BAB V

DESAIN GAME

Dalam bab ini akan dijelaskan tahap – tahap dalam menentukan aspek – aspek dalam membuat game Tugas Akhir ini. Diantaranya adalah UI, karakter, level tutorial dan UX. Nanti akan dijelaskan dalam detail apa yang dipertimbangkan dalam membuat aspek tersebut.

1. Flow aplikasi dan UX

Flow aplikasi adalah sebuah alur dimana aksi – aksi dapat mengganti state dari aplikasi atau dalam kasus game ini layar yang ditampilkan. Di gambar 5.1 dibawah ini ditunjukkan alur dari game “Splatted”.



Gambar 5.1

Flow dari Splatted

Dimulai dari Main menu, akan ada beberapa tombol yang menuju ke beberapa opsi, diantaranya adalah tombol untuk pergi ke tutorial, setting, keluar dari game dan memulai permainan. Tombol tutorial akan mengantar ke player ke level tutorial untuk mengajari dasar bermain Splatted, sama seperti tombol start dan juga tombol settings, dengan pengecualian tombol Exit yang mengeluarkan player dari game.

Bila tombol start game dijalankan, maka akan diadakan pengecekan apabila Tutorial pernah dijalankan. Bila tutorial belum pernah dijalankan, maka player akan diarahkan menuju tutorial level agar para player mengerti apa yang perlu dilakukan, tapi tutorial bisa diskip agar pemain yang tidak sabar bisa langsung masuk ke dalam aksi.

Setelah menyelesaikan tutorial melalui start game atau player pernah menyelesaikan tutorial maka player akan diarahkan menuju level settings, dimana akan ditampilkan beberapa opsi opsi yang disediakan dalam membuat sebuah level. Di Level setting terdapat opsi yang ada secara default dipilih agar bila player bingung pilihan tersebut dapat diabaikan.

Dari layar level settings player bisa memencet tombol start dan mulai membuat sebuah level. Akan ditampilkan layar baru yang menunjukkan bila sebuah level sedang dibuat dan setelah level selesai dibuat maka layar tersebut akan diganti menuju layar level utama dari game tersebut.

Terakhir adalah pause yang tersedia di tutorial dan level utama. Pause tersedia untuk memberi opsi player menghentikan game selama yang diperlukan dan bila diinginkan bisa digunakan untuk kembali ke main menu. Bila sudah siap, pause dapat ditutup untuk melanjutkan permainan.

1. Gameplay

Splatted adalah sebuah game dimana tim kita dengan tim musuh akan melempar bola ke satu sama lain untuk mendapatkan skor yang dipakai untuk menentukan pemenang dari permainan. Secara gameplay Splatted akan menjadi sebuah game dengan pacu cukup cepat dengan menggunakan bola yang muncul di tanah dan bola spesial yang tersebar di dalam level.

Dalam game ini akan ada pertarungan antara 2 tim masing – masing berisi 5 orang, dimana setiap orang dapat melempar bola dan bila bola tersebut kena tim lawan, maka skor dari tim pelempar akan bertambah. Bila waktu yang ditentukan sudah habis atau salah satu tim telah mencapai target skor, maka game akan dihentikan dan pemenang dari game akan dinyatakan.

Menggunakan konsep oper – operan dari Knockout City, di game ini player bisa melempar ke teman dan bila teman tersebut menangkap bolanya, maka bola akan menjadi lebih cepat dan bisa mendapat lebih banyak skor bila dilempar ke musuh. Tapi konsep ini juga berlaku untuk musuh, dimana bila musuh berhasil menangkap bola, bola juga menjadi lebih cepat dan mendapat skor lebih banyak. Terakhir bila player melempar ke teman setim dan teman tersebut gagal menangkap bola tersebut, maka teman tersebut akan kena stun tapi tidak memberi skor apapun ke tim teman maupun tim musuh.

1. Main Level

Setelah menentukan visi gameplay, sekarang akan didesain level yang akan digunakan sebagai lokasi permainan antara 2 tim tersebut. Setiap beberapa detik akan muncul beberapa gumpalan salju yang dapat diambil untuk dilempar.Bola tersebut akan dibuat mencolok dari latar belakang agar bisa mudah dilihat

Tidak hanya itu, terdapat juga beberapa bola spesial, terinspirasi dari Knockout City yang dapat muncul di titik – titik tertentu. Setelah beberapa detik titik – titik tersebut dapat memunculkan sebuah bola spesial dengan properti khusus yang dapat membuat permainan lebih bervariasi.

