**Submission Checklist for Python Game Project: Gun Arena**

**SUBMITTED BY:**

* Ayush Goyal(MABSPG24074)
* Rishaank Rao(MABSPG24070)
* Sanskar Tyagi (MABSPG24112)
* Muskan Sachdeva (MABSPG24103)
* Ishvi Soni (MABSPG24111)

**1. Python Script(s)**

All the code for the game is written in a single Python script file.

This script includes:

* Game window setup with background image and music
* Two-player character creation and controls
* Health bar and bullet mechanics
* Collision detection and shooting
* Winner announcement and game loop restart

Additional assets required:

* **Two player images**
* **Background image**
* **Background music file**

**2. Project Description, Setup Instructions, and Dependencies**

**Project Description:**

**Gun Arena** is a visually engaging two-player shooting game developed using Python and the Pygame library.

**Objective:** Each player controls a unique fighter and attempts to deplete the other player's health by shooting bullets. The game ends when one player's health reaches zero.

**Gameplay Features:**

* Colourful player avatars and animated bullets
* Aesthetic background image with looping background music
* Health bars for both players
* Dynamic movement and real-time bullet firing
* Player name display with stylish banners
* Win condition with display and game restart

**Setup Instructions:**

**Step 1: Open Jupyter Notebook**

Open Jupyter notebook using Anaconda Prompt.

**Step 2: Install Pygame**

Run the following command:

**“pip install pygame”**

**Step 3: Add Game Assets**

Place the following assets in accessible local directories:

* Player images (e.g., player1.jpg, player2.jpg)
* Background image (e.g., arena\_bg.jpg)
* Background music file (e.g., background\_music.mp3)

Update the file paths in the code accordingly.

**Step 4: Run the Game**

Save the code as gun\_arena.py and run:

python gun\_arena.py

**Dependencies:**

| **Dependency** | **Description** |
| --- | --- |
| Python | Main programming language |
| Pygame | For rendering game graphics, audio, and real-time controls |

**3. Key Features and Functionality**

**Game Features:**

| **Feature** | **Description** |
| --- | --- |
| Two-Player Mode | Both players control separate fighters using different keys |
| Bullet Mechanics | Players shoot projectiles in real-time |
| Health Bar | Tracks each player’s remaining health |
| Collision Logic | Health decreases on bullet hit |
| Background Theme | Custom image and music for immersive gameplay |
| Player Names | Displayed in the center with golden-red styling |
| Win Message | Displays winner text and resets game after delay |

**Functional Highlights (Code Overview):**

* Fighter Class  
  Manages movement, health, shooting, and rendering.
* handle\_bullets()  
  Handles bullet movement and collision logic.
* draw\_window()  
  Draws background, player names, fighters, bullets, and health bars.
* display\_winner()  
  Shows winning player and restarts game loop.
* main()  
  Handles event loop, player actions, and game flow.

**User Interface & Controls:**

| **Key** | **Player 1 Action** | **Player 2 Action** |
| --- | --- | --- |
| W/A/S/D | Move Up/Left/Down/Right | - |
| Arrow Keys | - | Move Up/Left/Down/Right |
| G | Shoot | - |
| K | - | Shoot |

**UI Aesthetics:**

* **Background:** Custom image
* **Health Bars:** Red and Green color scheme
* **Player Sprites:** Unique character images
* **Music:** Looping background soundtrack
* **Player Labels:** Golden-red banners with names

**Conclusion:**

**Gun Arena** is a dynamic and engaging two-player game that blends core programming logic with strong visual appeal. It showcases:

* Object-oriented programming
* Real-time input handling
* Collision detection
* Multimedia integration with images and sound

This project helps reinforce concepts of game loops, class structures, event-driven programming, and multimedia processing in Python.