Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

Tugas Besar I IF2211 Strategi Algoritma
Semester II Tahun 2021/2022

Pemanfaatan Algoritma Greedy dalam Aplikasi Permainan "Overdrive"

Disusun Oleh:

Ken Kalang AL Qalyubi - 13520010
Gregorius Moses Marevson - 13520052
Panawar H. - 13517129

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA INSTITUT TEKNOLOGI BANDUNG 2022

BAB I : Deskripsi Tugas

I. OverDrive

Overdrive adalah sebuah game yang mempertandingan 2 bot mobil dalam sebuah ajang balapan. Setiap pemain akan memiliki sebuah bot mobil dan masing-masing bot akan saling bertanding untuk mencapai garis finish dan memenangkan pertandingan. Agar dapat memenangkan pertandingan, setiap pemain harus mengimplementasikan strategi tertentu untuk dapat mengalahkan lawannya.

II. Spesifikasi Permainan

Spesifikasi permainan yang digunakan pada tugas besar ini disesuaikan dengan spesifikasi yang disediakan oleh game engine Overdrive pada tautan di atas. Beberapa aturan umum adalah sebagai berikut.

1. Peta permainan memiliki bentuk array 2 dimensi yang memiliki 4 jalur lurus. Setiap jalur dibentuk oleh block yang saling berurutan, panjang peta terdiri atas 1500 block. Terdapat 5 tipe block, yaitu *Empty, Mud, Oil Spill, Flimsy Wall,* dan *Finish Line* yang masing-masing karakteristik dan efek berbeda. Block dapat memuat powerups yang bisa diambil oleh mobil yang melewati block tersebut.
2. Beberapa powerups yang tersedia adalah:
    a. Oil item, dapat menumpahkan oli di bawah mobil anda berada.
    b. Boost, dapat mempercepat kecepatan mobil anda secara drastis.

c. Lizard, berguna untuk menghindari lizard yang mengganggu jalan mobil anda.
d. Tweet, dapat menjatuhkan truk di block spesifik yang anda inginkan.
e. EMP, dapat menembakkan EMP ke depan jalur dari mobil anda dan membuat mobil musuh (jika sedang dalam 1 lane yang sama) akan terus berada di lane yang sama sampai akhir pertandingan. Kecepatan mobil musuh juga dikurangi 3.
f. Bot mobil akan memiliki kecepatan awal sebesar 5 dan akan maju sebanyak 5 block untuk setiap round. Game state akan memberikan jarak pandang hingga 20 block di depan dan 5 block di belakang bot sehingga setiap bot dapat mengetahui kondisi peta permainan pada jarak pandang tersebut.

3. Terdapat command yang memungkinkan bot mobil untuk mengubah jalur, mempercepat, memperlambat, serta menggunakan powerups. Pada setiap round, masing-masing pemain dapat memberikan satu buah command untuk mobil mereka. Berikut jenis-jenis command yang ada pada permainan:
    a. NOTHING
    b. ACCELERATE
    c. DECELERATE
    d. TURN_LEFT
    e. TURN_RIGHT
    f. USE_BOOST
    g. USE_OIL
    h. USE_LIZARD
    i. USE_TWEET
    j. USE_EMP
    k. FIX
4. Command dari kedua pemain akan dieksekusi secara bersamaan (bukan sekuensial) dan akan divalidasi terlebih dahulu. Jika command tidak valid, bot mobil tidak akan melakukan apa-apa dan akan mendapatkan pengurangan skor.
5. Bot pemain yang pertama kali mencapai garis finish akan memenangkan pertandingan. Jika kedua bot mencapai garis finish secara bersamaan, bot yang akan memenangkan pertandingan adalah yang memiliki kecepatan tercepat, dan jika kecepatannya sama, bot yang memenangkan pertandingan adalah yang memiliki skor terbesar.

BAB II : Landasan Teori

I. Dasar Teori

Algoritma Greedy adalah suatu strategi naif yang cenderung sering digunakan dalam persoalan optimasi. Algoritma ini memecahkan persoalan secara langkah per langkah sedemikian sehingga setiap langkah yang dipilih merupakan optimum lokal (tanpa memperhatikan konsekuensi ke depan), dengan harapan akumulasi dari setiap langkah akan membawa pada optimum global. Strategi dalam memilih langkah optimum lokal dipilih secara heuristik sehingga tidak selalu akan memberikan hasil yang optimal.