1. Bola Spesial

Secara spesifik, bagaimana cara kerja bola spesial yang tersebar dalam level kita? Bola Spesial tidak sama seperti bola biasa yang muncul setiap beberapa detik secara acak. Di lokasi bola spesial yang ditentukan, terdapat lingkaran sihir yang telah dibuat untuk memanggil bola – bola tersebut. Setelah bola spesial di lingkaran sihir diambil, maka setelah beberapa detik bola spesial baru akan dipanggil lagi. Bola – bola spesial yang dapat muncul adalah sebagai berikut :

1. Ice Piercer



Gambar 5.2

Ice Piercer

Ice Piercer adalah sebuah bongkahan es batu berbentuk ketupat tajam yang dapat menembus player – player, musuh ataupun teman. Bila Ice piercer menembus seseorang, maka di tembusan selanjutnya bila Ice Piercer mengenai musuh, maka skor akan naik secara signifikan ketimbang bola biasa.

1. Snow-a-rang



Gambar 5.3

Snow-a-rang

Snow-a-rang adalah sebuah bumerang salju yang bila mengenai sesuatu atau seseorang dapat terbang kembali ke arah lemparan. Perlu diketahui juga bila snow-a-rang yang terpantul perlu ditangkap agar bisa dipakai lagi.

1. Explod-o-Ball



Gambar 5.4

Explod-o-Ball

Explod-o-Ball adalah sebuah bola dengan sihir ledakan yang bila mengenai seseorang, dapat menempel pada mereka. Bila bola menempel, maka setelah beberapa detik maka bola akan meledak, mengenai setiap musuh dan teman di area ledakan, meski begitu, hanya musuh yang terkena efek ledakan ini yang menambah skor tim, teman yang kena hanya terkena stun sesaat.

1. Freezing Winter



Gambar 5.5

Freezing Winter

Freezing Winter adalah sebuah bola mengandung sihir dingin yang sangat kuat, melambatkan siapapun yang kena secara drastis selama beberapa detik. Bola ini bisa mempengaruhi teman maupun musuh, jadi berhati – hati dalam menggunakan bola ini.

1. Stone Auger



Gambar 5.6

Stone Auger

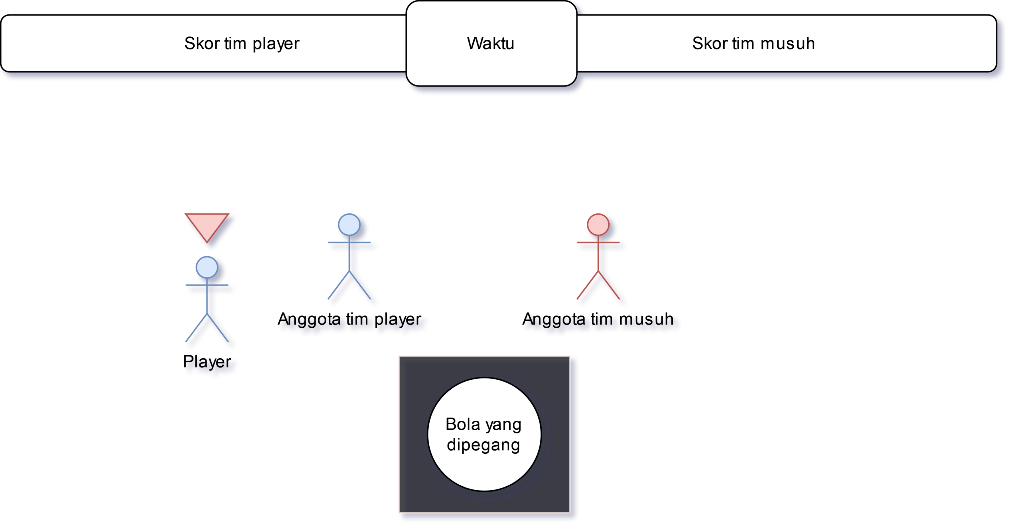
Dan bola terakhir diantara semua bola spesial ini adalah Stone Auger. Ini adalah sebuah bongkahan es dengan sihir kekerasan, membuatnya cukup kuat untuk mengebor dinding ataupun player. Perlu diketahui bila setelah mengebor menembus player atau dinding, maka Stone Auger akan pecah menjadi 3 bola kecil.

1. User Interface

Dengan gameplay ditentukan, maka berikutnya adalah untuk memikirkan UI yang akan dipakai saat permainan dijalankan. Ada beberapa informasi yang perlu ditunjukkan oleh UI, yaitu :

* Bola yang Dibawa Player
* Waktu yang tersisa dari permainan
* Skor yang dimiliki masing masing tim
* Karakter yang mana yang digunakan oleh player
* Karakter mana yang merupakan tim player dan musuh

Jadi dengan informasi yang ditentukan, berikut adalah perkiraan UI yang diinginkan :



Gambar 5.7

Design UI untuk permainan Splatted

Di atas akan ada waktu timer untuk menunjukkan waktu yang tersisa, selanjutnya adalah menunjukkan skor tim player dan musuh menggunakan bar di kiri dan kanan, dimana skor akan menjalar keluar. Selanjutnya untuk membedakan tim player dan musuh akan digunakkan kode warna agar bisa membedakan antara kedua tim, dan memberi penunjuk khusus seperti segitiga merah diatas karakternya player. Terakhir semua karakter bisa membawa bola di tangan, tapi untuk kemudahan, dibawah akan ada kotak yang menunjukkan bola milik player.

