3rd Strike Advanced Bot - README

Overview

`3rd_Strike_Advanced.py` is an AI bot for Street Fighter III: 3rd Strike on Fightcade. It features:

- Simple AI Mode: Hardcoded heuristic behavior (walk forward, throw hadokens, DP at close range)
- Ghost AI Mode: Learns from watching you play, Tekken-style
- Matchup-Aware Storage: Saves separate ghosts for each character matchup (e.g., Ken vs Chun-Li)
- Vision Support (Optional): Can read game state from screen (requires additional setup)
- CLI Interface: Command-line flags for different modes and configurations

Requirements

Essential Dependencies

```bash
pip install pynput
```

Optional (for vision-based game state reading)

```bash
pip install mss opencv-python numpy
```

System Requirements

- Linux (tested on MX Linux, Debian-based systems)
- xdotool - for window detection and input injection

```bash
 sudo apt install xdotool
```

- Fightcade with Street Fighter III: 3rd Strike loaded

Installation

1. Ensure xdotool is installed
2. Install Python dependencies:

```bash
pip install pynput
# Optional for vision:
pip install mss opencv-python numpy
```

3. Place `3rd_Strike_Advanced.py` in your working directory

Usage

Basic Command Structure

```bash
python3 3rd_Strike_Advanced.py --character <your_char> --opponent <opponent_char> --mode <mode> [option
```

Modes

1. Simple AI Mode (Default)

Runs basic hardcoded AI that walks forward and throws special moves.

```bash
python3 3rd_Strike_Advanced.py --character ken --mode simple
```

What it does:

- Distance > 250: Walks forward
- Distance 150-300: Throws hadoken
- Distance < 100: Dragon punch (DP)

2. Record Mode

Records your gameplay to teach the Ghost AI.

```bash
python3 3rd_Strike_Advanced.py --character ken --opponent chunli --mode record
```

How to use:

1. Start the script
2. Play matches on P2 controls (WASD + YUIHJK)
3. The bot watches and records your button presses with game context
4. Press Ctrl+C when finished
5. Choose to save the session when prompted
6. Optionally add a note to the filename

P2 Control Mapping:

- Movement: W (Up), S (Down), A (Left), D (Right)
- Punches: Y (Weak), U (Medium), I (Strong)
- Kicks: H (Weak), J (Medium), K (Heavy)
- Other: 6 (Coin/Select), 2 (Start)

3. Play Mode

Plays using the most recently saved Ghost AI patterns.

```
bash

python3 3rd_Strike_Advanced.py --character ken --opponent chunli --mode play
```

What it does:

- Automatically loads the latest ghost for the specified matchup
- Plays using learned patterns from your recordings
- Falls back to simple AI if no ghosts exist

4. Load Mode

Manually select which saved ghost to load and play.

```
bash

python3 3rd_Strike_Advanced.py --character ken --opponent chunli --mode load
```

What it does:

- Lists all saved ghosts for the matchup
- Prompts you to select one by number
- Loads selected ghost and starts playing

5. View Mode

View all saved ghosts without playing.

```bash
python3 3rd_Strike_Advanced.py --character ken --opponent chunli --mode view
```

Command-Line Options

Flag	Description	Default
`--character`	Your character (e.g., ken, ryu, chunli)	ken
`--opponent`	Opponent character	ryu
`--mode`	Mode: simple, record, play, load, view	simple
`--debug`	Enable debug logging	False
`--vision`	Enable vision-based extractor (experimental)	False
`--window-name`	Window name for xdotool to find	Fightcade

Examples

Record a Ken vs Chun-Li session with debug logging

```bash
python3 3rd_Strike_Advanced.py --character ken --opponent chunli --mode record --debug
```

Play with your saved Ken vs Ryu ghost

```bash
python3 3rd_Strike_Advanced.py --character ken --opponent ryu --mode play
```

Test simple AI against CPU

```bash
python3 3rd_Strike_Advanced.py --mode simple
```