II. Cara Kerja Program Secara Umum

Program ini adalah program CLI dimana dua buah bot akan bertanding untuk mencapai garis finish. Untuk setiap ronde, program akan meminta satu input dari masing-masing bot. Input akan diberikan oleh bot berdasarkan algoritma yang diimplementasikan pada file "*bot.java*". Input ini kemudian diproses oleh sistem untuk mengupdate game-state. Setiap ronde akan dicatat pada folder match-log. Pertandingan berakhir ketika salah satu bot berhasil mencapai garis finish atau max round sudah tercapai.

Program main berjalan dengan terus menerus meng-update game state dan menerima input dari bot melalui fungsi run(). Algoritma greedy diimplementasikan pada fungsi run() dengan menganalisis game state untuk memberikan command yang optimum lokal. Sebagai contoh: bila tidak ada halangan dalam jangkauan gerak bot, bot akan memberikan command ACCELERATE untuk meningkatkan speed.

BAB III : Aplikasi Strategi Greedy

I. Mapping Persoalan

Persoalan ini dapat dipetakan ke dalam algoritma greedy sebagai berikut :

1. Himpunan Kandidat (C) :
   Semua command yang sesuai dengan syntax aturan, yakni: NOTHING, ACCELERATE, DECELERATE, TURN_LEFT, TURN_RIGHT, USE_BOOST, USE_OIL, USE_LIZARD, USE_TWEET<lane><block>, USE_EMP, FIX.
2. Himpunan Solusi (S) :
   Command-command yang terpilih.
3. Fungsi Solusi :
   Memeriksa apakah command tersebut akan meningkatkan kecepatan bot, menghindari pengurangan skor, dan menghambat pergerakan lawan (berurut dari prioritas tertinggi).
4. Fungsi Seleksi :
   Pilihlah command yang memberikan output kecepatan tertinggi, menghindari pengurangan skor, dan menghambat pergerakan (berurut dari prioritas tertinggi).
5. Fungsi Kelayakan :
   Memeriksa apakah command yang diberikan valid
6. Fungsi Objektif :
   Score akhir lebih tinggi atau sampai ke finish line lebih cepat daripada bot musuh

II. Alternatif Solusi

Alternatif solusi algoritma greedy :

Permainan akan dimenangkan oleh pihak tercepat (ke garis finish) atau pihak dengan score tertinggi. Score tinggi bisa diperoleh dari pengambilan dan pemanfaatan powerups yang tersedia pada game state. Apabila prioritas utama adalah untuk meraih skor tertinggi, strategi ini bisa dimodifikasi menjadi untuk mengambil semua powerups yang tersedia dan mempergunakannya seefektif mungkin.

Untuk mempergunakan power ups dengan keefektifan yang tinggi, peletakan power ups itu haruslah akurat. Salah satu cara untuk mendapatkan akurasi ini adalah untuk memprediksi pergerakan lawan. Umumnya bot musuh yang cukup cerdas akan bergerak seperti pergerakan bot kami. Maka itu, prediksi itu bisa dilakukan dengan menggeser perspektif bot menjadi ke posisi musuh lalu memprediksi input yang akan diberikan ketika menghadapi game state tersebut.

III. Analisis Efisiensi

Memeriksa block di samping kanan, kiri dan depan sejauh n block. Lalu periksa obstacle yang ada.

IV. Analisis Efektifitas

Strategi akan efektif apabila lane tidak mengandung banyak obstacle dan tidak berganti lane. Sedangkan strategi tidak efektif bila banyak obstacle yang sama pada lane.

V. Strategi Greedy yang Diterapkan

Pada setiap round akan dilakukan

1. Pengecekan apakan lawan berada di jarak pandang
    a. jika lawan berada didepan(di lane sekarang), gunakan EMP jika ada, jika tidak akan diusahakan bergerak ke arah lane lain yang mengarah ke finish block.
    b. Jika lawan berada di belakang, akan dicek apakah memiliki OIL, jika ada maka ditembakkan.
2. Pada pemilihan lane yang akan dipilih pada setiap round, dicek pada lane depan, kiri, kanan, lane mana yang mengarah ke finish block. Kemudian dipilih pada lane mana yang akan memperoleh item untuk powerup terbanyak dan powerdown paling sedikit.
3. Jika finish block sudah terlihat pada jarak pandang dan tidak ada lawan ditengah-tengah posisi sekarang dan lawan, dipilih lane tersebut dan mempertahankan posisi.
4. Strategi yang diimplementasikan difokuskan pada fungsi pergerakan mobil seperti ACCELERATE, TURN_RIGHT, TURN_LEFT.

