

Implementasi Algoritma DFS untuk Pergerakan Ghost di Permainan Pac Man

Muhammad Ardhin

13509033

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

13509033@std.stei.itb.ac.id

Abstract

Pac-Man adalah sebuah permainan *arcade* yang dikembangkan oleh Namco. Pac-Man dikenal sebagai permainan klasik yang pernah menjadi ikon *videe game* pada dekade 80-an, dan sampai sekarang adalah salah satu permainan paling populer sepanjang masa, yang juga merupakan permainan dengan pendapatan kotor terbesar.

Salah satu elemen penting yang membuat Pac-Man menarik adalah kehadiran Ghost, yaitu musuh yang senantiasa mengganggu permainan dan harus dimusnahkan. Algoritma DFS, atau Depth-First-Search, adalah sebuah algoritma pencarian mendalam yang dapat digunakan untuk mengatur pergerakan Ghost, sehingga pergerakannya lebih menarik. Makalah ini difokuskan pada implementasi DFS untuk pergerakan Ghost di permainan Pac-Man

Index Terms — Algoritma, DFS, Ghost, Pac-Man

I. PENDAHULUAN

Pac-Man digerakkan untuk menelusuri labirin, mengumpulkan sesuatu yang disebut sebagai “Pac-Dots”. Permainan berganti level jika “Pac-Dots” telah habis dikumpulkan. Selama Pac-Man bergerak menelusuri labirin, Pac-Man akan berhadapan dengan 4 musuh, yang disebut Ghost, yang masing-masing bernama Blinky, Pinky, Inky, dan Clyde, yang akan terus bergerak mengejar Pac-Man. Jika salah satu dari mereka menyentuh Pac-Man, maka Pac-Man akan kehilangan nyawa, dan permainan di level itu dimulai kembali. Permainan diulang kembali sampai semua nyawa Pac-Man terpakai, dan setelah itu permainan mencapai tahap *Game Over*.

Terdapat pula “titik-titik” yang lebih besar dari “Pac-Dots” yang disebut “Power Pellets”, yang jika dimakan akan membuat Pac-Man mampu membalikkan keadaan untuk sementara waktu dan mengejar para Ghost. Jika Pac-Man menyentuh Ghost, Ghost akan hilang termakan dan Pac-Man akan mendapat poin tambahan.

Poster Pac-Man



Sumber :

<http://home.comcast.net/~jpittman2/pacman/pacmandossier.html>

Dari penjabaran diatas, diketahui bahwa pergerakan Ghost ada 2, yakni pergerakan saat mengejar Pac-Man, dan pergerakan saat menjauhi Pac-Man. Dalam bergerak, setiap Ghost mempunyai algoritma sendiri karena setiap Ghost memang didesain dengan kepribadian masing-masing. Setiap Ghost mengejar Pac-Man dengan cara yang berbeda. Akan tetapi, algoritma DFS dapat diimplementasikan untuk membuat pergerakan Ghost lebih menarik. Dengan BFS, Ghost mencari solusi jalur mana yang merupakan jalur terbaik untuk mendekati Pac-Man, atau pun jalur terbaik untuk menjauhi Pac-Man.

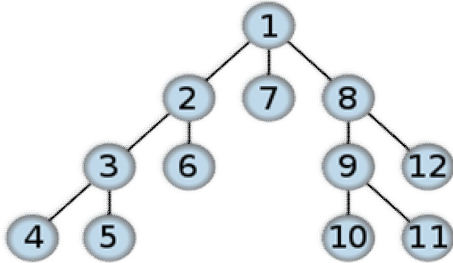
Tampilan Utama Permainan Pac-Man



Sumber : <http://gameinternals.com/post/2072558330/understanding-pac-man-ghost-behavior.htm>

Algoritma DFS adalah algoritma untuk menranversalkan pohon, struktur pohon, atau graf. Dimulai dari akar, pencarian dilakukan sejauh mungkin sebelum dilakukan “backtracking”, kecuali dengan aturan-aturan tertentu. Algoritma DFS banyak digunakan untuk penyelesaian masalah-masalah yang berhubungan dengan labirin.

