Python Code: Autonomous Vehicle Path Planning with Obstacles

```
import matplotlib.pyplot as plt
import numpy as np

# Starting point
start = np.array([0, 0])

# Target/goal point
goal = np.array([10, 10])

# Obstacles: (x, y, radius)
obstacles = [
(4, 4, 1),
(6, 6, 1.5),
(8, 2, 1)
]

# Function to detect if point collides with any obstacle
def is_collision(point, obs, threshold=0.5):
for x, y, r in obs:
distance = np.linalg.norm(point - np.array([x, y]))
if distance <= r + threshold:
return True
return False

# Generate a simple path avoiding obstacles
path = [start]
current = start.copy()

while np.linalg.norm(current - goal) > 0.5:
# Move toward the goal
direction = goal - current
direction = direction / np.linalg.norm(direction) * 0.5
next_point = current + direction

# If collision, take a simple detour
if is_collision(next_point, obstacles):
direction = np.array([-direction[1], direction[0]])  # 90 degree turn
next_point = current + direction

path.append(next_point)
current = next_point

# Convert path to array for plotting
path = np.array(path)

# Plotting
fig, ax = plt.subplots()
ax.plot(path[:,0], path[:,1], 'b-o', label='Path')     # Vehicle path
ax.plot(start[0], start[1], 'go', label='Start')       # Green start point
ax.plot(goal[0], goal[1], 'ro', label='Goal')       # Red goal point

# Draw obstacles as circles
for x, y, r in obstacles:
obstacle_circle = plt.Circle((x, y), r, color='gray', alpha=0.5)
ax.add_patch(obstacle_circle)

# Styling the plot
ax.set_aspect('equal')
ax.set_title("Autonomous Vehicle Path Planning")
ax.legend()
plt.grid(True)
plt.show()
```