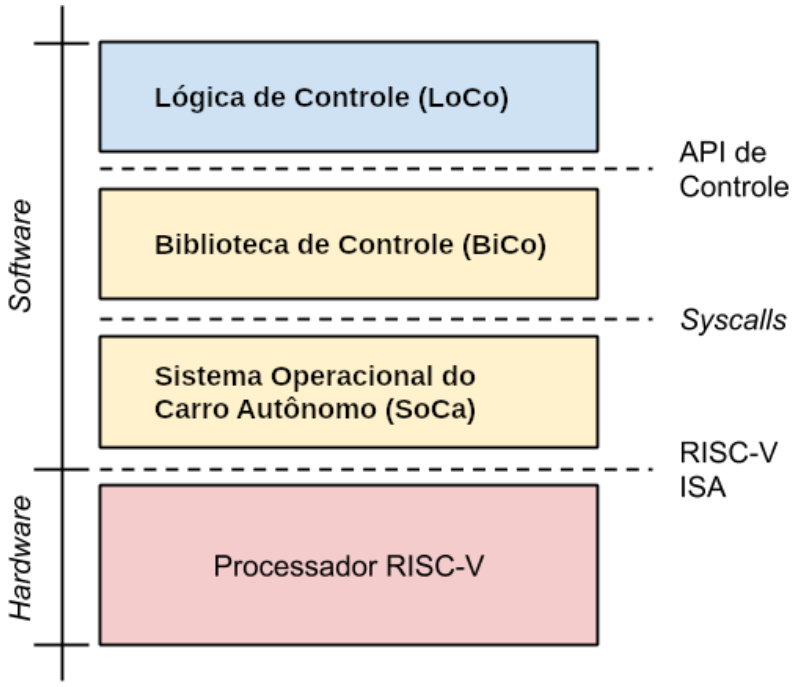
Trabalho 2 - Parte 2

Descrição

Nesta segunda parte deste trabalho, você deverá guiar o *Self-Driving Car* a locomover-se dentro de uma cidade, evitando colisões durante seu percurso. Para tanto, você utilizará a pilha de software descrita na primeira parte do segundo trabalho da disciplina, ilustrada na Figura abaixo e detalhada nas demais seções.

O código auxiliar para este trabalho está disponível neste link.



Subcamada LoCo

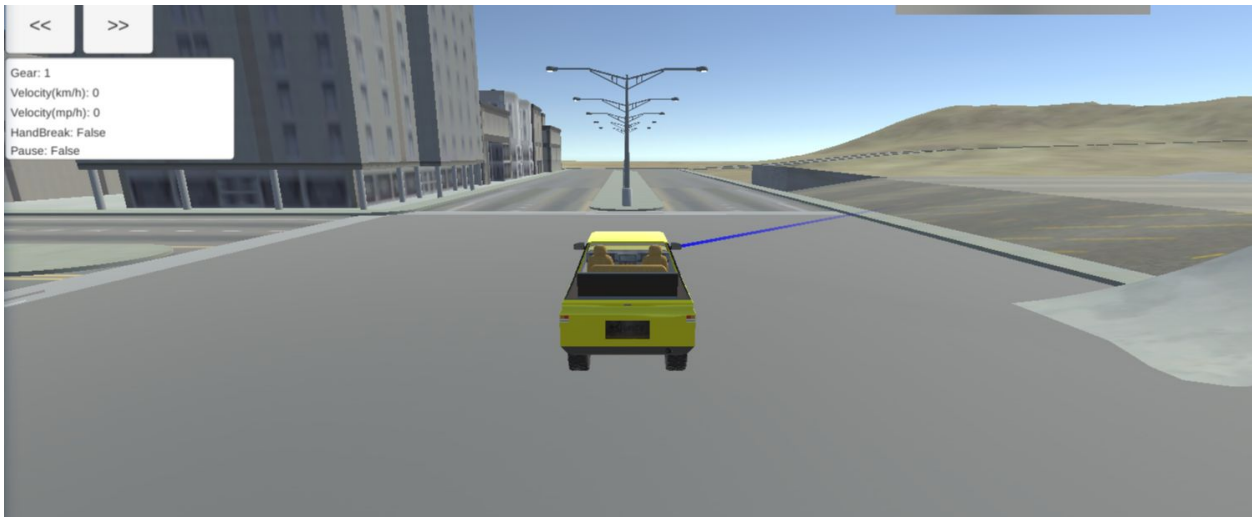
O código da subcamada LoCo deve ser implementado em linguagem C e deve fazer uso das rotinas disponíveis na API de Controle (arquivo `api_car.h`) para enviar comandos para o carro. Você desenvolverá um programa na linguagem C para a camada LoCo no arquivo `loco.c`.

Lógica de controle do programa

Nesta segunda parte do trabalho o carro deverá se mover autonomamente dentro da cidade, sem esbarrar/colidir com nenhum objeto durante seu trajeto. Durante seu passeio, o carro deverá passar próximo a três pontos (i.e, em um raio de 20 metros deles), indicados por coordenadas, descritas na Tabela abaixo.

	x	**y**	**z**
Cruzamento 1	-17	1	12
Cruzamento 2	141	1	8
Cruzamento 3	290	1	203

As proximidades dos cruzamentos 1, 2 e 3 são ilustrados nas Figuras abaixo, em sequência, respectivamente. Seu trajeto deve ser concluído em menos de 10 minutos e os pontos podem ser visitados em qualquer ordem.



Note que, as *syscalls* `Syscall_get_position` e `Syscall_get_rotation` podem ser utilizadas para obtenção da posição e rotação (variando de 0 a 360) atual do carro, respectivamente, e a *syscall* `Syscall_get_sensor_distance` pode ser utilizada para detectar objetos. Por fim, sua solução deverá considerar que o carro poderá iniciar em qualquer ponto válido dentro da cidade (excluindo o estacionamento e as rodovias).