1. Desain Aksi Player / Bot

Desain selanjutnya adalah desain dari karakter yang akan dimainkan oleh player dan juga bot, kita akan mendesain apa yang dapat dilakukan oleh karakter tersebut, dan apakah aksi tersebut membantu dalam memperseru game Splatted. Untuk kemudahan bersama, karakter ini akan kita namakan Dodo karena saya tidak memiliki nama resmi untuknya



Gambar 5.8

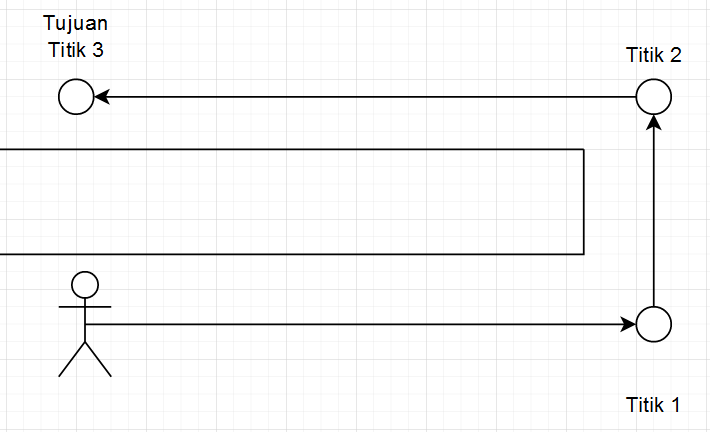
Dodo

Dalam subbab ini akan dipecah aspek – aspek dari Dodo yang akan didesain yang dapat berguna sebagai dasar permainan hingga digunakan untuk membuat permainan lebih menarik. Berikut adalah aspek – aspek dari Dodo yang perlu dipertimbangkan :

1. Movement / Gerakan

Kita akan mengutamakan melempar & menangkap dalam game Splatted, sehingga karena itu, Dodo tidak akan diberi kemampuan untuk melakukan dash atau gerakan yang spesial dalam Splatted. Tetapi sebagai kompensasi dari desain tersebut, Dodo akan diberi kecepatan bergerak yang cukup cepat untuk membantu dalam mengambil bola salju yang akan muncul di tanah.

**Untuk gerakan bot :** akan digunakan algoritma A\* untuk menemukan jalannya menuju sebuah lokasi, dimana A\* tersebut akan membuat titik – titik yang akan dilewati oleh bot untuk mencapai sebuah lokasi. Dengan ini kita hanya perlu memberi koordinasi tujuan dan Algoritma A\* akan mengurus bagian berjalannya.



Gambar 5.9

Ilustrasi cara kerja A\* untuk mencapai tujuan

1. Mengambil Bola

Untuk mengambil bola, player dan bot memiliki metode mengambil yang sama, dimana menggunakan sebuah manager bola salju, akan diambil gumpalan salju yang terdekat dari Dodo. Bila gumpalan tersebut cukup dekat, maka gumpalan tersebut akan dihapus dan Dodo akan mendapatkan sebuah bola biasa. Hal ini juga berlaku untuk lokasi bola spesial dimana bola spesial bisa hilang sesaat bila diambil.

**Untuk gerakan bot :** bot hanya akan mengambil gumpalan salju bila bot “melihat” bola tersebut. Setelah melihat gumpalan itu, bot akan menggunakan algoritma A\* untuk meraih dan bila sudah cukup dekat, mengambil gumpalan yang dilihat tersebut.

1. Menangkap Bola



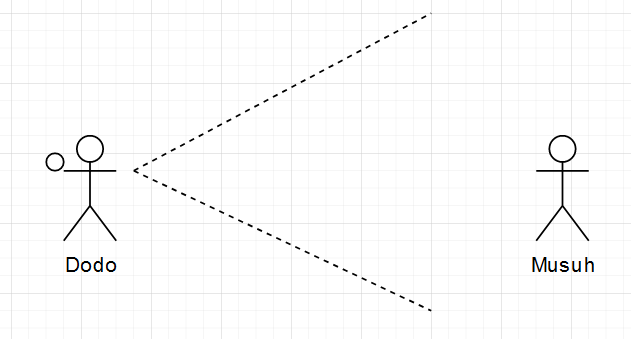
Untuk menangkap bola, Dodo memiliki kemampuan untuk menangkap sebuah bola. Dimana bila Dodo masuk state menangkap bola dan sebuah bola mengenai Dodo, bola tersebut akan masuk ke tangan Dodo. Bola yang ditangkap akan dilempar duluan sebelum bola yang diambil dari tanah, dan bila Dodo menangkap bola lain saat sedang memegang sebuah bola tangkapan maka bola tangkapan tersebut akan jatuh dan Dodo akan menyimpan bola tangkapan terbaru.