Depth-First-Search



Sumber : http://en.wikipedia.org/wiki/Depth-first_search

Semua karakter di permainan Pac-Man, yakni Pac-Man itu sendiri, dan juga semua Ghost, bergerak dalam sebuah labirin. Karena itulah, algoritma DFS merupakan algoritma yang tepat untuk diimplementasikan pada pergerakan Ghost, atau pun pada pergerakan Pac-Man jika ada Pac-Man ingin digerakkan secara otomatis secara lebih menarik.

II. DASAR TEORI

Algoritma DFS membantu ruang solusi dari solusi-solusi yang ada, dimana ruang solusi diorganisasikan menjadi struktur pohon. Pencarian dilakukan dengan mengunjungi simpul-simpul pohon. Dalam hal ini, pohon yang ditelusuri adalah pohon dinamis, yakni pohon yang dibangun selama pencarian solusi berlangsung.

Traversal dimulai dari sebuah simpul, simpul yang bertangga dengannya, lalu DFS diaplikasikan lagi di simpul tetangga tersebut, dst. “Backtrack” dilakukan ketika semua simpul-simpul yang bertetangga telah dikunjungi, dan pencarian selesai saat tidak ada lagi simpul yang belum dikunjungi dari simpul yang dapat dicapai.

Prosedur DFS Secara Umum

```

procedure DFS(input v:integer)
(Mengunjungi seluruh simpul graf dengan algoritma pencarian DFS
Masukan: v adalah simpul awal kunjungan
Keluaran: semua simpulyang dikunjungi ditulis ke layar
)
Deklarasi
w : integer
Algoritma:
write(v)
dikunjungi[v]←true
for tiap simpul w yang bertetangga dengan simpul v do
if not dikunjungi[w] then
DFS(w)
endif
endfor
  
```

Sumber : <http://www.informatika.org/~rinaldi/Smik/2006-2007/BFS%20dan%20DFS.pdf>

Prosedur DFS Secara Detail

```

procedure DFS(input v:integer)
(Mengunjungi seluruh simpul graf dengan algoritma pencarian DFS
Masukan: v adalah simpul awal kunjungan
Keluaran: semua simpulyang dikunjungi ditulis ke layar
)
Deklarasi
w : integer
Algoritma:
write(v)
dikunjungi[v]←true
for w←1 to n do
if A[v,w]=1 then (simpul v dan simpul w bertetangga )
if not dikunjungi[w] then
DFS(w)
endif
endif
endfor
  
```

Sumber : <http://www.informatika.org/~rinaldi/Smik/2006-2007/BFS%20dan%20DFS.pdf>

III. IMPLEMENTASI

A. Implementasi Secara Umum

Pada dasarnya, algoritma DFS adalah algoritma yang menelusuri suatu struktur pohon sampai kedalaman terdalam baru berpindah ke simpul lain. Dianalogikan dengan labirin, maka pergerakan karakter adalah menelusuri maze sampai batasnya, lalu kembali, “backtrack”, hingga karakter tersebut sampai di persimpangan, lalu menelusuri jalur lainnya.

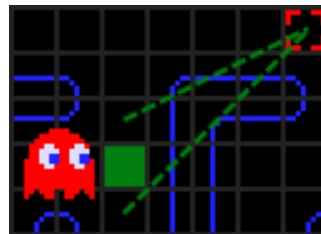
- Kunjungi sebuah percabangan
- Telusuri, hingga bertemu percabangan lagi
- Kunjungi percabangan hingga bertemu ujung jalan, atau percabangan lainnya
- Ulangi DFS

B. Implementasi untuk Pergerakan Mengejar (Chase Mode)

Pergerakan mengejar adalah pergerakan Ghost dengan tujuan berkolisi dengan Pac-Man

Sebelum menerapkan bergerak, Ghost perlu melakukan kalkulasi terlebih dahulu untuk menemukan posisi Pac-Man. Setelah menemukan posisi Pac-Man, Ghost akan bergerak menuju Pac-Man. Implementasi algoritma DFS yang sesungguhnya dilakukan setelah Ghost bergerak.

