

Cope with Cops 2: um Jogo Multijogador em Python

Victor Grovermann, João Rezende

Curso de Bacharelado em Ciência da Computação
Universidade Federal de Pelotas (UFPel)

vdtgrovermann@inf.ufpel.edu.br, jplrezende@inf.ufpel.edu.com

Resumo. *Este artigo descreve a implementação de um jogo multijogador de perseguição policial em Python utilizando comunicação por TCP.*

1. Introdução

O objetivo do trabalho é implementar um jogo multijogador em tempo real que permite que os usuários joguem pela Internet ou em rede local. Cada jogador controlará um veículo em um espaço bidimensional, utilizando botões direcionais para controlar suas manobras e aceleração.

O jogo ocorre em uma estrada larga e intensamente trafegada por carros. Caso não queira colidir e ser desqualificado, cada jogador precisa ser capaz de reagir rapidamente a carros que aparecerem adiante. A densidade do tráfego é pequena no início de cada partida, mas cresce gradativamente; e não demora muito para que se torne grande suficiente para forçar os jogadores a fazer uso frequente dos seus freios e, em casos extremos, ter que planejar com certa antecedência os caminhos que vai tomar no meio dos carros à frente.

Um dos jogadores – o primeiro a estabelecer conexão com o servidor no período que precede a partida – exerce o papel de motorista da viatura policial perseguidora – facilmente identificável no meio do trânsito por conta de sua única cor branca e chamativas luzes azuis e vermelhas piscantes. O próximo jogador a se conectar exerce o papel do fugitivo, que é identificado por sua cor vermelha.

A partida termina quando ocorre um dos seguintes eventos:

- a viatura policial alcança o carro fugitivo. Nesse caso, o policial vence.
- o carro fugitivo atinge uma distância grande da viatura policial. Nesse caso, o fugitivo vence.
- um dos jogadores sofre colisão com um carro de trânsito. Nesse caso, o outro jogador vence.

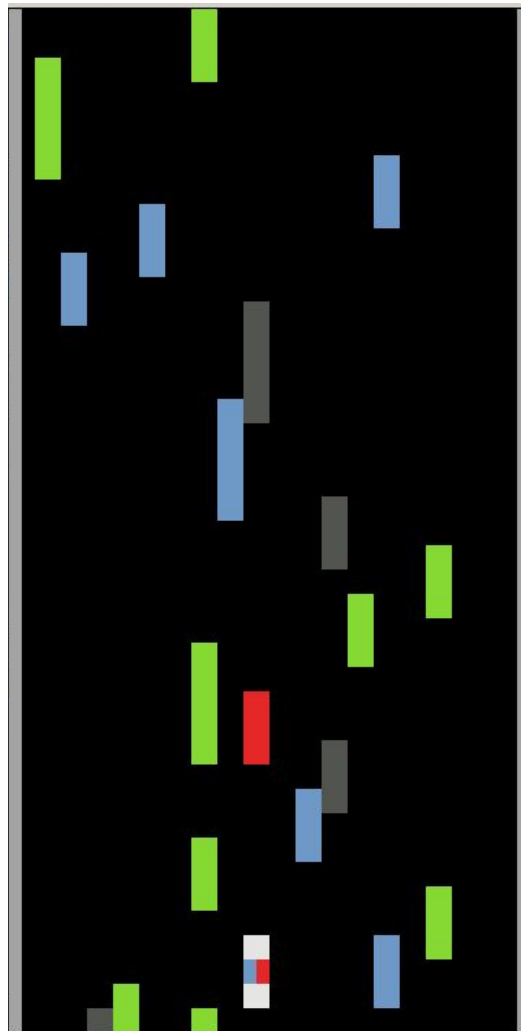


Figura 1. O jogo.

2. Comunicação

Uma conexão TCP entre o servidor e cada um dos clientes é usada para informá-los do estado da partida, das posições dos carros de trânsito e do carro do outro jogador.

Cada cliente coleta informações recebidas do servidor no início de cada *tick*, e as usa nos cálculos que faz para gerar o próximo quadro. Em cada mensagem, os parâmetros são separados por espaço, e mensagens diferentes são delimitadas por um caractere de nova linha ('\n').

