

Tugas Besar 1 IF2211 Strategi Algoritma
Pemanfaatan Algoritma Greedy dalam Pembuatan Bot Permainan
Diamonds



Dipersiapkan oleh kelompok Bawastub-RI:

Ahmad Mudabbir Arif 13522072

Sa'ad Abdul Hakim 13522092

Muhammad Dava Fathurrahman 13522114

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

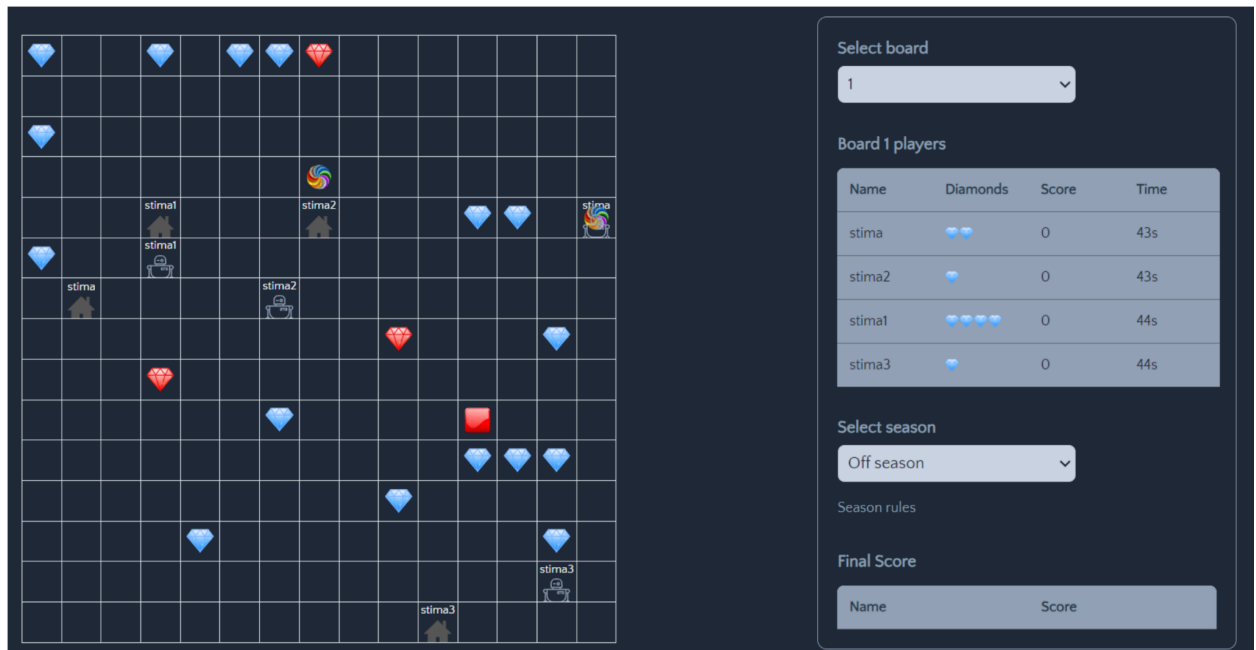
2024

DAFTAR ISI

DAFTAR ISI.....	1
I. Deskripsi Tugas.....	2
II. Landasan Teori.....	3
2.1. Algoritma Greedy.....	3
2.2. Etimo Diamonds.....	4
III. Aplikasi Strategi Greedy.....	6
3.1. Pemetaan Persoalan Diamonds pada Elemen Algoritma Greedy.....	6
3.2. Eksplorasi Alternatif.....	6
3.3. Analisis Efisiensi dan Efektivitas.....	8
3.4. Strategi greedy yang dipilih.....	9
IV. Implementasi dan Pengujian.....	9
4.1. Pseudocode.....	9
4.2. Struktur Data.....	11
4.3. Analisis dari Desain Solusi Algoritma Greedy yang Diimplementasikan.....	11
V. Kesimpulan dan Saran.....	12
5.1. Kesimpulan.....	12
5.2. Saran.....	12
VI. Lampiran.....	13
VII. Daftar Pustaka.....	13

I. Deskripsi Tugas

Diamonds merupakan suatu programming challenge yang mempertandingkan bot yang anda buat dengan bot dari para pemain lainnya. Setiap pemain akan memiliki sebuah bot dimana tujuan dari bot ini adalah mengumpulkan diamond sebanyak-banyaknya. Cara mengumpulkan diamond tersebut tidak akan sederhana itu, tentunya akan terdapat berbagai rintangan yang akan membuat permainan ini menjadi lebih seru dan kompleks. Untuk memenangkan pertandingan, setiap pemain harus mengimplementasikan strategi tertentu pada masing-masing bot-nya. Oleh karena itu, mahasiswa diminta untuk membuat sebuah bot yang nantinya akan dipertandingkan satu sama lain.



