NPC Colosseum Bot Programming Manual

DeepSeek

October 21, 2025

Create intelligent bots to battle in the arena!

Contents

1 Introduction

Welcome to the NPC Colosseum Bot Programming Guide! This document will help you understand how to create your own AI bots to battle in the arena. In NPC Colosseum, you can program bots that control characters during AI vs AI matches. Your bot will receive game state information and must return valid actions to compete against other bots.

2 Bot Function Structure

Your bot must be implemented as a Python function with the following signature:

```python
def get_player_bot_action(current_game_state, game_archive):
    # Your bot logic here
    return "Valid Action String"
```

Listing 1: Bot Function Signature

2.1 Parameters

- `current_game_state`: A dictionary containing the current state of the game
- `game_archive`: A list of all previous game states (for analyzing game history)

2.2 Return Value

Your function must return a string representing a valid action.

3 Game State Structure

The `current_game_state` dictionary contains:

```
{
    "player": {
        "hp": current_health_points,
        "max_hp": maximum_health_points,
        "skill_points": current_skill_points,
        "character_class": "Warrior"  # or "Mage" or "Rogue"
    },
    "opponent": {
        "hp": current_health_points,
        "max_hp": maximum_health_points,
        "skill_points": current_skill_points,
        "character_class": "Warrior"  # or "Mage" or "Rogue"
    },
    "inventory": {
        "Potion": count,
        "Ether": count,
        "Bomb": count,
        "Elixir": count
    }
}
```

Listing 2: Game State Structure

4 Actions Reference

4.1 Basic Actions

- `"Attack"` - Perform a basic attack
- `"Defend"` - Defend against the next attack

4.2 Item Actions

- `"Use Potion"` - Heal 30 HP
- `"Use Ether"` - Heal 15 HP
- `"Use Bomb"` - Deal 40 damage
- `"Use Elixir"` - Fully restore HP

4.3 Class-Specific Skills

4.3.1 Warrior Skills

- `"Cast Power Strike"` - Heavy weapon attack (30-40 damage, 3 SP)
- `"Cast Shield Bash"` - Stuns and damages (20-30 damage, 2 SP)
- `"Cast Battle Cry"` - Heals and boosts morale (15-25 heal, 3 SP)
- `"Cast Whirlwind"` - Hits all enemies (25-35 damage, 4 SP)

4.3.2 Mage Skills

- `"Cast Fireball"` - High damage fire spell (35-45 damage, 5 SP)
- `"Cast Ice Shard"` - Freezing damage (25-35 damage, 3 SP)
- `"Cast Arcane Heal"` - Powerful healing (30-40 heal, 4 SP)
- `"Cast Lightning Bolt"` - Massive lightning damage (40-50 damage, 6 SP)

4.3.3 Rogue Skills

- `"Cast Backstab"` - High damage from behind (35-45 damage, 4 SP)
- `"Cast Poison Dart"` - Damage over time (0-10 damage + poison, 3 SP)
- `"Cast Shadow Dodge"` - Evade next attack (100 defense for 1 turn, 3 SP)
- `"Cast Quick Strike"` - Fast, low cost attack (25-35 damage, 2 SP)

5 Character Classes

5.1 Warrior

- **Max HP**: 120
- **Basic Attack**: 18-25 damage
- **Skill Points**: 4
- **Defense**: 15
- **Starting Inventory**: 4 Potions, 1 Ether, 1 Bomb

5.2 Mage

- **Max HP**: 80
- **Basic Attack**: 10-16 damage
- **Skill Points**: 8
- **Defense**: 25
- **Starting Inventory**: 2 Potions, 3 Ether, 2 Bombs, 1 Elixir

5.3 Rogue

- **Max HP**: 100
- **Basic Attack**: 15-22 damage
- **Skill Points**: 6
- **Defense**: 40
- **Starting Inventory**: 3 Potions, 2 Ether, 3 Bombs

6 Inventory Items

- **Potion**: Heals 30 HP
- **Ether**: Heals 15 HP
- **Bomb**: Deals 40 damage
- **Elixir**: Fully restores HP (rare)

7 Strategy Guide

7.1 General Tips

1. **Monitor Health**: Use healing items when HP drops below 40
2. **Manage Skill Points**: Basic attacks restore SP for some classes
3. **Class Synergy**: Choose actions that complement your class strengths
4. **Opponent Awareness**: Consider the opponent's class and health when choosing actions
5. **Resource Management**: Don't waste powerful items or skills unnecessarily

7.2 Class-Specific Strategies

- **Warrior**: Use high-damage skills and defend to build skill points
- **Mage**: Manage skill points carefully and use powerful spells strategically
- **Rogue**: Use evasion and poison effects to control the battle

8 Example Bot

Here's the simple bot provided in the game:

```python
import random

def get_player_bot_action(current_game_state, game_archive):
    player_hp = current_game_state['player']['hp']
    inventory = current_game_state['inventory']

    # If HP is low and has a Potion, heal
    if player_hp < 40 and inventory.get("Potion", 0) > 0:
        return "Use Potion"

    # If opponent is low HP and we have a bomb, finish them!
    if current_game_state['opponent']['hp'] < 45 and inventory.get("Bomb", 0) > 0:
        return "Use Bomb"

    # Otherwise, cast a powerful spell or just attack
    if random.random() < 0.5: # 50% chance to cast Fireball
        return "Cast Fireball"
    else:
        return "Attack"
```

Listing 3: Example Bot Implementation

9 Testing Your Bot

1. Replace the `get_player_bot_action` function in `bots.py` with your implementation
2. Run the game and select "AI vs AI" mode
3. Watch how your bot performs against the default opponent bot
4. Iterate and improve your strategy!

10 Troubleshooting

- **Invalid Action**: Make sure your bot returns exactly the action strings shown in this manual
- **Skill Points**: Check that you have enough skill points before using skills
- **Inventory**: Verify item counts before trying to use items
- **Class Skills**: Remember that skills are class-specific

Happy coding and may the best bot win!