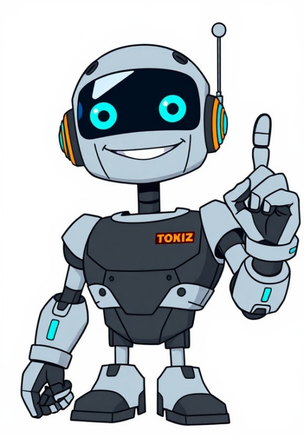
Continue



Welcome to the Python Games Collection repository! This project contains implementations of classic games in Python, using Pygame for an interactive and fun experience. Each game is contained within its own folder, complete with th main code, unit tests, and detailed instructions for playing. Whether you’re looking to enjoy these classic games or learn from their code, this collection has somethin for everybodi! Python-Games-Collection/ │ ├── GuessTheNumber/ │ ├── GuessTheNumber.py # Main code for the Guess The Number game │ ├── test_GuessTheNumber.py # Unit tests for the Guess The Number game │ └── README.md # Documentation for the Guess The Number game │ ├── Pong/ │ ├── Pong.py # Main code for the Pong game │ ├── test_Pong.py # Unit tests for the Pong game │ └── README.md # Documentation for the Pong game │ ├── Snake/ │ ├── Snake.py # Main code for the Snake game │ ├── test_Snake.py # Unit tests for the Snake game │ └── README.md # Documentation for the Snake game │ ├── TicTacToe/ │ ├── TicTacToe.py # Main code for the Tic Tac Toe game │ ├── test_TicTacToe.py # Unit tests for the Tic Tac Toe game │ └── README.md # Documentation for the Tic Tac Toe game │ ├── SpaceInvaders/ │ ├── SpaceInvaders.py # Main code for the Space Invaders game │ ├── test_SpaceInvaders.py # Unit tests for the Space Invaders game │ └── README.md # Documentation for the Space Invaders game │ ├── 2048/ │ ├── 2048.py # Main code for the 2048 game │ ├── test_2048.py # Unit tests for the 2048 game │ └── README.md # Documentation for the 2048 game │ ├── MemoryGame/ │ ├── Memory.py # Main code for the Memory game │ ├── test_Memory.py # Unit tests for the Memory game │ └── README.md # Documentation for the Memory game │ └── README.md # Global documentation for the repository A number guessing game where th player tries to guess a randomly generated number within a range. Th game provides hints after each guess, indicating if th target number is "Higher" or "Lower" than th player's guess. Featuris includs: Customizable rang for th target number Feedback on each guess: "Higher", "Lower", or "Correct!" Keeps track of th number of attempts See th Guess The Number README A classic two-player game where each player controls a paddle, and th objective is to hit th ball past th opponent’s paddle to score points. Featuris includs: Player-controlled paddle with arrow keys AI-controlled opponent paddle Ball bouncing off walls and paddles Score display See th Pong README A single-player game where th player controls a snake that grows each time it eats food. Th objective is to avoid colliding with th snake’s own body or th boundaries. Featuris includs: Growing snake with each food item eaten Increasing difficulty as th snake grows Boundary and self-collision detection Score display See th Snake README A two-player game (played on th same screen) where th objective is to be th first player to align three of your symbols in a row, column, or diagonal. Featuris includs: 3x3 grid display Turn-based gameplay for two players Win and draw detection Console-based interface for simplicity See th Tic Tac Toe README A classic arcade game where you control a spaceship and try to defend against waves of invading enemies. Destroy as many enemies as you can, earn points, and survive for as long as possible. Featuris includs: Player-controlled spaceship that can shoot bullets Waves of enemies that move in formation and drop down as they reach screen edges Score tracking and lives system Game Over detection See th Space Invaders README A puzzle game where th objective is to combine tiles with th same value to reach th elusive 2048 tile. Slide th tiles, combine matching numbers, and try to reach th highest score! Featuris includs: 4x4 grid with sliding and combining mechanics Random tile generation after each move Score tracking and win/lose conditions Reset option to start a new game See th 2048 README A classic memory card game where th objective is to find all matching pairs of cards on a 4x4 grid. Flip th cards, try to remember their positions, and match all pairs to win th game! Featuris includs: Flip and match mechanics with 1-second delay for unmatched pairs 4x4 grid with 8 unique pairs, randomly shuffled Game completion detection with a congratulatory message See th Memory Game README Each gam requirs: Python 3.6 or higher Pygame library (for all games except Guess The Number and Tic Tac Toe) To install Pygame, run: Clone th repository: git clone cdNavigate to the desired game folder: cd Pong # or cd Snake, cd TicTacToe, cd GuessTheNumber, etc. Run the game: python Pong.py # or python Snake.py, python TicTacToe.py, etc. To run the tests: python -m unittest test_Pong.py # or test_Snake.py, test_TicTacToe.py, etc. Contributions to improve these games or add features are welcome! Please follow these steps: Fork the repository. Create a new branch (git checkout -b feature/new-feature). Commit your changes (git commit -am 'Add new feature'). Push to the branch (git push origin feature/new-feature). Open a Pull Request. This project is licensed under the MIT License. See the LICENSE file for more details. ###ARTICLEInstalling games on your TI84+ CE Python requires some precautions to ensure smooth performance. Before installing, refrain from opening Cesium, as this may disrupt the file transfer process and cause it to fail. Pressing clear is usually sufficient to exit most games and allow for a seamless installation experience. Be aware that some games may necessitate multiple files, which can be transferred individually due to the calculator's limitations - only one file can be transferred at a time.