Program permainan Diamonds terdiri atas:

1. Game engine, yang secara umum berisi:
 - a. Kode backend permainan, yang berisi logic permainan secara keseluruhan serta API yang disediakan untuk berkomunikasi dengan frontend dan program bot
 - b. Kode frontend permainan, yang berfungsi untuk memvisualisasikan permainan
2. Bot starter pack, yang secara umum berisi:
 - a. Program untuk memanggil API yang tersedia pada backend
 - b. Program bot logic (bagian ini yang akan kalian implementasikan dengan algoritma greedy untuk bot kelompok kalian)
 - c. Program utama (main) dan utilitas lainnya

Komponen-komponen dari permainan Diamonds antara lain:

1. Diamonds
Untuk memenangkan pertandingan, kita harus mengumpulkan diamond ini sebanyak-banyaknya dengan melewati/melangkahnya. Terdapat 2 jenis diamond

yaitu diamond biru dan diamond merah. Diamond merah bernilai 2 poin, sedangkan yang biru bernilai 1 poin. Diamond akan di-regenerate secara berkala dan rasio antara diamond merah dan biru ini akan berubah setiap regeneration.

2. Red Button/Diamond Button

Ketika red button ini dilewati/dilangkahi, semua diamond (termasuk red diamond) akan di-generate kembali pada board dengan posisi acak. Posisi red button ini juga akan berubah secara acak jika red button ini dilangkahi.

3. Teleporters

Terdapat 2 teleporter yang saling terhubung satu sama lain. Jika bot melewati sebuah teleporter maka bot akan berpindah menuju posisi teleporter yang lain.

4. Bots dan Bases

Pada game ini kita akan menggerakkan bot untuk mendapatkan diamond sebanyak banyaknya. Semua bot memiliki sebuah Base dimana Base ini akan digunakan untuk menyimpan diamond yang sedang dibawa. Apabila diamond disimpan ke base, score bot akan bertambah senilai diamond yang dibawa dan inventory (akan dijelaskan di bawah) bot menjadi kosong.

5. Inventory

Bot memiliki inventory yang berfungsi sebagai tempat penyimpanan sementara diamond yang telah diambil. Inventory ini memiliki kapasitas maksimum sehingga sewaktu waktu bisa penuh. Agar inventory ini tidak penuh, bot bisa menyimpan isi inventory ke base agar inventory bisa kosong kembali.

II. Landasan Teori

2.1. Algoritma Greedy

Algoritma greedy adalah pendekatan dalam pemecahan masalah komputasi yang berusaha menemukan solusi optimum dengan cara memilih opsi terbaik pada setiap langkahnya dengan harapan menemukan solusi global yang optimum. Keefektifan algoritma greedy tergantung pada masalahnya, dan tidak semua masalah dapat diselesaikan secara efektif dengan pendekatan ini.

Terdapat beberapa elemen yang perlu didefinisikan pada algoritma greedy, yaitu

1. Himpunan kandidat (C) : kandidat yang mungkin dipilih pada setiap langkahnya.
2. Himpunan solusi (S) : kandidat yang sudah dipilih sebagai solusi
3. Fungsi solusi : fungsi yang menentukan apakah kandidat yang dipilih sudah memberikan solusi.
4. Fungsi seleksi : fungsi yang menyeleksi kandidat berdasarkan strategi greedy tertentu yang bersifat heuristik.
5. Fungsi kelayakan : fungsi yang memeriksa apakah kandidat yang dipilih dari fungsi seleksi dapat dimasukkan ke dalam himpunan solusi.
6. Fungsi objektif : Memaksimumkan atau meminimumkan suatu parameter pada suatu persoalan.

