



Aluno: Davi Andrzejewski Junkes

Matéria: Inteligência artificial

Relatório da Simulação de Tráfego

Introdução

Este projeto consiste em uma simulação de tráfego desenvolvida em Python utilizando a biblioteca Pygame. O objetivo da simulação é modelar o comportamento de veículos em um cruzamento controlado por semáforos, permitindo a visualização do movimento dos veículos e a interação dinâmica entre eles e os sinais de trânsito.

Estrutura do Código

Principais Componentes

1. Classes:

- **TrafficSignal**: Representa um semáforo. Possui atributos para os tempos de luz vermelha, amarela e verde. A classe permite a atualização dos estados dos sinais conforme os veículos se aproximam.
- **Vehicle**: Representa um veículo na simulação. Inclui atributos como lane (faixa), vehicleClass (tipo de veículo), speed (velocidade) e suas coordenadas (x, y). A classe gerencia o movimento do veículo com base no estado do semáforo e na presença de outros veículos.

2. Inicialização:

- A função `initialize` cria e configura os semáforos com seus tempos padrão. Esta função é chamada ao início da simulação.

```
ts1 = TrafficSignal(0, defaultYellow, defaultGreen[0])
```

3. Geração de Veículos:

- A função `generateVehicles` é responsável por gerar veículos aleatórios em intervalos de tempo fixos. Essa função é executada em uma thread separada para garantir que a simulação continue fluida enquanto novos veículos são adicionados.

```
vehicle_type = random.randint(0, 3)
```

4. Loop Principal:

- O loop principal do programa gerencia a renderização dos gráficos e a atualização dos estados dos veículos e semáforos. Ele inclui:
 - A verificação de eventos do Pygame.
 - A renderização do fundo e dos sinais de trânsito.
 - A atualização das posições dos veículos e a contagem dos veículos parados.
 - A atualização e exibição do temporizador dos semáforos.

Lógica de Movimento dos Veículos

A movimentação dos veículos é baseada em regras definidas na classe `Vehicle`. As principais condições para o movimento incluem:

Condição de Parada: Cada veículo deve parar ao alcançar sua linha de parada definida. O método `isVehicleStopped` determina se um veículo está completamente parado, verificando se ele está em sua posição de parada e se não cruzou a linha de parada.

python

Copiar código

```
if vehicle.direction == 'right':  
    return vehicle.x + vehicle.image.get_rect().width <=  
stopLines[vehicle.direction] and vehicle.crossed == 0
```

•

- **Condição de Movimento:** Um veículo pode se mover se:

- A luz do semáforo estiver verde.
- Ele não tiver atingido sua posição de parada.

- Houver espaço suficiente na faixa para que ele avance, considerando o espaço ocupado por outros veículos na mesma faixa.

```
if((self.x+self.image.get_rect().width<=self.stop or self.crossed ==  
1 or (currentGreen==0 and currentYellow==0)) and (self.index==0 or  
self.x+self.image.get_rect().width<(vehicles[self.direction][self.la  
ne][self.index-1].x - movingGap))):  
    self.x += self.speed # move o veículo
```

•

Contagem de Veículos Parados

A contagem de veículos parados ao lado de cada semáforo é gerenciada em um dicionário (`stoppedVehiclesCount`), onde as chaves correspondem às direções (direita, baixo, esquerda, cima). A contagem é atualizada no loop principal, apenas quando um veículo é considerado parado.

python

Copiar código

```
stoppedVehiclesCount[vehicle.direction] += 1
```

Temporização dos Sinais

Os semáforos alternam entre os estados de verde, amarelo e vermelho. A lógica de temporização é implementada na função `repeat`, que gerencia a duração de cada estado do semáforo. Ao expirar o tempo do sinal verde, o semáforo muda para o estado amarelo, e então para o vermelho, permitindo a mudança controlada dos sinais conforme as regras do tráfego.

```
while(signals[currentGreen].green>0):  
    updateValues()  
    time.sleep(1)
```

Conclusão

A simulação de tráfego oferece uma representação visual e interativa do controle de tráfego em um cruzamento. Através da implementação de classes para semáforos e veículos, a lógica de movimento, temporização e contagem de veículos, este projeto permite a observação de como diferentes fatores influenciam o fluxo de tráfego. O uso de threads para geração de veículos e a renderização contínua garantem uma simulação fluida e realista.