last Updated**: May 2025

**Overview**

Pucket is a fast-paced, competitive game where players use a wooden board with holes to launch pucks into their opponent's side. This implementation features a computerized version where players compete against an AI opponent that uses strategic decision-making to determine optimal moves.

**Game Features**

**Current Implementation**

The current stable version of the game includes:

* **Core Gameplay**: Launch pucks across the board and aim for the goal on your opponent's side
* **Physics Engine**: Realistic puck movement with friction, collisions, and boundary interactions
* **Turn-Based System**: Players take turns launching their pucks
* **AI Opponent**: Computer player that strategically aims for goals and adds randomness for unpredictability
* **Scoring System**: Keep track of how many pucks each player has successfully scored
* **Win Condition**: First player to get all their pucks into the opponent's goal wins

**Planned Extensions**

Future versions will gradually reintroduce more complex features:

* **Power-Ups**: Speed boosts, shields, freeze effects, and multi-launch capabilities
* **Obstacles**: Bumpers, portals, and walls that affect puck trajectories
* **Advanced AI**: Enhanced opponent using Minimax algorithm with Alpha-Beta pruning

**Technical Implementation**

**Core Components**

* **Game Engine**: Built with Pygame for graphics, input handling, and game loop management
* **Physics System**: Custom implementation for puck movement, collisions, and boundary interactions
* **AI Strategy**: Decision making system that evaluates board state and calculates optimal moves
* **Turn Management**: System to handle alternating turns with proper transition logic

**Game States**

1. **Initialization**: Set up board, players, pucks, and game objects
2. **Player Turn**: Handle player input for aiming and launching pucks
3. **AI Turn**: Calculate optimal move and execute AI action
4. **Physics Update**: Apply movement and collision rules to all game objects
5. **Win Condition Check**: Determine if the game has been won
6. **Game Over**: Display results and offer restart option

**How to Play**

**Requirements**

* Python 3.x
* Pygame library

**Installation**

1. Make sure you have Python installed on your system
2. Install Pygame: pip install pygame
3. Download the game files
4. Run the game: python pucket\_game.py

**Controls**

* **Mouse**: Aim and launch pucks (click and drag to set direction and power)
* **Tab**: Switch between your active pucks
* **F3**: Toggle debug mode (shows additional game state information)
* **R**: Restart the game (after game over)

**Gameplay Tips**

1. **Aim Carefully**: The precise angle and power of your launch is crucial
2. **Plan Ahead**: Consider where your puck will end up after collisions
3. **Use Rebounds**: Sometimes bouncing off walls or other pucks is the best strategy
4. **Watch the AI**: Learn from your opponent's tactics

**Troubleshooting**

**Common Issues**

* **Game Freezes**: If the game becomes unresponsive, try restarting it
* **Graphics Issues**: Make sure your Python and Pygame installations are up to date
* **Performance Problems**: Close other resource-intensive applications while playing

**Debug Mode**

Press F3 during gameplay to enable debug mode, which displays:

* Current player information
* Turn state and timing
* Puck velocity indicators
* Frame rate and performance metrics

**Development Roadmap**

**Phase 1: Core Gameplay (Current)**

* Basic board, pucks, and goals
* Simple physics engine
* Turn-based gameplay
* Basic AI opponent

**Phase 2: Enhanced Features (Upcoming)**

* Power-ups system
* Obstacles implementation
* Improved physics interactions
* More sophisticated AI

**Phase 3: Advanced AI (Future)**

* Minimax algorithm with Alpha-Beta pruning
* Heuristic evaluation functions
* Learning components
* Difficulty levels

**Credits**

This project was developed as part of the AI course requirements, implementing game theory concepts and artificial intelligence techniques in an interactive game environment.

**License**

This project is for educational purposes only.