

Devlog 1

20-9-25

Devlog - Projeto Neuron Racers

Introdução

O desenvolvimento do projeto começou no início deste mês. Por questões de organização e prioridades pessoais, só comecei a registrar anotações hoje, dia 20 de setembro.

Abaixo está um resumo dos aproximadamente 20 dias de desenvolvimento:

Interinas:

O desenvolvimento começou no início deste mês, mas por questões de organização e prioridades pessoais, eu comecei a fazer anotações hoje, dia 20 de setembro.

Segue aqui um resumo destes ~20 dias de desenvolvimento:

Primeiro fim de semana:

Criei o repositório, escolhi a versão do Godot e decidi que iria codificar a IA em gdscript, dentro do próprio jogo.

Com o auxílio do ChatGPT, decidi seguir com a ideia de fazer um MLP¹ básico que usa uma array de RayCasts² como entrada e tem dois valores de output, um para esterçamento e outro para aceleração e frenagem.

Passei boa parte do primeiro dia dedicado a fazer uma física decente para os carros, o que me custou um bom tempo de desenvolvimento, mas me poupou muita dor de cabeça até agora.

Foi neste dia também que eu decidi o estilo de arte/jogabilidade que eu seguiria no projeto, me inspirando em antigos jogos de flash que rodavam no navegador, mais especificamente jogos top-down e também no título/franquia da NADEO, Trackmania (mais especificamente Trackmania Nations Forever, TMNF). Estou usando um asset pack^[1] de corrida que encontrei de graça e de uso livre na itch.io feito pelo usuário *looneybits*

Ao fim do primeiro dia, eu tinha uma pista com colisão e um carro que eu podia dirigir manualmente.

No segundo dia eu me dediquei a fazer uma IA ao menos dirigir o carro. Isso se provou um desafio porque eu não tinha separado a lógica de controle do carro da input de um player humano (difícil de explicar porque tinha tomado uma decisão dessas). Mas por fim, quando refatorei parte da lógica do carro, eu tinha um módulo dedicado ao controle por IA e outro módulo dedicado ao controle humano (que eu removi em algum ponto porque estava quebrado e eu desisti um pouco da ideia de deixar o player competir contra a IA).

De início, eu decidi criar a MLP em uma configuração 5, 8, 2 (5 raycasts como entrada, 8 neurônios como camada escondida e 2 saídas).

O fitness inicialmente era calculado com base na velocidade total que um carro acumulava e na distância total viajada desde a origem do circuito.

Inclusive, de início eu não havia preparado uma mecânica para desabilitar/matar os carros que colidiam com as paredes, então com o tempo eu comecei a ver muitos carros fazendo o que poderia ser considerado um "*wallride*", ou seja, raspando na parede enquanto fazia a volta.

Mas uma coisa não estava certa. E isto seria um problema que eu demoraria alguns dias pra perceber, mas eu estava sobrescrevendo os cérebros que eu treinava e mutava com outros novos no script do carro, então eles de fato não estavam aprendendo.

Após um hiato de alguns dias, voltei a trabalhar no jogo. Foi aqui que identifiquei o bug que bloqueava novas gerações de aprenderem com a anterior. Eu também fiz um sistema rudimentar de checkpoints ao longo da pista, para dar à IA um senso melhor de progressão (ou assim eu pensava que seria, com os 36 checkpoints que eu coloquei, um número absurdo que mais atrapalhou do que ajudou).

Eu também dediquei um bom tempo a fazer um sistema de input que até agora (dia 20 de set) não usei direito porque quebrei ele de alguma forma.

Desde então, tem sido uma questão de aprender como a IA funciona e refinar o sistema de recompensa.

Foi por aqui inclusive que eu decidi trocar o sistema de recompensa por algo mais robusto e significativo. Mais uma vez me inspirando em Trackmania, eu usei os checkpoints como recompensa para os agentes e também somava a distância de um carro para o próximo checkpoint, de forma que ele tinha um senso da direção que ele devia ir. Também escrevi um código para matar os agentes que estagnassem na pista ou andassem em círculos. No seu formato atual, esse código mata os agentes que tem uma velocidade média menor do que 40% da melhor velocidade na geração.