How Ghost AI Works

Learning Process

1. Recording: Captures your keypresses with timing and duration
2. Context Association: Links each action to game situation (health, distance, super meter)
3. Pattern Storage: Saves situation → action mappings
4. Bucketing: Groups similar situations (e.g., 70% health ≈ 75% health)

Decision Making

1. Exact Match: If it's seen this exact situation before (≥2 times), do what you did
2. Fuzzy Match: If similar situation exists, use closest match
3. Randomness: 15% chance to pick a random learned action (adds variety)
4. Fallback: If no patterns match, do nothing (safer than random)

Storage Structure

Ghosts are saved in matchup-specific directories:

```
ghost_data/
  ken_vs_chunli/
    ken_vs_chunli_ghost_20250929_143022_1.pkl
    ken_vs_chunli_ghost_20250929_145533_2.pkl
  ken_vs_ryu/
    ken_vs_ryu_ghost_20250929_150122_1.pkl
```

Each file contains:

- Learned patterns (situation → actions)
- Situation frequency counts
- Character/opponent metadata
- Timestamp and record count

Vision Mode (Experimental)

The vision extractor attempts to read game state from screen pixels.

Enable Vision

```
bash

python3 3rd_Strike_Advanced.py --mode play --vision
```

Requirements

- `mss` - for screen capture
- `opencv-python` - for image processing
- Proper window geometry detection via xdotool

Limitations

- Highly experimental - requires tuning for your setup
- Health bar regions are hardcoded estimates
- Character position detection is very basic
- May not work with different screen resolutions or window sizes

Recommendation: Stick with dummy extractor for now. Vision support needs calibration.

Troubleshooting

"Could not find 3rd Strike window"

Solutions:

1. Ensure Fightcade is running with 3rd Strike loaded (not just in menu)
2. Check window name: `xdotool search --name "Fightcade"`
3. Try different window name: `--window-name "3rd Strike"`

"pynput not available"

Recording mode requires pynput to capture keyboard inputs.

```
bash

pip install pynput
```

Bot isn't pressing buttons

Check:

1. Window ID is correctly detected (shown in startup logs)
2. You're in a match (not in menu)
3. xdotool has permissions: `xdotool key --window <ID> w`

Ghost AI does nothing

Reasons:

1. No patterns loaded - record a session first
2. Current situation doesn't match learned patterns - play more varied matches
3. Check debug mode: `--debug` to see decision making

Inputs are delayed/wrong

Check:

1. Action duration settings (default: 0.016s cooldown)
2. System lag - close other programs
3. Fightcade input lag settings

Tips for Better Ghosts

1. Play Varied Matches: More diverse situations = smarter ghost
2. Record Multiple Sessions: Combine different playstyles
3. Practice Specific Scenarios:
   - Low health situations
   - Corner pressure
   - Anti-air responses
   - Different distance ranges
4. Iterative Training:
   - Record → Review → Record more → Improve
5. Matchup Specific: Record separate ghosts for each opponent character

Known Limitations

- No actual game state reading (unless vision enabled and calibrated)
- Dummy extractor provides static values - bot doesn't actually "see" health/distance
- Vision extractor needs per-setup calibration
- Cannot detect frame data or animation states
- No combo recognition or optimization
- P2 controls only (WASD layout)

File Structure

```
working_directory/
├── 3rd_Strike_Advanced.py        # Main bot script
├── ghost_data/                   # Auto-created on first save
│   ├── ken_vs_chunli/            # Matchup-specific directory
│   │   └── *.pkl                 # Saved ghost sessions
│   └── ken_vs_ryu/
│       └── *.pkl
└── README.txt                    # This file
```

Future Improvements

Potential enhancements:

- Actual computer vision for health bars
- Frame data integration
- Combo detection and optimization
- Multi-session merging
- Pattern confidence scoring
- Real-time performance metrics
- Web UI for ghost management

Credits

Built for Fightcade Street Fighter III: 3rd Strike on Linux. Uses xdotool for input injection and optional pynput for input recording.

License

Use at your own risk. This is for educational/entertainment purposes. Don't use in ranked matches without opponent consent.