**Untuk bot :** bot juga dapat melakukan tangkap bola, tetapi ada 3 syarat penting yang harus dipenuhi agar bot bisa menangkap bola. Pertama, bot harus dapat melihat sebuah bola yang dilempar. Kedua, bot yakin bila bola bisa mengenai bot tersebut. Dan terakhir, pengecekan kemungkinan apakah bot melakukan aksi tangkap bola diraih. Bila ketiga syarat tersebut diraih, maka bot akan melakukan tangkap bola

1. Melempar Bola

Untuk melempar bola, Dodo memerlukan sebuah bola entah daru gumpalan salju di tanah atau dari menangkap bola lemparan Dodo lainnya. Setelah pengecekan, maka akan dilakukan fase melempar Bola, player dan bot memiliki aksi yang mirip, cuma dengan tahap yang sedikit berbeda dengan satu sama lain ;

**Untuk lemparan player :** player harus menahan klik kiri dulu. Melakukan ini akan membuat 2 garis titik – titik yang menunjukkan akurasi lemparan muncul dari Dodo menuju arah mouse, sesuai dengan inspirasi menembak dari game Crosscode. Bila 2 titik ini ditahan, maka kedua garis ini akan saling mendekat dan akan menjadi 1. Bila garis sudah menjadi 1 dan player melepas klik kiri, maka Dodo akan melempar bola dengan lurus. Tapi apabila player melepas klik kiri saat 2 garis tersebut belum menyatu, maka lemparan Dodo tidak akan akurat.



Gambar 5.10

Ilustrasi Dodo membidik seseorang

**Untuk lemparan bot :** sebelum melempar bola, bot harus melihat Dodo dari tim lawan terlebih dahulu. Bila kelihatan, maka bot akan berhenti untuk membidik bolanya. Setelah beberapa detik, bot akan melempar bola terhadap Dodo lawan apabila Dodo tersebut tidak dihadang oleh sebuah dinding. Bila Dodo lawan dihadang dinding maka bot akan mengejar Dodo tersebut hingga bisa kelihatan.

1. Kena Lempar Bola

Saat kena lempar bola, Dodo akan terkena stun untuk waktu yang singkat, dan saat terkena stun, Dodo masih bisa kena lempar bola lagi, dimana bila ada bola yang mengenai Dodo tetap akan termasuk dalam skor. Ini berguna untuk memberi beban terhadap lemparan yang diberikan dan untuk menghukum player yang terkena bola untuk diam selama beberapa detik, rentan terhadap tembakan lain.

1. Fakeout (Khusus Player)

Fakeout adalah aksi untuk menarik seseorang menuju pilihan yang salah atau keadaan yang tidak menguntungkan menggunakan tipuan. Dalam game Splatted, aksi Fakeout ini berupa melempar bola tanpa menggunakan sebuah bola, sehingga lawan akan bingung saat berusaha menangkap bola yang tidak pernah dilempar. Jadi bila player melakukan fakeout, maka musuh yang melihatnya akan berusaha menangkap bola player, memberi kesempatan untuk player melempar bola dan kena dengan memberi Dodo waktu jeda sebelum bisa menangkap lagi.

1. Tutorial

Tutorial digunakan seluruh game untuk mengajari dasar – dasar mekanin dari sebuah game, dimana dasar – dasar tersebut akan dipakai dan diperdalam dalam keseluruhan game. Dan karena itu tutorial merupakan sebuah aspek yang penting dalam sebuah game bila game tersebut memiliki mekanik dan kontrol yang unik, dimana biasanya begitu kasusnya.