Dalam implementasinya, algoritma greedy memiliki skema umum sebagai berikut

```

function greedy(C : himpunan_kandidat) → himpunan_solusi
{ Mengembalikan solusi dari persoalan optimasi dengan algoritma greedy }
Deklarasi
x : kandidat
S : himpunan_solusi

Algoritma:
S ← {}      { inisialisasi S dengan kosong }
while (not SOLUSI(S)) and (C ≠ {} ) do
    x ← SELEKSI(C)    { pilih sebuah kandidat dari C }
    C ← C - {x}        { buang x dari C karena sudah dipilih }
    if LAYAK(S ∪ {x}) then    { x memenuhi kelayakan untuk dimasukkan ke dalam himpunan solusi }
        S ← S ∪ {x}    { masukkan x ke dalam himpunan solusi }
    endif
endwhile
{ SOLUSI(S) or C = {} }

if SOLUSI(S) then    { solusi sudah lengkap }
    return S
else
    write('tidak ada solusi')
endif

```

Gambar 1: Skema umum algoritma greedy
<https://informatika.stei.itb.ac.id/~rinaldi.munir/>

2.2. Etimo Diamonds

Permainan ini merupakan permainan berbasis web, sehingga setiap aksi yang dilakukan – mulai dari mendaftarkan bot hingga menjalankan aksi bot – akan memerlukan HTTP request terhadap API endpoint tertentu yang disediakan oleh backend.

Berikut adalah urutan requests yang terjadi dari awal mula permainan.

1. Program bot akan mengecek apakah bot sudah terdaftar atau belum, dengan mengirimkan POST request terhadap endpoint `/api/bots/recover` dengan body berisi email dan password bot. Jika bot sudah terdaftar, maka backend akan memberikan response code 200 dengan body berisi id dari bot tersebut. Jika tidak, backend akan memberikan response code 404.
2. Jika bot belum terdaftar, maka program bot akan mengirimkan POST request terhadap endpoint `/api/bots` dengan body berisi email, name, password, dan team. Jika berhasil, maka backend akan memberikan response code 200 dengan body berisi id dari bot tersebut.
3. Ketika id bot sudah diketahui, bot dapat bergabung ke board dengan mengirimkan POST request terhadap endpoint `/api/bots/{id}/join` dengan body berisi board id yang diinginkan (`preferredBoardId`). Apabila bot berhasil bergabung, maka backend akan memberikan response code 200 dengan body berisi informasi dari board.
4. Program bot akan mengkalkulasikan move selanjutnya secara berkala berdasarkan kondisi board yang diketahui, dan mengirimkan POST request terhadap endpoint `/api/bots/{id}/move` dengan body berisi direction yang akan ditempuh selanjutnya (“NORTH”, “SOUTH”, “EAST”, atau “WEST”). Apabila berhasil, maka backend akan memberikan response code 200 dengan body berisi

kondisi board setelah move tersebut. Langkah ini dilakukan terus-menerus hingga waktu bot habis. Jika waktu bot habis, bot secara otomatis akan dikeluarkan dari board.

5. Program frontend secara periodik juga akan mengirimkan GET request terhadap endpoint `/api/boards/{id}` untuk mendapatkan kondisi board terbaru, sehingga tampilan board pada frontend akan selalu ter-update.

Selanjutnya, untuk mengimplementasikan algoritma bot permainan Diamonds, hal yang perlu dilakukan adalah dengan membuat sebuah file Python baru di direktori `/src/game/logic`. Kemudian di dalam file tersebut import class `BaseLogic` dari file `base.py` dan class `GameObject` serta `Board` dari file `models.py` yang ada di direktori `/src/game`. Setelah itu, buatlah sebuah class dengan method `next_move` yang memiliki parameter `self`, `board_bot` dengan tipe `GameObject`, dan `board` dengan tipe `Board`. Dengan menggunakan parameter `board_bot`, kita dapat mengakses status bot saat ini, seperti posisi bot dan base bot. Sedangkan parameter `board` memberikan akses ke status papan permainan seperti lokasi diamonds, tombol diamond, teleporter, bot lain, dan sebagainya. Dalam method `next_move`, akan diterapkan algoritma greedy untuk menentukan arah gerak bot. Algoritma ini akan memilih pilihan lokal optimum dengan harapan mencapai solusi global optimum. Arah gerak yang telah ditentukan akan dikembalikan oleh method `next_move` dalam bentuk nilai `delta_x` dan `delta_y` dan akan dikirimkan ke API endpoint. Berikutnya, untuk menghubungkan algoritma logic bot yang telah dibuat lakukan import class dari file `logic` yang telah dibuat pada file `/src/main.py`. Lalu daftarkan logic bot tersebut pada dictionary `CONTROLLERS` di file `/src/main.py`.