BAB IV : Implementasi

1. Implementasi Strategi Greedy :

Function getBlocksInFront : digunakan untuk mengetahui block yang ada di depan target :

```
private List<Object> getBlocksInFront(int lane, int block, GameState gameState) {
    List<Lane[]> map = gameState.lanes;
    List<Object> blocks = new ArrayList<>();
    int startBlock = map.get(0)[0].position.block;

    Lane[] laneList = map.get(lane - 1);
    for (int i = max(block - startBlock, 0); i <= block - startBlock + Bot.maxSpeed; i++) {
        if (laneList[i] == null || laneList[i].terrain == Terrain.FINISH) {
            break;
        }

        blocks.add(laneList[i].terrain);

    }
    return blocks;
}
```

Function getBlocksOnLeft : digunakan untuk mengetahui block yang ada di kiri target :

```
private List<Object> getBlocksOnLeft(int lane, int block, GameState gameState) {
    List<Lane[]> map = gameState.lanes;
    List<Object> blocks = new ArrayList<>();
    int startBlock2 = map.get(0)[0].position.block;

    Lane[] laneList = map.get(lane - 2);
    for (int i = block - startBlock2; i <= block - startBlock2 + 15; i++) {
        if (laneList[i] == null || laneList[i].terrain == Terrain.FINISH) {
            break;
        }

        blocks.add(laneList[i].terrain);
    }
    return blocks;
}
```

Function getBlocksOnRight : digunakan untuk mengetahui block yang ada di kanan target :

```
private List<Object> getBlocksOnRight(int lane, int block, GameState gameState) {
    List<Lane[]> map = gameState.lanes;
    List<Object> blocks = new ArrayList<>();
```

```
        int startBlock3 = map.get(0)[0].position.block;

        Lane[] laneList = map.get(lane);
        for (int i = block - startBlock3; i <= block - startBlock3 + 15; i++) {
            if (laneList[i] == null || laneList[i].terrain == Terrain.FINISH) {
                break;
            }

            blocks.add(laneList[i].terrain);
        }
        return blocks;
    }
```

2. Struktur Data

Implementasi algoritma greedy terdapat pada file bot.java. File main akan mendapatkan gamestate dari game engine dan menjalankan method run. Bot kemudian melakukan algoritma greedy berdasarkan gamestate yang ada. Terdapat beberapa method fungsi pada bot yang dibuat untuk mengecek kondisi berdasarkan state, yaitu:

1. CheckPowerUp : Fungsi Boolean untuk mencari apakah mobil memiliki power up target.
2. getBlocksOnRight : Untuk mengetahui object di kanan target.
3. getBlocksOnLeft : Untuk mengetahui object di kiri target.

Method run pada bot akan mengembalikan sebuah objek yaitu command yang akan dijalankan. Objek command sudah di define untuk membentuk objek command tersebut jika di assign. Objek command yang direturn akan diterima oleh main.java

BAB V : Kesimpulan dan Saran

I. Kesimpulan

1. Algoritma Greedy tidak selalu bisa menghasilkan hasil yang optimum, tapi bisa memberikan hasil sub-optimum.
2. Penggunaan prioritas greedy berfungsi dengan benar. Namun, masih ada kekurangan yaitu algoritma tidak cukup lincah untuk menentukan command yang terbaik untuk memenangkan pertandingan.

II. Saran

1. Perlu melakukan perbaikan command yang kurang efektif dalam algoritma.

Daftar Pustaka

1. Munir, Rinaldi. 2021. Algoritma Greedy (Bagian 1)

https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag1.pdf. Diakses pada 18 Februari 2022.
2. Entelect Chalenge, 2020, Overdrive, https://github.com/EntelectChallenge/2020-Overdrive Diakses pada 15 Februari 2022.

Link Github : https://github.com/kenkalang/TubesStimaOverDrive