Untuk Tutorial Splatted akan digunakan metode yang sederhana, dimana akan disediakan beberapa ruangan yang akan mengajari sebuah konsep / aksi, lalu mendesain ruangan itu berdasarkan aksi tersebut. Jadi di dalam Splatted akan ada 5 aksi yang perlu diajari dan sebagai akibatnya 5 ruangan yang perlu dibuat berdasarkan aksi – aksi tersebut, yaitu ;

* 1. Gerakan, dimana meskipun gerakan yang diaplikasikan sudah merupakan standar gerakan dari semua game serupa (WASD), lebih baik diajari sekilas agar para pemain awam tidak kebingungan.
  2. Mengambil dan melempar bola ; ini merupakan dasar dari game Splatted, dan bukanlah sebuah aksi yang diaplikasikan di banyak game, sehingga mengajari player untuk mengambil dan melempar bola merupakan sebuah kewajiban dari Splatted.
  3. Menangkap bola ; Satu dari dua pilihan untuk menghindari musuh mendapat poin dari player, pilihan lainnya adalah untuk berlari menghindari bola dari arah lemparan. Sangat diperlukan untuk visi dari game ini yang berpusat pada melempar dan menangkap.
  4. Fakeout; dimana game berpusat pada aksi lempar menangkap, fakeout adalah variabel tak tentu milik player yang bisa membawa kemenangan bila dipakai dengan benar oleh player.
  5. Bola Spesial ; dipakai sebagai demonstrasi tampilan dari bola spesial, mekaniknya dan untuk menguji coba efek dari setiap bola yang ada agar player tidak kebingungan di tengah permainan

1. Map Generation

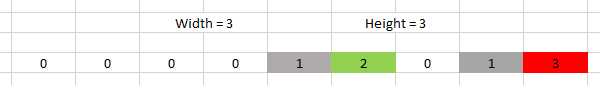
Aspek terakhir yang perlu dipertimbangkan dalam game Splatted dan yang bisa diperdebatkan paling penting adalah Map Generation-nya, mengingat bila judul tugas akhir yang tertera adalah “Procedural Map Generation untuk game Splatted”. Dikarenakan itu membuat map generation tidak bisa semena – mena menggunakan random number generator, tapi menggunakan algoritma – algoritma khusus yang merupakan random number generator tapi lebih berkelas.

Algoritma yang disinggung diatas adalah Genetic Algorithm, hati dan jiwa dari kualitas level dan nilai tugas akhir ini. Seperti yang telah dijelaskan pada bab sebelumnya, konsep yang digunakan untuk menentukan apa yang baik atau buruk pada sebuah Genetic Algorithm adalah fitnessnya, dan dari situ dibuat sebuah level berdasarkan fitness tersebut. Dalam Subbab ini akan dijelaskan mengenai tahap – tahap dari Genetic Algorithm, dimulai dengan representasi dari sebuah Map.

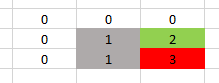
1. Jenis Generasi Map & Representasi

Di dalam Splatted akan dibuat 2 jenis Representasi didasarkan oleh metode generasi Map yang tersedia, dimana kedua representasi dapat dipakai untuk mencapai hasil yang berbeda dalam membuat sebuah Map. Berikut ini adalah 2 Representasi yang akan digunakan pada Splatted :

1. Tile Generation



**(a)**



**(b)**

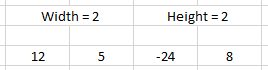
Gambar 5.11

Contoh Representasi Tile berbentuk (a) 1 Dimensi (b) dan 2 Dimensi

Metode pertama yang akan dipakai adalah Tile Generation. Dalam Tile Generation, setiap tile akan diisi dengan 4 nilai berbeda. Nilai tersebut secara urut adalah 0 yaitu tile kosong, 1 yang berisi sebuah batu, 2 yang memiliki sebuah spawner bola khusus dan terakhir adalah 3 yang merupakan lokasi awal penempatan player dalam game.

Dari Tile Generation akan dilakukan representasi menggunakan array 1 dimensi seperti pada gambar 5.11a diatas. Akan digunakan sebuah array 1 dimensi sepanjang luas dari Map, dalam contoh ini luas 9 tile bersama dengan variabel simpanan yang menyimpan panjang dan lebar dari Map, dimana variabel – variabel tersebut akan dipakai untuk mengubah array 1 dimensi menjadi sebuah array 2 dimensi yang merupakan isi dari sebuah Map.

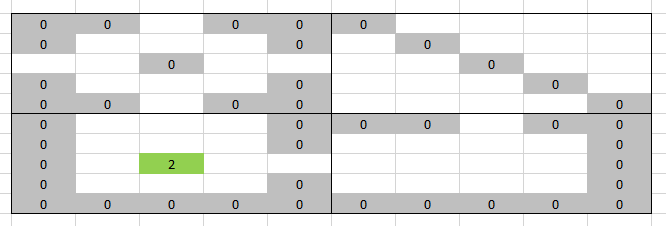
1. Template Generation

****

**(a)**

****

**(b)**

****

**(c)**

Gambar 5.12

Contoh Representasi Template (a) 1 Dimensi, (b) 2 Dimensi dan (c) hasil akhir

Metode kedua yang tersedia adalah Template Generation. Template Generation sama seperti Tile Generation dimana Template Generation menggunakan array 1 dimensi bersama dengan variabel panjang dan lebar dari sebuah Map. Tetapi daripada menyimpan isi tile-tile individual, setiap Gene menyimpan ID dari sebuah template, dimana setiap template memiliki ukuran 5X5 Tile. Oleh sebab itu sesuai dengan gambar 5.12a, array 1 dimensi lebih pendek dibandingkan dengan array di gambar 5.11a, dimana variabel panjang dan lebar juga berubah sesuai dengan bentuk levelnya.

