Apex Legends Events Tracker - Overwolf App

This Overwolf Native app tracks Apex Legends game events and sends them to a Python server for processing.

Project Structure

```
apex-events-tracker/
├── manifest.json          # Overwolf app manifest
├── background.html        # Background page
├── background.js          # Main event handling logic
├── in_game.html           # In-game overlay
├── desktop.html           # Desktop window (optional)
├── IconMouseNormal.png    # App icon
├── IconMouseGray.png      # Grayed icon
├── desktop-icon.ico       # Desktop icon
├── python_server.py       # Python event receiver
└── requirements.txt       # Python dependencies
```

Setup Instructions

Part 1: Python Server Setup

1. **Install Python dependencies:**

bash

```
pip install -r requirements.txt
```

2. **Run the Python server:**

bash

```
python python_server.py
```

The server will start on `http://localhost:5000` and wait for events.

Part 2: Overwolf App Setup

1. **Enable Overwolf Developer Mode:**
   - Open Overwolf settings
   - Go to "About" tab
   - Click "Development Options"
   - Enable "Developer Mode"
2. **Load the app as unpacked extension:**
   - In Overwolf, click the wrench icon (Developer mode)
   - Click "Load unpacked extension"
   - Select the folder containing your manifest.json
3. **Create placeholder icons:** You need to create three icon files:
   - `IconMouseNormal.png` (256x256 px)
   - `IconMouseGray.png` (256x256 px)
   - `desktop-icon.ico` (ICO format)
4. **Launch Apex Legends:**
   - Start the game
   - The Overwolf app will automatically detect it
   - Events will start flowing to your Python server

Features

Tracked Events:

- **Kills** - When you eliminate an enemy
- **Deaths** - When you die
- **Assists** - When you assist in kills
- **Damage** - Damage dealt to enemies
- **Knockdowns** - When you knock down an enemy
- **Headshots** - Headshot kills
- **Revives** - When you revive teammates
- **Match state** - Match start, end, etc.
- **Location** - Player position updates
- **Inventory** - Inventory changes
- **Team/Roster** - Team information

Python Server Endpoints:

- **POST /events** - Receives events from Overwolf (automatically called)
- **GET /events** - Retrieve all stored events
- **GET /events/stats** - Get statistics about events
- **POST /events/clear** - Clear all stored events

How It Works

1. The Overwolf app runs in the background and registers for Apex Legends events
2. When Apex Legends launches, the app starts listening for game events
3. Each event is captured and sent to the Python server via HTTP POST
4. The Python server logs events to console and saves them to `apex_events.json`
5. You can query the Python server for statistics and stored events

Customization

Change Python Server URL:

In `background.js`, modify:

javascript

```
const PYTHON_SERVER_URL = 'http://localhost:5000/events';
```

Add Custom Event Processing:

In `python_server.py`, modify the `process_event()` function to add your own logic:

python

```
def process_event(event):
    event_type = event.get('type')
    # Add your custom processing here
    if event_type == 'game_event':
        # Do something with game events
        pass
```

Change Tracked Features:

In `background.js`, modify the `setRequiredFeatures` array to add/remove tracked events.

Troubleshooting

Events not being received:

- Check if Python server is running
- Verify the server URL in background.js
- Check Overwolf console for errors (F12 in Overwolf)

App not loading:

- Ensure all required files are present
- Check manifest.json for syntax errors
- Make sure icon files exist

Game not detected:

- Verify game_id is correct (21566 for Apex Legends)

- Make sure Overwolf is running before launching the game

Development

To view logs:

- **Overwolf app logs**: Right-click the app in Overwolf and select "Inspect"
- **Python server logs**: Check the terminal where you ran the server

Available Apex Legends Events

Refer to the Overwolf Apex Legends API documentation for complete event details.

License

This is a sample project for educational purposes.