***(1)** : Multi-Layer Perceptron: uma rede de neurônios que multiplica valores de entrada contra pesos para obter valores de saída. Os pesos são modificados seguindo diferentes

estratégias, mas até este ponto do projeto eu venho tomando uma aproximação geracional e de seleção dos melhores exemplares para servirem de base para a próxima geração.

***(2)** : (pense que estes são vetores direcionais que partem do centro do carro em uma direção predefinida e tem valores entre 0 e 1 dependendo se ele intersecta com uma parede)

Resumo:

Primeiro Fim de Semana

- Criação do repositório.
 - Escolha da versão do Godot^[2].
 - Decisão de codificar a IA em GDScript^[3], diretamente no jogo.
 - Com auxílio do ChatGPT, defini a abordagem de implementar um MLP^[1-1] básico, utilizando uma array de RayCasts^[4] como entrada e dois valores de saída: um para esterçamento e outro para aceleração/frenagem.
 - Dediquei boa parte do primeiro dia à implementação de uma física decente para os carros. Isso consumiu tempo, mas evitou problemas futuros.
 - Decidi o estilo de arte/jogabilidade inspirado em antigos jogos de flash top-down e na franquia Trackmania^[5] (especialmente Trackmania Nations Forever, TMNF). Utilizei um asset pack^[6] gratuito da itch.io criado por *looneybits*.
 - Ao final do primeiro dia, já havia uma pista com colisão e um carro controlável manualmente.
-

Segundo Dia

- Foco em fazer a IA dirigir o carro.
- Enfrentei dificuldades porque a lógica de controle do carro não estava separada da entrada do jogador humano (decisão difícil de explicar).
- Após refatoração, criei um módulo dedicado ao controle por IA e outro para controle humano (este último removido posteriormente por estar quebrado e por desistir da ideia de competição direta entre player e IA).
- Inicialmente, configurei a MLP com arquitetura 5-8-2 (5 raycasts de entrada, 8 neurônios na camada escondida, 2 saídas).
- O cálculo de fitness era baseado na velocidade total acumulada e na distância total percorrida desde a origem do circuito.
- No início, não havia uma mecânica para desabilitar/matar carros que colidiam com as paredes, resultando em muitos carros realizando "wallride"^[7] (raspando na parede enquanto contornavam curvas).

- Percebi um problema após alguns dias: estava sobrescrevendo os cérebros treinados e mutados com novos cérebros no script do carro, impedindo o aprendizado efetivo.
-