Subcamada BiCo

O código da subcamada BiCo deve implementar as rotinas da API de Controle em linguagem de montagem do RISC-V. A API está descrita no arquivo `api_car.h` e deve ser implementada no arquivo `bico.s`. Para controlar o *hardware*, o código deve realizar chamadas ao sistema (instrução `ecall`), ou *syscalls*. As *syscalls* estão definidas na Seção ‘Subcamada SoCa’ no documento da primeira parte do trabalho.

Subcamada SoCa

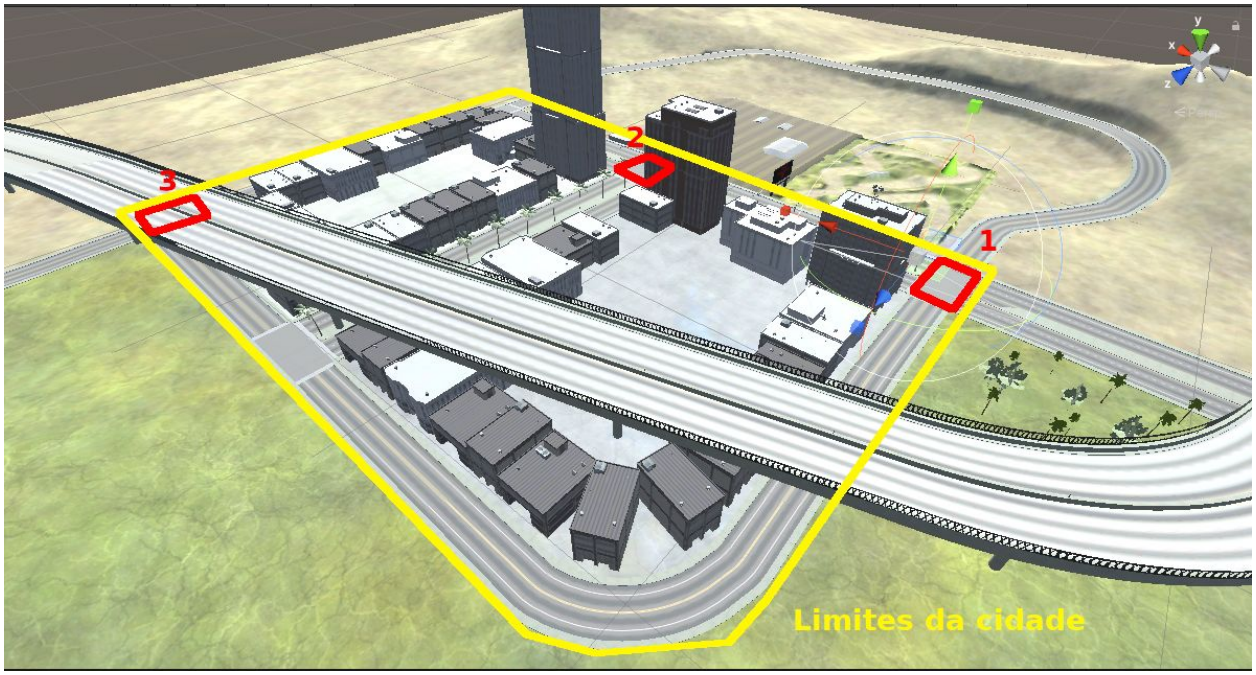
Você deve programar a subcamada SoCa, no arquivo `soca.s`, para atender as chamadas de sistemas definidas na Seção 'Subcamada SoCa' no documento da primeira parte do trabalho. A inicialização do sistema e descrição dos demais periféricos, mapeamento de memória e da infraestrutura também estão disponíveis nas demais Seções do documento da primeira parte do trabalho**.**

Entrega

Você deve submeter três arquivos: `loco.c, bico.s e soca.s` cada um deles adaptados com sua solução, conforme descrito nas seções acima. Assim sendo, **você não deve submeter outros arquivos**. Além disso, no código auxiliar disponibilizado há um `Makefile` que deve ser utilizado para a compilação do seu programa e que será utilizado para avaliação. Você não deve alterar o conteúdo do `Makefile` ou dos outros arquivos que não devem ser submetidos (e.g., `api_car.h`). Por fim, os arquivos `bico.s` e `soca.s` podem ser os mesmos dos submetidos para a primeira parte do trabalho ou diferentes.

Dicas e Observações

- Habilite a opção "*Enable debug controls*" para conhecer a pista. Para tanto, utilize as setas direcionais ou as teclas W, A, S e D. Note que esta opção deve ficar desabilitada durante o teste/execução de seu código.
- Os pontos dos cruzamentos são fixos. O carro poderá iniciar em qualquer local **válido e dentro da cidade.**
- Tente utilizar uma aba anônima caso tenha problemas para carregar/utilizar o *self-driving car.*
- Haverá penalidade caso o carro saia das ruas da cidade.
- O carro **não pode bater ou esbarrar em nenhum objeto**.
- A espera ocupada pode interferir no desempenho do simulador por realizar muitas requisições no barramento. Desta forma, no simulador, na aba *Hardware*, você pode alterar a frequência do barramento. Sugerimos a frequência de aproximadamente 10Hz para realizar testes e depuração. Para avaliação, utilizaremos o valor de 50Hz.
- Para o teste, o carro inicia na coordenada: `(-16, 1, 80)` , com rotação: `(0, 180, 0)` para os eixos x, y e z, respectivamente.
- A cidade tem um formato quase retangular. Uma vista da área da cidade e dos cruzamentos 1, 2 e 3 é mostrada na Figura abaixo.


1
2
3
Limites da cidade