**Space invaders: Text Edition:** - Python Game Documentation and development

This document provides an overview and instructions for playing the "Space Invaders: Text Edition" game uniquely written in Python as a simple sample you can not find anywhere else for our clients to testify to our unique programming experience. Clean code testify to our adherence to global best practices and clear and unanimous documentation testify to our comprehensive and easy approach using all software engineering principles best suited for the project.

**Game Description**

This is a simple, text-based game where you control a spaceship and defend against an invading enemy spaceship.

**Gameplay**

1. The game board is represented by a 2D grid of characters displayed in the console.

2. You control your spaceship ("@") using the keyboard's left and right arrow keys.

3. Fire a laser ("|") towards the enemy ("#") by pressing the "f" key.

4. Destroy the enemy before it reaches the bottom of the screen to score points.

5. The game ends when the enemy collides with your spaceship.

**Playing the Game**

1. Save the provided Python code as `space\_invaders.py`.

2. Run the script from your terminal:

```bash

python space\_invaders.py

```

3. Follow the on-screen instructions to control your spaceship and fire lasers.

**Additional Notes**

1. The game utilizes basic text characters to represent the game elements.

2. The enemy moves down one space every turn.

3. Firing the laser involves a visual animation by repeatedly updating the game board.

4. Scoring and difficulty levels are not implemented in this basic version. This gives room for upgrade

**Customization and upgrade**

This code provides a foundation for further development. You can explore adding features like:

1. Multiple levels with increasing difficulty and faster enemy movement.

2. Power-ups that enhance your spaceship's capabilities.

3. Different enemy types with varying behavior and difficulty levels.

4. Sound effects to add to the gameplay experience.

**Disclaimer**

This game is provided as a sample from compusolvetechnologies@gmail.com and remains only for sampling purposes only for you to attest to our programming strength and see how our imagination crumbles big problems into simple crumps by providing solutions. It is however intended to demonstrate basic game development concepts in Python and may not be fully polished or feature-rich.

**Code snippets for the game:**

import random

# Define the game board size

BOARD\_WIDTH = 20

BOARD\_HEIGHT = 10

# Define initial player and enemy positions

player\_x = BOARD\_WIDTH // 2

player\_y = BOARD\_HEIGHT - 1

enemy\_x = random.randint(0, BOARD\_WIDTH - 1)

enemy\_y = 0

# Define symbols

PLAYER = "@"

ENEMY = "#"

LASER = "|"

EMPTY = " "

# Initialize the game board

board = []

for \_ in range(BOARD\_HEIGHT):

board.append([EMPTY] \* BOARD\_WIDTH)

# Function to display the game board

def display\_board():

for row in board:

print(" ".join(row))

# Function to move the player

def move\_player(direction):

global player\_x

if direction == "left" and player\_x > 0:

player\_x -= 1

elif direction == "right" and player\_x < BOARD\_WIDTH - 1:

player\_x += 1

# Function to fire a laser

def fire\_laser():

global laser\_y

laser\_y = player\_y - 1

while laser\_y >= 0:

board[laser\_y][player\_x] = LASER

laser\_y -= 1

display\_board()

# Simulate delay

time.sleep(0.1)

board[laser\_y + 1][player\_x] = EMPTY

# Function to move the enemy

def move\_enemy():

global enemy\_y

if enemy\_y == BOARD\_HEIGHT - 1:

# Game over

print("Game Over! You were invaded!")

exit()

enemy\_y += 1

# Function to check for collision

def check\_collision():

if board[enemy\_y][enemy\_x] == LASER:

global enemy\_x, enemy\_y

enemy\_x = random.randint(0, BOARD\_WIDTH - 1)

enemy\_y = 0

print("Enemy hit! Score +1")

# Game loop

while True:

# Clear the screen (optional)

# import os

# os.system('cls')

# Display the game board

display\_board()

# Get user input

direction = input("Enter 'a' for left, 'd' for right, or 'f' to fire: ")

# Move the player

move\_player(direction)

# Update the board

board[player\_y][player\_x] = PLAYER

board[enemy\_y][enemy\_x] = ENEMY

# Check for laser collision

if direction == "f":

fire\_laser()

check\_collision()

# Move the enemy

move\_enemy()

# Reset the board for the next iteration

board[player\_y][player\_x] = EMPTY

board[enemy\_y][enemy\_x] = EMPTY