Langkah yang perlu dilakukan untuk menjalankan bot:

- Masuk ke dalam folder `/src`
- Jalankan command berikut pada terminal:

Jika menjalankan satu bot:

```
python main.py --logic your_logic --email=your_email@example.com  
--name=your_name --password=your_password --team etimo
```

Jika menjalankan lebih dari satu bot:

- Untuk windows

```
./run-bots.bat
```

- Untuk Linux / macOS

```
./run-bots.sh
```

dengan mengubah script yang ada pada file `run-bots.bat` atau `run-bots.sh` menyesuaikan pada nama logic yang dibuat, email dan password yang diinginkan.

```

run-bots.bat
1 @echo off
2 start cmd /c "python main.py --logic Random --email=test@email.com --name=stima --password=123456 --team etimo"
3 start cmd /c "python main.py --logic Random --email=test1@email.com --name=stima1 --password=123456 --team etimo"
4 start cmd /c "python main.py --logic Random --email=test2@email.com --name=stima2 --password=123456 --team etimo"
5 start cmd /c "python main.py --logic Random --email=test3@email.com --name=stima3 --password=123456 --team etimo"
6

```

III. Aplikasi Strategi Greedy

3.1. Pemetaan Persoalan Diamonds pada Elemen Algoritma Greedy

- Himpunan kandidat adalah kumpulan seluruh koordinat objek pada papan permainan.
- Himpunan Solusi adalah kumpulan koordinat solusi dari algoritma greedy sesuai dengan strategi greedy yang dipilih. Himpunan solusi meliputi koordinat seluruh objek dalam game yang telah dilewati oleh bot.
- Fungsi Solusi adalah sebuah fungsi yang melakukan pengecekan apakah bot sudah mencapai koordinat tujuan.
- Fungsi Seleksi adalah fungsi untuk memilih koordinat diamond berdasarkan strategi greedy yang diterapkan.
- Fungsi Kelayakan adalah sebuah fungsi yang memeriksa apakah langkah selanjutnya tetap berada papan permainan.
- Fungsi Objektif merupakan fungsi yang memaksimalkan poin diamond yang dikoleksi dalam waktu yang terbatas.

3.2. Eksplorasi Alternatif

Algoritma greedy yang dilakukan dalam eksplorasi alternatif algoritma adalah sebagai berikut.

A. Jarak Terdekat

a. Koordinat Base sebagai acuan

i. Diamond terdekat

Bot akan mencari Diamond apapun, baik Red Diamond dengan bobot 2 maupun Blue Diamond dengan bobot 1. Pada strategi ini, bot akan mencari semua Diamond dengan jarak terdekat dari posisi base bot. Kemudian bot akan bergerak menuju lokasi Red Diamond atau Blue Diamond terdekat dari posisi base bot tersebut. Strategi diatas akan terus dilakukan berulang hingga poin yang didapat sama dengan 5. Bot akan kembali ke base saat poin yang didapat sama dengan 5.

ii. Diamond merah terdekat

Bot akan berfokus untuk mencari Red Diamond yaitu Diamond dengan bobot 2. Pada strategi ini, bot akan mencari Red Diamond dengan jarak terdekat dari posisi base bot ketika poin yang didapat kurang dari 4, kemudian bot akan mencari Blue Diamond dengan jarak terdekat dari posisi base bot saat poin

yang didapat sama dengan 4. Kemudian bot akan bergerak menuju lokasi Red Diamond atau Blue Diamond terdekat dari posisi base bot tersebut. Ketika bot dalam perjalanan menuju Red Diamond dan tidak sengaja melewati Blue Diamond, bot akan mengambil Blue Diamond tersebut, jika pada saat mengambil Blue Diamond tersebut poin menjadi sama dengan 4 maka bot akan mencari Blue Diamond seperti pada kondisi diatas. Strategi diatas akan terus dilakukan berulang hingga poin yang didapat sama dengan 5. Bot akan kembali ke base saat poin yang didapat sama dengan 5.

