```
import random

# Game variables
player_progress = 0
shards_collected = 0
alliances_formed = 0
inventory = {'Sword': 1, 'Healing Potion': 3, 'Map': 1}
quest_log = ['Find the Shard in the Enchanted Forest', 'Help the Village Elder']

# Game functions
def enchanted_forest():
    global player_progress
    print("You enter the enchanted forest...")
    # Player choices and consequences
    choice = input("1. Follow the glowing lights. 2. Take the hidden path. Choose your path (1/2): ")
    if choice == '1':
        print("You follow the lights and discover a Shard of Destiny!")
        shards_collected += 1
    elif choice == '2':
        print("You take the hidden path and encounter a magical creature.")
        # Implement combat or diplomacy logic
    player_progress += 20

def encounter_npc():
    print("You meet a mysterious NPC...")
```

```
    print("1. Engage in conversation.")
    print("2. Ignore and continue your journey.")
    choice = input("Enter your choice (1/2): ")
    if choice == '1':
        print("The NPC shares valuable information.")
        alliances_formed += 1

def solve_puzzle():
    print("You come across a perplexing puzzle...")
    print("1. Attempt to solve the puzzle.")
    print("2. Look for an alternative route.")
    choice = input("Enter your choice (1/2): ")
    if choice == '1':
        print("You solve the puzzle and find a hidden passage.")
        player_progress += 15

def battle_enemy():
    print("A fearsome creature blocks your path...")
    print("1. Fight the creature.")
    print("2. Try to sneak around.")
    choice = input("Enter your choice (1/2): ")
    if choice == '1':
        print("You engage in battle and defeat the creature.")
        player_progress += 10
    elif choice == '2':
```

```
        print("You sneak around the creature successfully.")
        player_progress += 5

def main_quest():
    global player_progress
    # Track progress of the main quest
    print("Current main quest: ", quest_log[0])
    enchanted_forest()  # Example location

def side_quest():
    global player_progress
    # Implement side quests with unique challenges and rewards
    print("Current side quest: ", quest_log[1])
    solve_puzzle()  # Example puzzle-solving

def explore_location():
    global player_progress
    # Handle exploration of different locations with varying events
    print("Exploring a new location...")
    # Update player's stats and choices

def true_ending():
    print("Congratulations! You have successfully reunited all seven shards, bringing balance to Eldoria.")

def chaotic_ending():
```

```
    print("Alas! Your choices have unleashed chaos upon Eldoria. The shards' power has corrupted the realm.")


# Game loop
while player_progress < 100:
    main_quest()
    side_quest()
    explore_location()


# Determine endings based on player progress and decisions
if shards_collected == 7 and alliances_formed >= 3:
    true_ending()
else:
    chaotic_ending()
```