```
def clean_model_code(model_code_str: str) -> str:
    """
    Cleans up the generated model code string.

    Args:
        model_code_str (str): The raw model code as a string.

    Returns:
        str: The cleaned-up model code.
    """
    cleaned_code = (
        model_code_str.replace("\\n", "\n")
        .replace("\\"", """)
        .replace('\\'', "'")
    )
    return cleaned_code.strip()


code = """

# Quantum Dimensions: A game of shifting realities\\n\\nimport random\\n\\nclass QuantumDimensionsGame:\\n    def __init__(self):\\n        self.player_position = (0, 0)\\n        self.realities = []\\n        self.current_reality = 0\\n        self.generate_realities()\\n\\n    def generate_realities(self):\\n        # Create a multi-dimensional reality space\\n        for _ in range(3):  # three parallel realities\\n            reality = [[random.choice([\'empty\', \'enemy\', \'treasure\']) for _ in
```

```
range(5)] for _ in range(5)]\\n            self.realities.append(reality)\\n\\n    def display_reality(self):\\n
    print(f\'Reality #{self.current_reality + 1}:\')\\n        for row in self.realities[self.current_reality]:\\n
        print(\' \'.join(row))\\n\\n    def shift_reality(self):\\n        print(\\"Shifting dimensions...\\")\\n
self.current_reality = (self.current_reality + 1) % len(self.realities)\\n\\n        def move_player(self,
direction):\\n            x, y = self.player_position\\n            if direction == \'up\' and x > 0:\\n
self.player_position = (x - 1, y)\\n                elif direction == \'down\' and x < 4:\\n
self.player_position = (x + 1, y)\\n        elif direction == \'left\' and y > 0:\\n            self.player_position
= (x, y - 1)\\n        elif direction == \'right\' and y < 4:\\n            self.player_position = (x, y + 1)\\n
else:\\n                print(\\"Can\'t move in that direction.\\")\\n\\n        def play_turn(self):\\n
self.display_reality()\\n            move = input(\\"Enter move (up/down/left/right) or shift to change
realities: \\").strip().lower()\\n        if move == \'shift\':\\n            self.shift_reality()\\n        else:\\n
self.move_player(move)\\n                x, y = self.player_position\\n                current_state =
self.realities[self.current_reality][x][y]\\n                if current_state == \'enemy\':\\n
print(\\"You\'ve encountered an enemy!\\")\\n            elif current_state == \'treasure\':\\n
print(\\"You\'ve found a treasure!\\")\\n        print(f\'Player position: {self.player_position}\')\\n\\n    def
start_game(self):\\n        print(\\"Welcome to Quantum Dimensions!\\")\\n        while True:\\n
self.play_turn()\\n\\nif __name__ == \'__main__\':\\n    game = QuantumDimensionsGame()\\n
game.start_game()
"""


cleaned = clean_model_code(code)
# print(cleaned)
exec(cleaned)
```