iii. Diamond biru terdekat

Bot akan berfokus untuk mencari Blue Diamond yaitu Diamond dengan bobot 1. Pada strategi ini, bot akan berfokus hanya pada Blue Diamond saja, dengan melakukan pengecekan jarak antara base dari bot dan Blue Diamond terdekat. Kemudian bot akan bergerak menuju lokasi Blue Diamond terdekat dari posisi bot saat ini. Ketika bot dalam perjalanan menuju Blue Diamond dan tidak sengaja melewati Red Diamond, bot akan mengambil Red Diamond tersebut. Strategi diatas akan terus dilakukan berulang hingga poin yang didapat sama dengan 5. Bot akan kembali ke base saat poin yang didapat sama dengan 5.

b. Koordinat pada saat itu sebagai acuan

i. Diamond terdekat

Bot akan mencari Diamond apapun, baik Red Diamond dengan bobot 2 maupun Blue Diamond dengan bobot 1. Pada strategi ini, bot akan mencari semua Diamond dengan jarak terdekat dari posisi bot. Kemudian bot akan bergerak menuju lokasi Red Diamond atau Blue Diamond terdekat dari posisi bot tersebut. Strategi diatas akan terus dilakukan berulang hingga poin yang didapat sama dengan 5. Bot akan kembali ke base saat poin yang didapat sama dengan 5.

ii. Diamond merah terdekat

Bot akan berfokus untuk mencari Red Diamond yaitu Diamond dengan bobot 2. Pada strategi ini, bot akan mencari Red Diamond dengan jarak terdekat dari posisi bot ketika poin yang didapat kurang dari 4, kemudian bot akan mencari Blue Diamond dengan jarak terdekat dari posisi bot saat poin yang didapat sama dengan 4. Kemudian bot akan bergerak menuju lokasi Red Diamond atau Blue Diamond terdekat dari posisi bot tersebut. Ketika bot dalam perjalanan menuju Red Diamond dan tidak sengaja melewati Blue Diamond, bot akan mengambil Blue

Diamond tersebut, jika pada saat mengambil Blue Diamond tersebut poin menjadi sama dengan 4 maka bot akan mencari Blue Diamond seperti pada kondisi diatas. Strategi diatas akan terus dilakukan berulang hingga poin yang didapat sama dengan 5. Bot akan kembali ke base saat poin yang didapat sama dengan 5.

iii. Diamond Biru terdekat

Bot akan berfokus untuk mencari Blue Diamond yaitu Diamond dengan bobot 1. Pada strategi ini, bot akan berfokus hanya pada Blue Diamond saja, dengan melakukan pengecekan jarak antara bot dan Blue Diamond terdekat. Kemudian bot akan bergerak menuju lokasi Blue Diamond terdekat dari posisi bot saat ini. Ketika bot dalam perjalanan menuju Blue Diamond dan tidak sengaja melewati Red Diamond, bot akan mengambil Red Diamond tersebut. Strategi diatas akan terus dilakukan berulang hingga poin yang didapat sama dengan 5. Bot akan kembali ke base saat poin yang didapat sama dengan 5.

B. Pembobotan Poin Diamond Terhadap Jarak

Pada strategi ini akan dilakukan kalkulasi jumlah poin yang dapat diperoleh dari sebuah diamond, kemudian membaginya dengan jarak dari posisi saat ini ke diamond tersebut. Setelah pembobotan dilakukan untuk semua diamond yang tersedia, bot akan memilih untuk mengambil diamond yang memiliki skor tertinggi. Dengan demikian, bot dapat memaksimalkan keuntungan dengan mengambil diamond dengan poin terbaik relatif terhadap jarak dari posisi saat ini.

C. Perampokan Diamond

Mula-mula, bot akan mencari diamond dengan jarak terdekat. Namun, jika terdapat lawan yang berjarak 1 langkah, bot akan memprioritaskan untuk mengunjungi koordinat tersebut secepat mungkin. Setelah terjadi tackle, bot ini akan melanjutkan pencarian diamond atau kembali ke base jika diamond yang dibawa sudah 5 poin.