Decision



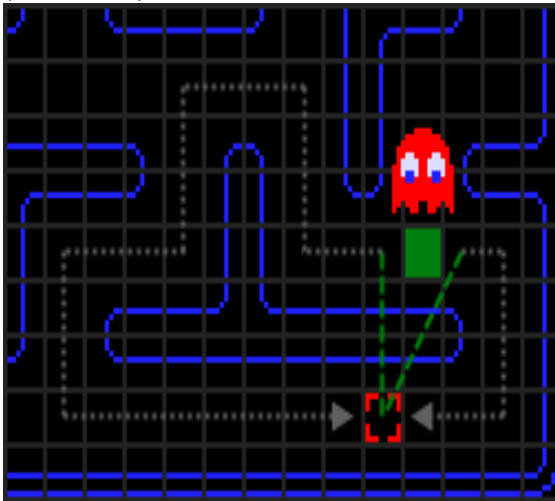
Sumber : <http://gameinternals.com/post/2072558330/understanding-pac-man-ghost-behavior>

Pac-Man akan selalu bergerak, sesuai dengan keinginan

pemain. Akan tetapi, Pac-Man tidak selalu bergerak tanpa bisa ditebak. Pac-Man akan bergerak menyusuri suatu jalan labirin sebelum akhirnya menemukan sebuah persimpangan, dimana saat itu pergerakan Pac-Man tidak bisa ditebak (mengikuti kemauan pemain).

Ghost akan bergerak dari satu petak ke petak lainnya (lahan permainan dibagi atas petak-petak). Kalkulasi dilakukan untuk menentukan posisi Pac-Man setiap saat Ghost berpindah petak, sehingga dapat ditetapkan satu dari dua jalan yang terdekat kepada Pac-Man, sehingga Ghost dapat dipastikan akan selalu bergerak mendekati Pac-Man, bukan menjauhi.

(Chance of a) Bad Decision



Sumber : <http://gameinternals.com/post/2072558330/understanding-pac-man-ghost-behavior>

Barulah saat Ghost sampai di perjalanan, Ghost perlu membuat pilihan. Penelusuran DFS saat ini dibutuhkan agar Ghost dapat menentukan rute terbaik untuk mendekati kepada Pac-Man.

Setelah menentukan jalur yang tepat, Ghost akan bergerak menuju Pac-Man, dan di titik ini, pergerakan Ghost dapat dikembangkan kembali menggunakan “kepribadian” masing-masing agar permainan menjadi lebih menarik. Dapat disimpulkan, DFS, atau penelusuran labirin digunakan hanya untuk menentukan arah gerak Ghost di setiap titik persimpangan sebelum bergerak lagi dengan cara yang dikehendaki, bisa juga dengan kembali menggunakan DFS.

Penggunaan algoritma DFS ini adalah sebagai alternatif. Jika tidak menggunakan DFS, Ghost juga dapat mengkalkulasikan rute terbaik menggunakan algoritma BFS. Akan tetapi, dengan DFS, selalu ada peluang Pac-Man akan lebih cepat terdeteksi, dan dengan begitu Ghost akan lebih “pintar” karena lebih cepat menemukan dan mengejar Pac-Man.

- Tentukan posisi Pac-Man setiap waktu menggunakan kalkulasi matematis, bisa dengan menggunakan trigonometri, setidaknya setiap