Setelah itu seperti Tile Generation, array 1 dimensi dibentuk menjadi array 2 dimensi seperti gambar 5.12b. Setelah membuat array 2 dimensi, Gene-Gene template id yang ada diubah dengan isi dari template id tersbut, dengan hasil seperti pada gambar 5.12c, siap untuk dipakai sebagai sebuah Map. Bila id yang diterima bernilai negatif, maka tengah dari template bisa memiliki sebuah spawner bola spesial bila tidak ada batu yang menghalanginya, dan terakhir di template tidak ada penempatan player, player akan selalu ditempatkan di tempat yang sama setelah map dibuat.

1. Fitness Level

Setelah menentukan Metode Generasi yang telah dibuat, bagian selanjutnya yang perlu disediakan adalah Fitness-Fitness yang menentukan

1. Jumlah batu dalam Level (Tile Generation)

(5.1)

m = Nilai minus

R = Jumlah batu dalam sebuah level

MinR = Minimal batu yang ditentukan

MaxR = Maximal batu yang ditentukan

(5.2)

M = Nilai minus maximal

A = Luas dari sebuah Level

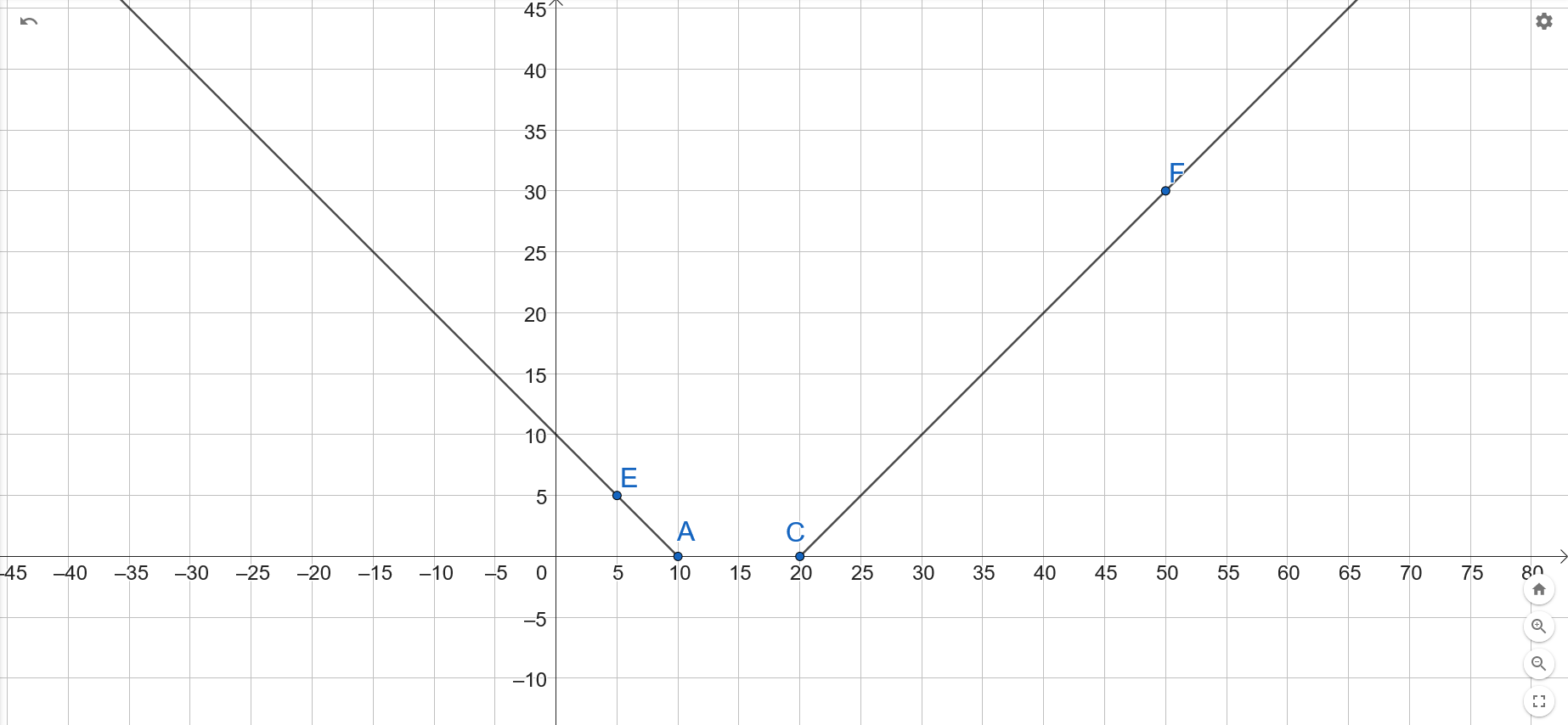
(5.3)