2.1. Comunicação cliente-servidor

Tabela 1. Mensagens do cliente e seus formatos e significados:

| TCP | |
|----------|-------------|
| Mensagem | Significado |

| | |
|---------------------------|---|
| INIT | Enviado pelo policial para solicitar o início da partida. |
| BOOM | O carro do cliente sofreu colisão catastrófica com um veículo do trânsito. Encerra a conexão. |
| CAPTURE | A viatura policial alcançou o carro fugitivo. Termina a conexão. |
| MV <latitude> <longitude> | O carro do cliente se deslocou para a posição (latitude, longitude). |

2.2. Comunicação servidor-cliente

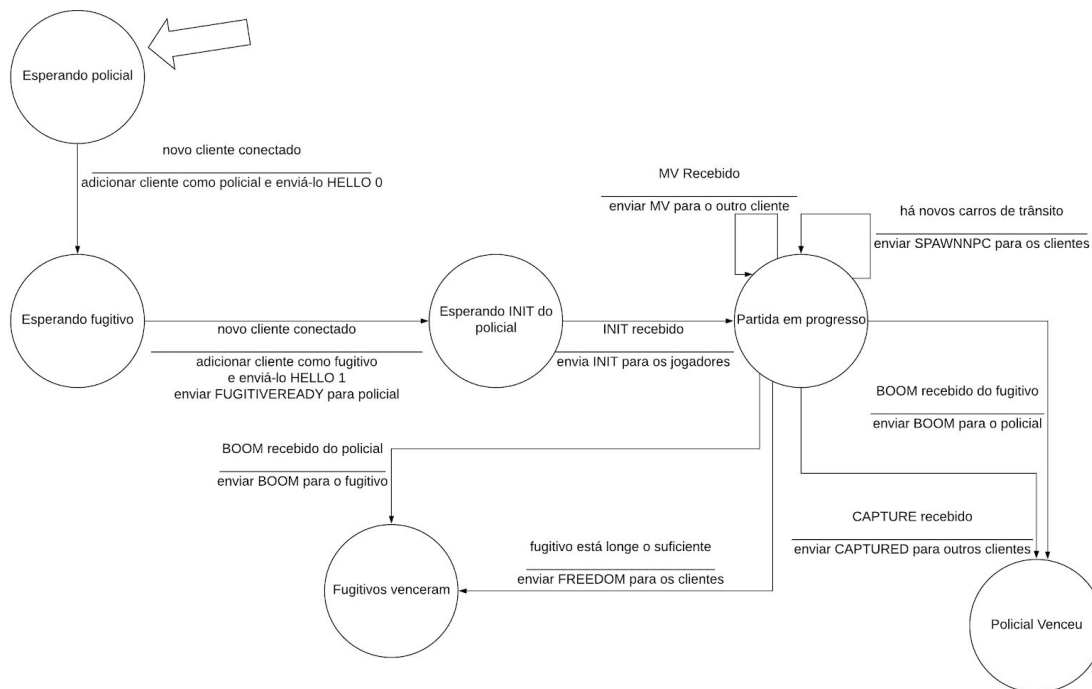
Tabela 2. Mensagens do servidor e seus formatos e significados.

| TCP | |
|--|---|
| Mensagem | Significado |
| HELLO <id> | Enviada para cada cliente recém conectado, informando seu ID (inteiro). (O policial possui ID 0; o fugitivo possui ID 1.) |
| FUGITIVEREADY | Notifica o policial de que o fugitivo se conectou e está aguardando o início da partida. |
| INIT | Notifica os jogadores do começo do jogo. |
| SPAWNNPC <latitude> <longitude> <color> | Enviada para informar a posição de um carro de trânsito. |
| DED | Enviada para um cliente quando o outro manda BOOM. Termina a conexão. |

| | |
|---------------------------|--|
| CAPTURED | Enviada para o fugitivo quando é alcançado pela viatura policial. Termina a conexão. |
| FREEDOM | Enviada para os clientes quando o fugitivo está longe o suficiente do policial. |
| MV <latitude> <longitude> | Enviado para cada cliente em cada tick para informá-lo da posição do outro cliente. |

3. Máquinas de estado

3.1. Máquina de Estados do Servidor



3.2. Máquina de Estados do Cliente