3.3. Analisis Efisiensi dan Efektivitas

A. Jarak Terdekat

Kompleksitas dari algoritma ini adalah $O(n)$. Algoritma ini akan membandingkan n pasang koordinat untuk mendapatkan koordinat yang terdekat. Strategi ini hanya memanfaatkan class `GameObject` pada game untuk mendapatkan jarak antara Diamond dengan bot atau *base* kemudian langsung menuju Diamond terdekat dengan cepat. Kelemahan strategi ini adalah efektivitas dalam memilih diamond mana yang harus diambil karena strategi ini hanya mementingkan jarak tetapi tidak poin diamond yang dituju.

B. Pembobotan Poin Diamond Terhadap Jarak Diamond

Kompleksitas dari algoritma ini adalah $O(n)$. Bot akan mengecek seluruh diamond yang ada di papan lalu melakukan pembobotan pada diamond tersebut yaitu dengan menghitung poin diamond per jarak antara diamond dengan suatu koordinat. Strategi ini kemudian akan memilih bobot terbesar dari diamond-diamond yang ada dan menjadikan posisi diamond tersebut sebagai tujuan bot.

C. Perampokan Diamond

Kompleksitas dari algoritma ini adalah $O(1)$. Bot akan mengecek masing-masing satu koordinat di sisi kiri, kanan, atas, dan bawah dari posisi bot berdiri. Algoritma ini kurang efektif karena sensitif terhadap waktu, yaitu perbedaan waktu eksekusi bot dengan lawan. Jika bot lawan mengambil keputusan lebih dahulu, lawan berpotensi merampok bot kami atau sudah menjauh.

3.4. Strategi *greedy* yang dipilih

Berdasarkan eksplorasi alternatif pada sub bab 3.2, kami berhasil merumuskan strategi yang bernama BawastubBot. Strategi ini memiliki karakteristik sebagai berikut,

1. Strategi ini memiliki prioritas pertama untuk melakukan tackle jika terdapat lawan yang berjarak 1 langkah. Apabila gagal melakukan tackle dalam sekali langkah, bot akan menonaktifkan prioritas ini untuk satu langkah selanjutnya.
2. Jika poin diamond yang dibawa sama dengan 5, bot akan pulang ke base.
3. Dengan mempertimbangkan gerak jalan bot tiap satu kotak sekitar 1 detik, bot akan kembali ke base ketika waktu yang tersisa adalah jarak bot dengan base (dalam kotak) dikali dengan 1 detik ditambah dengan 1,75 detik sebagai waktuantisipasi berdasarkan beberapa percobaan yang telah dilakukan, jika waktu belum habis, tetapi bot sudah berada pada base maka bot akan melakukan gerakan random 1 langkah ke arah kanan, kiri, atas, ataupun bawah secara terus menerus hingga waktu habis.
4. Jika jarak antara bot dengan tombol diamond adalah 1 kotak, maka bot akan menuju tombol dan menekan tombol tersebut, hal ini dilakukan karena berdasarkan beberapa percobaan yang dilakukan, dengan menekan tombol tersebut, bot bisa menghasilkan lebih banyak diamond.
5. Algoritma utama pada strategi ini adalah pembobotan poin diamond terhadap jarak diamond dari posisi bot, yaitu dengan mengecek seluruh diamond yang ada di papan pada saat itu dan melakukan pembobotan seluruhnya serta mencari diamond dengan bobot paling besar, lalu bot akan bergerak menuju diamond tersebut.

IV. Implementasi dan Pengujian

4.1. Pseudocode

```
function next_move(C: himpunan_kandidat) → himpunan_solusi
{ Mengembalikan koordinat solusi dari algoritma greedy }

Deklarasi:
    current_post: Position, posisi bot saat ini
```

```

base: Position, posisi base
props: Properties, properti bot
diamond: himpunan DiamondGameObject
button: himpunan DiamondButtonGameObject
goal: GameObject
delta_x: integer, perubahan koordinat x untuk langkah selanjutnya
delta_y: integer, perubahan koordinat y untuk langkah selanjutnya
justTackle: boolean

