



**NS** Claude please create a game for me using python and pygame. In the game the player should navigate a 2d space using the arrow keys. In this game there should be a maze like region with rocks and stalagmites. Inside the region should be chests which contain gold. The player should be able to navigate the maze and collect gold from the chests.

I'd be happy to create a 2D maze game with pygame where the player collects gold from chests! This will be a single Python file with all the necessary components.

Let me create this game for you:

maze\_treasure\_game.py

Code

```
# /// script
# requires-
python =
">=3.12"
#
```

I've created a pygame-based maze treasure hunting game for you! This single Python file includes everything you need to run a complete 2D maze game where you can navigate with arrow keys to collect gold from chests.

Here's what the game includes:

- A procedurally generated maze using a randomized depth-first search algorithm
- Player character that can be controlled with arrow keys
- Wall obstacles that prevent movement
- Treasure chests containing random amounts of gold
- A scoring system that tracks how much gold you've collected
- Win condition when all chests are opened

To run the game:

1. Save the code as `maze_treasure_game.py`

Reply to Claude...



Claude 3.7 Sonnet ▾



```
player, walls, chests, monster = create_level()
frame_count = 0
game_over = False
win = False

elif event.key == pygame.K_SPACE and not game_over and not win:
    # Try to break a wall
    broken_wall_pos = player.break_wall(walls)
    if broken_wall_pos:
        break_effect = broken_wall_pos
        break_effect_duration = 15 # Show effect for 15 frames

# Get key states
keys = pygame.key.get_pressed()

# Process game logic only if the game is not over
if not game_over and not win:
    # Update player (cooldowns, etc)
    player.update()

# Handle player movement
dx, dy = 0, 0
if keys[pygame.K_LEFT]:
    dx = -player.speed
if keys[pygame.K_RIGHT]:
    dx = player.speed
if keys[pygame.K_UP]:
    dy = -player.speed
if keys[pygame.K_DOWN]:
    dy = player.speed

player.move(dx, dy, walls)

# Move the monster
monster.move(player, walls, frame_count)

# Check if monster caught the player
if monster.check_collision(player):
    game_over = True
```