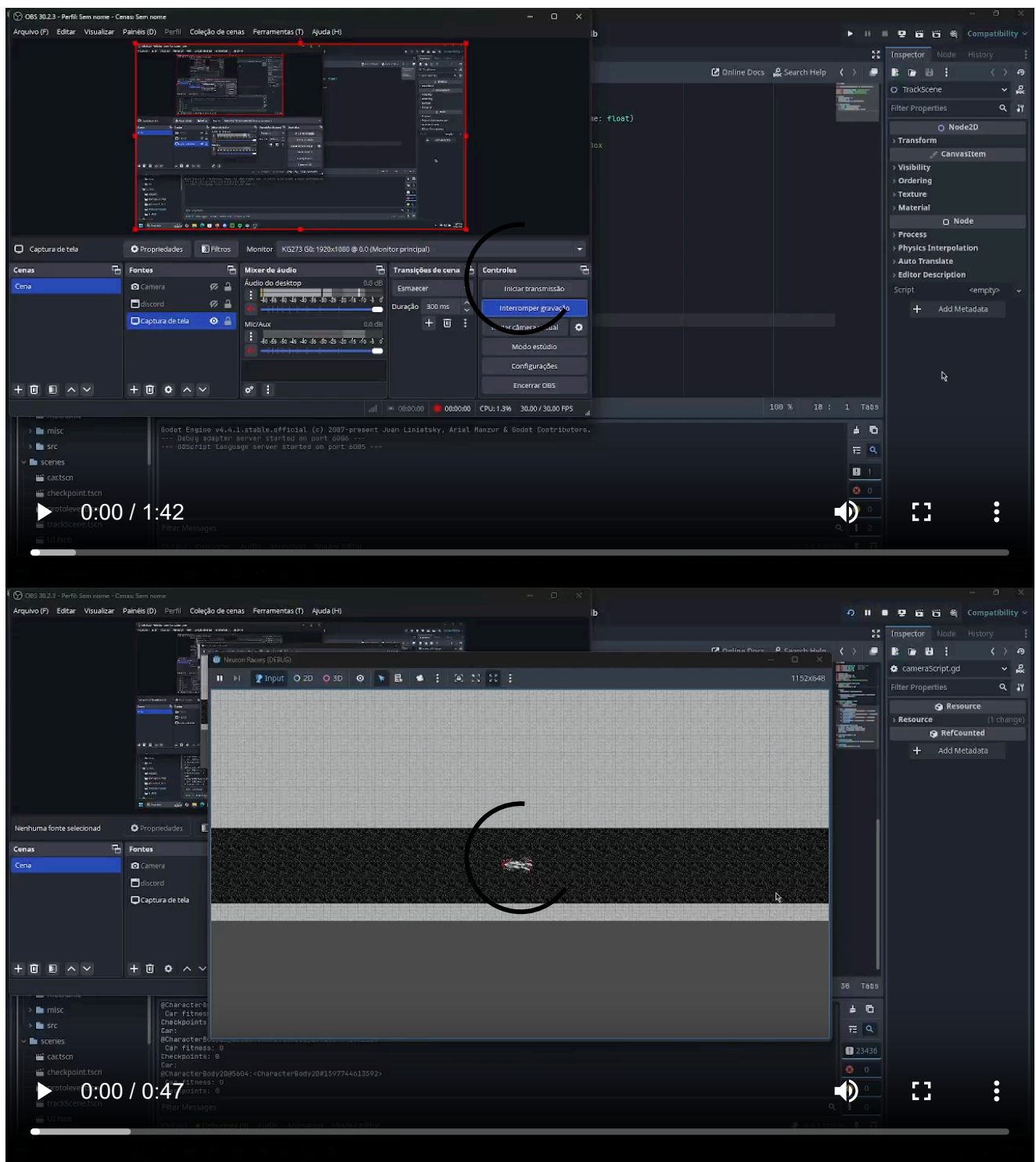
Pós-hiato e Iterações Recentes

- Após alguns dias de pausa, retomei o desenvolvimento e identifiquei o bug que impedia novas gerações de aprenderem com a anterior.
 - Implementei um sistema rudimentar de checkpoints ao longo da pista para dar à IA um senso melhor de progressão (utilizei 36 checkpoints, o que acabou sendo excessivo e mais atrapalhou do que ajudou).
 - Dediquei tempo ao sistema de input, que até o momento (20 de setembro) não utilizei corretamente por ter quebrado de alguma forma.
 - Desde então, o foco tem sido aprender como a IA funciona e refinar o sistema de recompensa.
 - Troquei o sistema de recompensa por algo mais robusto e significativo, inspirado novamente em Trackmania. Os checkpoints passaram a ser usados como recompensa para os agentes, somando também a distância do carro até o próximo checkpoint, dando um senso de direção.
 - Implementei um código para eliminar agentes que estagnassem ou andassem em círculos. Atualmente, agentes com velocidade média menor que 40% da melhor velocidade da geração são eliminados.
-

