

## Text Adventure Game

Table of Contents
1. Project Overview
2. Features
3. Installation
4. How to Play
5. Game Mechanics
6. Technical Architecture
7. File Structure
8. Customization Guide
9. Troubleshooting
10. Future Roadmap

### 1. Project Overview

A Python-based interactive fiction game featuring:

- Immersive text-based gameplay
- Persistent game state saving
- Multiple endings based on player choices

### 2. Features

Core Gameplay

- Room-based navigation system
- Inventory management (keys, weapons, artifacts)
- Health and combat system
- Gold economy and treasure collection

Technical Features

- Object-oriented design
- Save/load functionality using pickle
- Input validation system

### 3. Installation

Requirements

- Python 3.6+

Setup Instructions

```
bash
```

```
Copy
```

```
# Clone the repository
```

```
git clone https://github.com/your-username/text-adventure-game.git
```

```
# Navigate to project directory  
cd text-adventure-game
```

```
# Run the game  
python adventure_game.py
```

#### 4. How to Play

##### Basic Controls

- Type numbers (1, 2, 3) or text commands
- Press Enter or click Submit
- Use Save/Load buttons to manage progress

##### Game Progression

1. Start in the initial room
2. Explore different areas (corridor, kitchen, pantry)
3. Collect items to solve puzzles
4. Reach the treasure room for final choices

#### 5. Game Mechanics

##### Room System

Room	Key Features	Required Items
Start	Contains initial chest	None
Corridor	Locked door	Key
Kitchen	Dangerous pantry	Knife
Treasure	Final challenge	Medallion

##### Combat System

- Random damage calculation (30-50 HP)
- Knife provides combat advantage
- Health restoration options

## 6. Technical Architecture

Class Diagram

Key Algorithms

### 1. Input Processing:

- Handles both numeric and text input
- Supports partial command matching

### 2. Game State Management:

- Tracks player progress
- Manages inventory and health

## 7. File Structure

Copy

text-adventure-game/

```
├── adventure_game.py    # Main game file
├── README.md            # Project documentation
```

## 8. Customization Guide

Adding New Rooms

1. Create new room method (e.g., `dungeon_room()`)
2. Add to `room_descriptions` dictionary
3. Connect to existing rooms via choices

Modifying Items

python

Copy

# Example: Add healing potion

```
self.inventory.append("potion")
```

```
self.heal(30) # Restore 30 HP when used
```

## 9. Troubleshooting

Issue	Solution
Tkinter errors	Install tk package: <code>sudo apt-get install python3-tk</code>
Save file corruption	Delete <code>adventure_save.pkl</code>
Input not working	Use exact command or number

## 10. Future Roadmap

- Add more rooms and puzzles
- Implement NPC dialogue system
- Add graphical assets
- Develop web version