

Setup and Execution Guide for Street Fighter II Turbo Game Bot

May 2025

Abstract

This report outlines the steps to set up and run a machine learning (ML)-driven game bot for Street Fighter II Turbo, using the BizHawk emulator and a Python API. The bot employs an XGBoost model to control player actions in single-player (Player 1 vs. CPU) and two-player (Player 1 vs. Player 2) modes. Instructions cover software installation, Python environment setup, model training, and game execution, with debugging steps for the freezing issue after two moves. The solution is fully ML-driven, ensuring smart, non-idle gameplay.

1 Introduction

The Street Fighter II Turbo game bot project develops an AI to control players in the classic fighting game using a machine learning model. The bot, implemented in Python, predicts button inputs based on game state features, trained on augmented game data. This report details the setup and execution process to replicate the bot on a new machine, addressing software dependencies, model training, and game operation for both single-player and two-player modes. It also provides debugging steps for common issues, such as freezing after two moves.

2 Prerequisites

To run the game bot, ensure the following:

2.1 Hardware

- **OS:** Windows 10/11
- **CPU:** Modern processor (e.g., Intel i5)
- **RAM:** 8 GB minimum
- **Storage:** 500 MB free

2.2 Software

- **BizHawk Emulator:** Version 2.9 or later (<https://tasvideos.org/BizHawk>)
- **Python:** Version 3.8 or 3.9
- **ROM:** Street Fighter II Turbo (U).smc (must be sourced legally)
- **Command Prompt/PowerShell:** For running scripts

2.3 Python Libraries

- pandas (1.5.3)
- scikit-learn (1.3.2)
- xgboost (2.0.3)
- joblib (1.3.2)
- numpy (1.24.4)

3 Setup

Follow these steps to prepare the environment:

3.1 Install BizHawk

1. Download BizHawk 2.9+ from <https://tasvideos.org/BizHawk>.
2. Extract to C:\BizHawk\.
3. Verify EmuHawk.exe is present.

3.2 Setup Python Environment

1. Install Python 3.8/3.9 from <https://www.python.org>, ensuring “Add Python to PATH” is checked.

2. Create a virtual environment:

```
python -m venv gamebot_env
.\gamebot_env\Scripts\activate
```

3. Install libraries:

```
pip install pandas==1.5.3 scikit-learn==1.3.2 xgboost==2.0.3 joblib==
```

3.3 Organize Project Files

1. Create C:\GameBot\single-player\ and two-players\ directories.
2. Place Street Fighter II Turbo (U).smc, model.py, bot.py, controller.py, command.py, buttons.py, and game_data_augmented.csv in each directory. Update model.py to use the

4 Model Training

The bot uses an XGBoost model trained on game_data_augmented.csv.

4.1 Preprocess and Train

3. Navigate to C:\GameBot\single-player:


```
cd C:\GameBot\single-player
.\gamebot_env\Scripts\activate
```
2. Run:


```
python model.py
```
3. **Output:** Generates scaler.pkl and game_bot_model.pkl, with validation accuracies for buttons. The model preprocesses idle state to ensure active moves (left, right, A, B), preventing idle behavior.

5 Running the Bot

5.1 Single-Player Mode (Player 1 vs. CPU)

4. Open EmuHawk.exe in C:\BizHawk\.
2. File > Open ROM, select Street Fighter II Turbo (U).smc from single-player\.
3. Tools > Tool Box (Shift+T).
4. Choose Normal Mode, select Player 1 character.
5. In command prompt:


```
cd C:\GameBot\single-player
.\gamebot_env\Scripts\activate
python controller.py 1
```
6. In Tool Box, click Gyroscope Bot icon.
7. **Output:** Terminal shows "Connected to game!", bot controls Player 1.

5.2 Two-Player Mode (Player 1 vs. Player 2)

1. Open EmuHawk.exe, load ROM from two-players\.
2. Open Tool Box.
3. Choose VS Battle Mode, select characters for both players.
4. Open two command prompts, navigate to two-players\
`cd C:\GameBot\two-players`
`.\gamebot_env\Scripts\activate`
5. In first prompt:
`python controller.py 1`
6. In second prompt:
`python controller.py 2`
7. Click Gyroscope Bot icon.
8. **Output:** Both bots control players, fighting strategically.

6 Debugging Freezing Issue

If the game freezes after two moves:

1. **Check Logs:** Inspect `game_data_log.csv`:

```
import pandas as pd
log_df = pd.read_csv('game_data_log.csv')
print(log_df.tail(10))
print(log_df[['left', 'right', 'A', 'B']].sum())
```

Expect many frames, active buttons.

Console Output: Look for “timer stuck” or “position not updating” warnings.

Game Loop: Add debug prints to `controller.py` to verify loop iterations.

Engine Test: Send manual commands (`buttons.left = True`) to check movement.

Training Data: Ensure no idle states in `game_data_augmented.csv`.

7 Conclusion

The Street Fighter II Turbo game bot is fully ML-driven, using an XGBoost model to predict smart, non-idle actions. The setup involves installing BizHawk, Python, and libraries, training the model, and running the bot with the emulator. Debugging steps address freezing, likely due to game loop or engine issues. This guide ensures reproducible experiments for single-player and two-player modes.