F = Nilai Fitness

W = Weight yang ditetapkan

Untuk fitness ini, terdapat 3 tahap untuk mencari fitness sesuai dengan rumus diatas. Rumus pertama dan kedua digunakan untuk mencari Nilai Minus dan Nilau Minus Max untuk dinormalisasikan dan digunakan untuk mengurang nilai Fitness. Terakhir setelah dikurangi nilai fitness akan dikuadrat untuk memperbesar pengaruh ketidaksempurnaan dari fitness tersebut, dan dikali dengan Weight yang ditentukan, maka Fitness telah jadi.

Sekarang untuk penjelasan mengenai rumus diatas, dimulai dari rumus 5.1, m atau nilai minus adalah selisih antara batu yang ada di level dan salah satu dari 2 batas batu terdekat yang telah ditetapkan, yaitu MinR dan MaxR, semisal bila jumlah batu adalah 5 dan MinR = 10 dan MaxR = 20, maka m akan bernilai 5 karena selisih antara 5 dan MinR, batas batu terdekat. Tetapi bila jumlah batu 50, maka m akan bernilai 30, dikarenakan selisih antara jumlah batu dan batas terdekat yaitu MaxR (20), sementara bila jumlah batu diantara 2 batas maka m = 0.

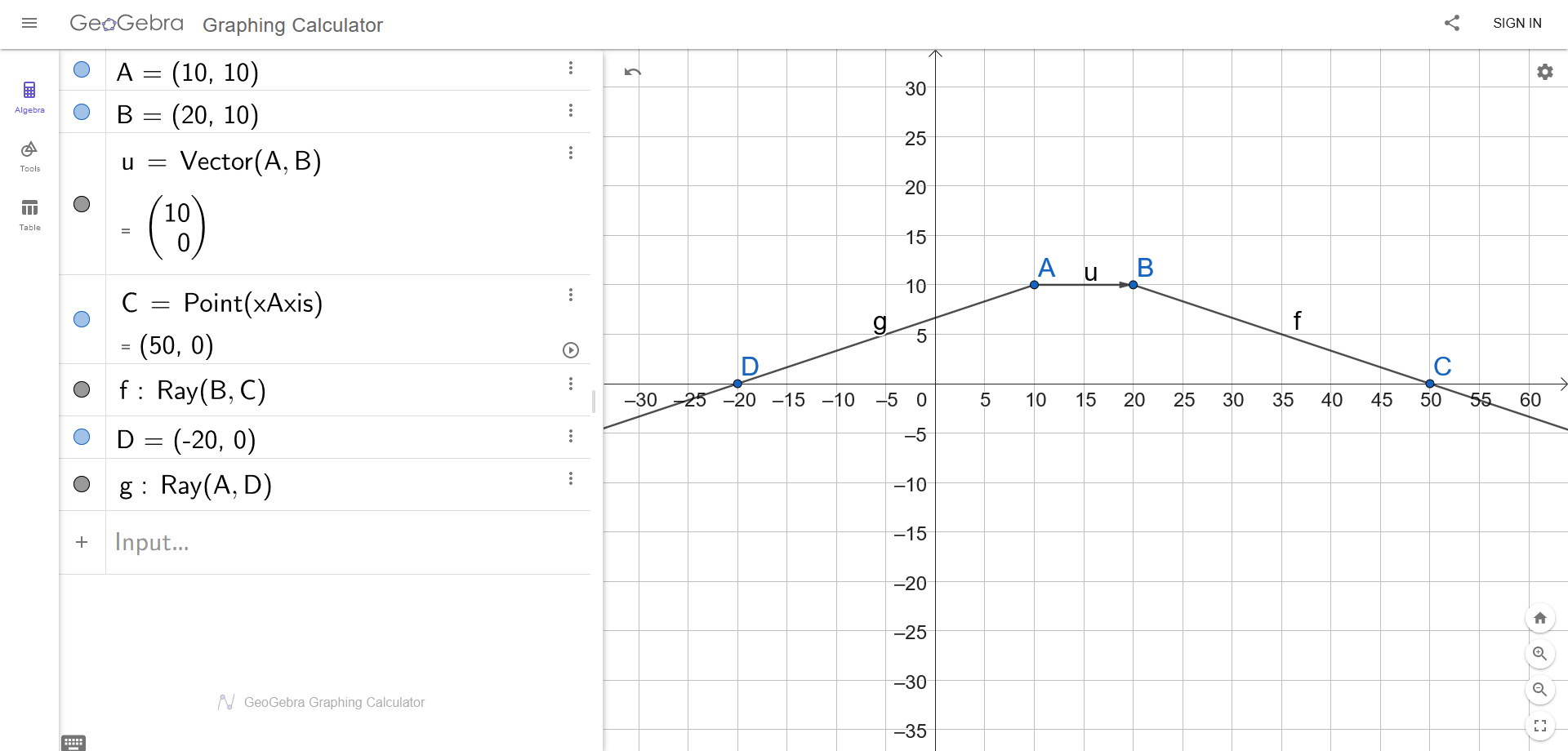


Gambar 5.13

Hasil nilai m fungsi 5.1

Di gambar 5.13 diatas, terdapat grafik fungsi 5.1 menggunakan variabel MinR 10 dan MaxR 10 dan sumbu x merepresntasikan jumlah batu. Di grafik ini E dan F merupakan contoh jumlah batu sebelumnya dimana bila dalam level ada 5 batu, maka nilai m akan 5, sementara bila batu berjumlah 50, maka m akan bernilai 30.

Selanjutnya adalah variabel M, yang merupakan nilai minus maximal, atau nilai minus tertinggi yang dapat diraih dalam sebuah pembuatan level. Metode menentukan nilai M ada diantara 2 selisih yang terbesar, yaitu diantara Luas area – MaxR atau MinR – 0, bisa juga digantikan dengan MinR saja. Kedua nilai tersebut dibandingkan, dan pilihan dengan nilai yang lebih besar akan dipakai sebagai nilai M.



Gambar 5.14

Grafik hasil fitness contoh yang diberi sebelum dikuadrat