```

Algoritma:

```

current_post ← board_bot.position
base ← board_bot.properties.base
props ← board_bot.properties
diamond ← {} { inisialisasi diamond sebagai himpunan kosong }
button ← None ; goal ← None
delta_x ← 0 ; delta_y ← 0

{ Tackle First if one move (next) there is a bot }
if justTackle then
    justTackle ← false
else
    bot_positions ← {obj.position | obj in board.game_objects if
obj.type == 'BotGameObject'}

    for direct in directions do
        new_x ← current_post.x + direct[0]
        new_y ← current_post.y + direct[1]

        { jika posisi selanjutnya masih di dalam papan }
        if 0 ≤ new_y < board.height and 0 ≤ new_x < board.width then
            if Position(x=new_x, y=new_y) in bot_positions then
                justTackle ← true
                return direct
            endif
        endif

    endfor

    { GO HOME }
    if props.diamonds == 5 then
        delta_x, delta_y ← direction(current_post.x, current_post.y, base.x,
base.y, board.width, board.height)
        return delta_x, delta_y

    for obj in board.game_objects do
        { mengumpulkan diamond dan mencari button }
        if obj.type == "DiamondGameObject" then
            diamond ← diamond U {obj}
        elif obj.type == "DiamondButtonGameObject" then
            button ← obj

        { Jika waktu tersisa kurang dari jarak ke base ditambah buffer }
        if props.milliseconds_left < ((abs(current_post.x - base.x) +
abs(current_post.y - base.y)) * 1000) + 1750 then
            delta_x, delta_y ← direction(current_post.x, current_post.y, base.x,
base.y, board.width, board.height)
            return delta_x, delta_y
        else
            { Jika bot berada di sebelah button }
            if abs(current_post.x - button.position.x) + abs(current_post.y -
button.position.y) ≤ 1 then
                delta_x, delta_y ← direction(current_post.x, current_post.y,
button.position.x, button.position.y, board.width, board.height)
            else
                for gem in diamond do

```

```

        { Pilih diamond terdekat yang memiliki poin lebih besar }
        if props.diamonds == 4 and gem.properties.points == 2 then
            continue
        if goal = None then
            goal ← gem
        else
            if point_per_distance(current_post.x, current_post.y, gem)
> point_per_distance(current_post.x, current_post.y, goal) then
                goal ← gem
            delta_x, delta_y ← direction(current_post.x, current_post.y,
goal.position.x, goal.position.y, board.width, board.height)

        → delta_x, delta_y { solusi berupa koordinat }

```

4.2. Struktur Data

Program ini menggunakan beberapa struktur data untuk menyimpan informasi terkait permainan. Informasi tersebut dapat berupa detail objek dalam permainan. Struktur data yang digunakan terdiri atas struktur data yang sudah ada pada kit dan struktur data baru yang dibuat untuk menyelesaikan persoalan. Struktur data yang digunakan antara lain:

1. Game Object, yaitu struktur data untuk menyimpan informasi terkait objek dalam permainan seperti posisi tipe, dan properti objek.
2. Board, yaitu struktur data untuk menyimpan informasi terkait papan permainan seperti ukuran, senarai berisi objek-objek dalam permainan, dan sebagainya.
3. Position, yaitu struktur data untuk menyimpan informasi tentang posisi objek dalam permainan berupa koordinat.
4. List of Tuple, yaitu struktur data untuk menyimpan informasi koordinat dan objek game.

4.3. Analisis dari Desain Solusi Algoritma Greedy yang Diimplementasikan

Strategi algoritma greedy yang kami implementasikan didasari atas pembobotan poin diamond terhadap jaraknya. Pemilihan strategi pembobotan poin diamond terhadap jaraknya terbukti optimal setelah kami bandingkan dengan strategi mengunjungi diamond dengan jarak terdekat. Menurut kami, strategi ini lebih efisien dalam mengumpulkan poin karena telah meminimumkan langkah dalam mengambil beberapa diamond. Strategi ini juga terbukti optimal saat posisi diamond merah memiliki jarak lebih jauh sedikit daripada diamond biru, maka dengan algoritma ini bot akan berjalan ke arah diamond merah karena berdasarkan pembobotan nilai diamond merah akan lebih besar daripada diamond biru jika jarak keduanya dengan posisi bot tidak berbeda jauh.