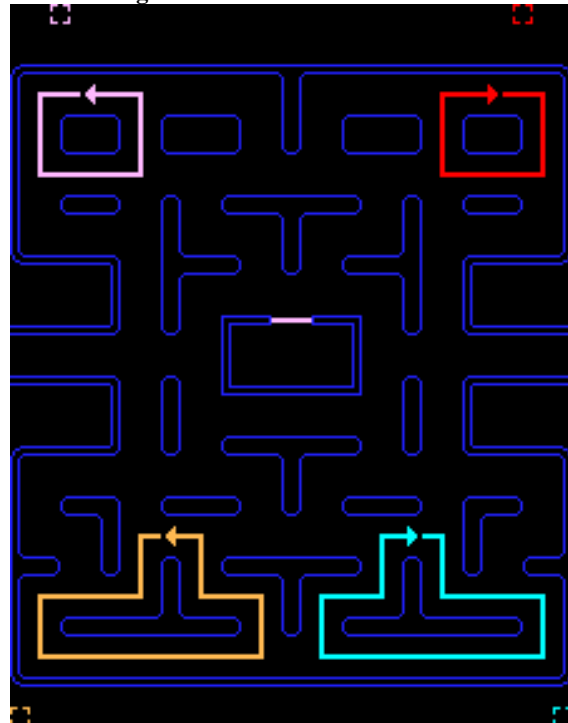
kali Ghost sampai di petak yang berbeda

- Tentukan pilihan dengan memilih jarak terdekat, sehingga Ghost tidak akan bergerak menjauhi Pac-Man
- Saat Ghost sampai di persimpangan, tentukan kembali jalur terdekat menggunakan algoritma DFS, karena pada dasarnya kalkulasi selama ini juga adalah penelusuran, akan tetapi tidak membutuhkan proses yang rumit, karena hanya ada 2 kemungkinan : Menjauh atau Mendekat, dan bisa dilakukan percabangan biasa
- Setelah posisi ditentukan, jalankan kembali Ghost

C. Implementasi untuk Pergerakan Kabur (Scatter Mode)

Pada dasarnya, setiap Ghost akan bergerak menuju sebuah daerah tertentu yang menjadi “rumah” bagi mereka (menggunakan algoritma *path finding* masing-masing) dan akan mulai berputar-putar untuk jangka waktu tertentu selama Pac-Man mendapat hak untuk memangsa mereka. Biasanya, dikarenakan waktu “kabur” yang amat singkat, Ghost tidak sempat kembali ke rumahnya jika Ghost sudah terlalu jauh berjalan, atau paling tidak, Ghost tidak dapat melakukan banyak putaran di daerahnya.

Scatter-Target

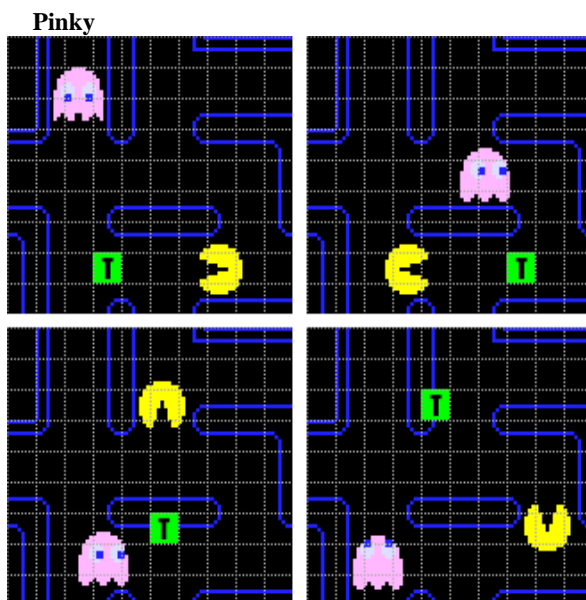


Sumber : <http://gameinternals.com/post/2072558330/understanding-pac-man-ghost-behavior>

Akan tetapi, dengan algoritma DFS, dapat dikembangkan “kecerdasan” kabur yang lebih baik, yaitu dengan memutar balikkan algoritma untuk mengejar.

Ghost akan memperhitungkan posisi Pac-Man, dan kemudian mencari rute terjauh dari Pac-Man, dan saat Ghost berada di persimpangan, DFS akan kembali diimplementasikan .

- Tentukan posisi Pac-Man setiap waktu menggunakan kalkulasi matematis, bisa dengan menggunakan trigonometri, setidaknya setiap kali Ghost sampai di petak yang berbeda
- Tentukan pilihan dengan memilih jarak **terjauh**, sehingga Ghost tidak akan bergerak **mendekati** Pac-Man
- Saat Ghost sampai di persimpangan, tentukan kembali jalur **terjauh** menggunakan algoritma DFS
- Setelah posisi ditentukan, jalankan kembali Ghost



Sumber :
<http://home.comcast.net/~jpitman2/pacman/pacmandossier.html>

IV. KESIMPULAN DAN SARAN

Permainan Pac-Man adalah permainan yang telah mendunia dan telah banyak dibuat dan dikembangkan dalam berbagai versi. Alangkah banyak cara-cara yang digunakan untuk membuat permainan ini lebih menarik, diantaranya dengan memasukkan algoritma-algoritma baru dan menggabungkannya dengan algoritma dasarnya.