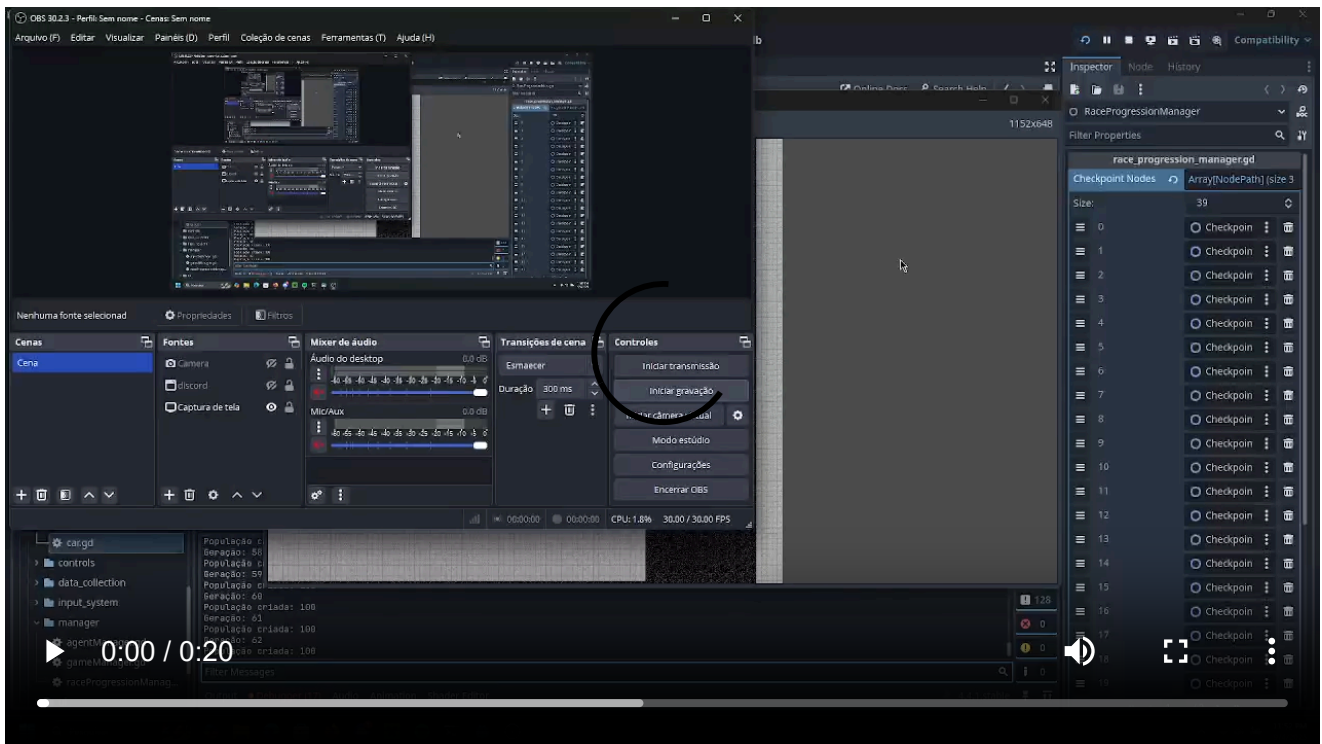
Notas de Rodapé

Vídeos:

03/09



04/09



1. **MLP (Multi-Layer Perceptron):** Rede neural que multiplica valores de entrada por pesos para obter valores de saída. Os pesos são modificados por diferentes estratégias; até este ponto, estou utilizando uma abordagem geracional, selecionando os melhores exemplares para base da próxima geração. ←
2. **Godot:** Engine de desenvolvimento de jogos open-source, popular por sua flexibilidade e suporte a múltiplas plataformas. ←
3. **GScript:** Linguagem de script própria da Godot, similar ao Python, voltada para desenvolvimento rápido de jogos. ←
4. **RayCast:** Vetores direcionais que partem do centro do carro em direções predefinidas, com valores entre 0 e 1 dependendo da interseção com uma parede. ←
5. **Trackmania:** Franquia de jogos de corrida desenvolvida pela NADEO, conhecida por pistas criativas e jogabilidade arcade. ←
6. **Asset pack:** Conjunto de recursos gráficos, sons ou outros elementos prontos para uso em projetos de jogos. ←
7. **Wallride:** Técnica ou comportamento em jogos de corrida onde o veículo desliza ou raspa na parede para contornar curvas, geralmente intencional. ←