Pemilihan tackle sebagai prioritas pertama tidak dilakukan sembarang, dalam pengujiannya kami mendapatkan “Circle of Death” jika lebih dari tiga bot saling mengejar untuk melakukan tackle dan tidak ada tackle yang berhasil dilakukan. Terkadang, lawan yang dikejar berbalik arah sehingga mentackle bot kami lebih dahulu. Oleh karena itu, kami membatasi pengajaran hanya satu kali langkah, baik berhasil maupun tidak. Dua langkah berikutnya, prioritas tackle bot akan diaktifkan kembali.

Strategi algoritma greedy yang kami implementasikan juga memiliki langkah yang sudah kami modifikasi yaitu gerakan langkah tidak selalu secara konstan menyamakan

koordinat x dari tujuan lalu baru menuju arah koordinat y dari tujuan. Langkah dilakukan secara random tetapi tetap mendekati arah tujuan. Hal tersebut kami lakukan karena berdasarkan beberapa percobaan yang kami lakukan, algoritma tersebut terbukti efektif menghasilkan nilai yang optimal terutama ketika ada objek teleport pada jalan yang ingin dituju. Pada kondisi tersebut, algoritma dapat menghindari teleport karena arah langkah yang tidak tentu. Meskipun demikian, algoritma ini memiliki kelemahan yang dapat membuat solusi tidak optimal yaitu ketika arah jalan yang tidak tentu tersebut justru mengarah ke arah objek teleport lain yang dapat menjauhkan bot dengan tujuan yang diinginkan.

Strategi optimalisasi pengumpulan diamond berdasarkan waktu yang ada juga kami implementasikan pada algoritma yang kami buat yaitu dengan memperhatikan jarak bot dengan *base* dan waktu yang tersisa. Strategi tersebut terbukti menghasilkan nilai yang optimal karena dari percobaan yang kami lakukan seringkali pada saat waktu permainan telah habis, bot masih memiliki diamond pada penyimpanannya dan bot tidak sempat untuk mencapai *base* untuk menjadikan diamond tersebut poin yang didapat. Sehingga dengan algoritma ini, meskipun poin yang didapat oleh bot belum mencapai maksimal yaitu 5, bot akan kembali ke *base* saat waktu yang tersisa tinggal sedikit bergantung pada jarak bot dengan *base*, dengan begitu diamond yang ada pada bot akan menambah skor yang dihasilkan.

Algoritma greedy yang kami implementasikan juga memperhatikan tombol diamond yang ada pada papan sehingga jika ada tombol diamond berjarak 1 kotak dengan bot maka akan ditekan oleh bot. Algoritma tersebut optimal berdasarkan percobaan yang kami lakukan karena dengan menekan tombol diamond tersebut, jumlah diamond yang ada pada papan akan kembali banyak sehingga bot dapat menghasilkan poin yang lebih tinggi karena bot dapat mengumpulkan diamond lebih banyak.

V. Kesimpulan dan Saran

5.1. Kesimpulan

Pada Tugas Besar 1 Strategi Algoritma ini, kami telah menerapkan algoritma greedy dalam pembuatan bot permainan Etime Diamonds. Tugas ini telah memberikan pemahaman terkait algoritma greedy dan kemampuannya dalam penyelesaian suatu masalah dengan optimalisasi di setiap langkahnya. Dengan algoritma greedy, kami tidak perlu menangani banyak skenario karena konsep algoritma greedy adalah mencari solusi lokal untuk mencapai solusi global yang optimum

5.2. Saran

1. Lebih banyak melakukan eksplorasi terhadap parameter yang ada di permainan diamonds.
2. Lebih banyak melakukan pengujian algoritma greedy dengan bot kelompok lain sehingga dapat mengukur kemampuan bot lebih jauh lagi.

VI. Lampiran

Repositori GitHub : https://github.com/mdavaf17/Tubes1_Bawastub-RI

Tautan video kelompok : https://youtu.be/yTaW7E8i_hI

VII. Daftar Pustaka

Munir, Rinaldi. 2024. “Algoritma Greedy (Bagian 1)”.

[https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-\(2021\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Greedy-(2021)-Bag1.pdf)

(Diakses pada 9 Maret 2024)

Etimo. (2020). Etimo Diamonds. GitHub.

<https://github.com/Etimo/diamonds>