Algoritma DFS adalah sebuah algoritma yang sangat dibutuhkan dan biasa digunakan dalam pemecahan masalah labirin. Pac-Man yang merupakan permainan labirin cocok mengimplementasikan algoritma DFS.

Dalam pergerakan Ghost yang otomatis, dibutuhkan algoritma-algoritma tertentu sehingga Ghost bisa bergerak secara menarik. Dalam hal ini, algoritma DFS diimplementasikan pada pergerakan Ghost saat Ghost berada di sebuah percabangan. Algoritma DFS akan

menelusuri dan mengkalkulasi semua jalan yang mungkin, dan kemudian menghasilkan sebuah jalur.

Dalam mode mengejar Pac-Man, algoritma DFS akan mengkalkulasi dan menyusuri rute-rute yang mungkin, dan menghasilkan jalur terdekat.

Dalam mode kabur, algoritma DFS dapat dimanfaatkan untuk membuat Ghost lebih “cerdas” agar dapat menghindari Pac-Man secara lebih menarik, dan tidak sekedar kabur ke rumahnya dan berputar-putar menunggu habisnya masa penggunaan “Power Pellet”.

Di masa depan, sebaiknya lebih banyak pembahasan mengenai permainan Pac-Man, sehingga mahasiswa dapat belajar lebih banyak mengenai aplikasi algoritma-algoritma yang dipelajari di kuliah dan lebih bersemangat mengimplementasikannya.

V. ACKNOWLEDGMENT

Penulis ingin berterima kasih kepada Allah SWT, yang telah menganugerahkan kesempatan mempelajari teknik informatika, dan juga telah menghadirkan permainan-permainan seru di dunia ini, lewat tangan-tangan para pencipta *video game*, yang salah satunya adalah Pac-Man. Penulis berterima kasih kepada dosen penulis, Rinaldi Munir, yang telah memberikan tugas makalah ini, sehingga penulis mendapat semangat untuk mendalami suatu permasalahan, dan pada akhirnya dapat memahami algoritma pergerakan Ghost di permainan Pac-Man. Harapannya, penulis dapat mengimplementasikan permainan Pac-Man versi penulis di masa depan, mungkin di saat liburan.

REFERENCES

- [1] Chandra, Timotius Nugroho. *Aplikasi Algoritma Greedy untuk Pergerakan Musuh pada Permainan Pac-Man*
- [2] Istiqomah, Anis. *Penyelesaian Permainan “Pacman” yang disederhanakan dengan Algoritma Backtracking*
- [3] Munir, Rinaldi. *Strategi Algoritma*. Bandung : Januari 2009
- [4] Widiyadi, Emeraldy. *Aplikasi Algoritma BFS pada Hantu dalam Permainan Pac-Man*
- [5] http://en.wikipedia.org/wiki/Depth-first_search
- [6] <http://en.wikipedia.org/wiki/Pac-Man>
- [7] <http://www-scf.usc.edu/~csci561b/assignment/project1.pdf>
- [8] <http://www.daniweb.com/software-development/python/threads/71210>
- [9] <http://games.slashdot.org/story/10/12/03/2237200/pac-mans-ghost-behavior-algorithms>
- [10] http://donhodes.com/pacman_pinky_explanation.htm
- [11] <http://stackoverflow.com/questions/2604022/pathfinding-algorithm-for-pacman>

- [12] <http://www.daniweb.com/software-development/python/threads/71210>
- [13] <http://www.programmersheaven.com/mb/game/165986/165986/pacman-ghost-movement/>
- [14] <http://stackoverflow.com/questions/7437489/able-to-find-path-using-dfs-but-not-able-specify-the-right-directions-to-pacman>
- [15] <http://gameinternals.com/post/2072558330/understanding-pac-man-ghost-behavior>
- [16] <http://home.comcast.net/~jpittman2/pacman/pacmandossier.html>
- [17] <http://stackoverflow.com/questions/3101378/pathfinding-algorithm-for-2-pacmans>

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 9 Desember 2011

ttd



Muhammad Ardhin
13509033