Gambar 5.14 diatas adalah hasil grafik dari rumus 5.3 dimana y adalah fitnessnya (seharusnya 1 tapi dikali 10 agar bisa dilihat) dan x adalah jumlah batunya. Di contoh diatas diberi contoh level dengan ukuran 50. Dari 10 – 20 fitness bernilai 1, tapi selain itu nilai fitness akan menurun hingga 0. Nilai fitness tidak bisa dibawah 0 dikarenakan jumlah batu tidak bisa dibawah 0, dan nilai minus dinormalisasi terhadap ukuran level (50) sehingga bila level memiliki 50 batu fitness pasti 0.

Dengan m dan M ditemukan, Langkah terakhir adalam menggunakan M untuk menormalisasikan m, dan menggunakan m yang dinormalisasikan untuk mengurangi Fitness ini yang memiliki nilai 1. Dengan itu, rumus terakhir adalah fungsi 5.3, dimana kuadrat tersebut dipakai untuk memperbesar ketidaksempurnaan fitness dan weight dipakai untuk mengatur pengaruh fitness terhadap fitness lain. Bila semua berjalan lancar, maka fungsi akan memiliki hasil seperti gambar 5.14 saat sebelum dikuadrat dan dikali weight.

1. Ukuran kelompok batu (Tile Generation)

(5.4)

m = Nilai minus

R = Jumlah batu yang berkumpul bersama dalam 1 kelompok

MinR = Minimal batu yang ditentukan

MaxR = Maximal batu yang ditentukan

(5.5)

M = Nilai minus maximal

A = Luas dari sebuah Level

(5.6)

F = Nilai Fitness

W = Weight yang ditetapkan

n = Jumlah kelompok batu dalam sebuah level

Fitness ini berguna untuk menentukan ukuran dari bongkahan – bongkahan batu yang ada di dalam level. Fungsi 5.4 dan 5.5 memiliki fungsi yang sama dengan fungsi 5.1 dan 5.2 karena prinsipnya sama, ditentukan range jumlah batu yang diperboleh dan bila melewati range tersebut maka fitness akan berkurang. Tetapi yang berbeda adalah Fungsi 5.6, dimana fungsi tersebut menghitung fitness dari setiap kelompok batu, dan merata-rata fitness yang didapat.

1. Luas area yang bisa diakses

(5.7)

F = Nilai Fitness

aterbesar = Area terbesar yang bisa diakses

atotal = Area total dalam sebuah level

W = Weight yang ditetapkan

Fitness ini memberitahu Genetic Algorithm bila kita menginginkan level dimana seluruh area (tile yang bukan batu) dapat diakses. Fitness ini bekerja dengan mengambil area terluas yang ada di dalam map, lalu dibagi dengan semua area yang terdapat di dalam map, entah bisa diakses (didatangi oleh player/bot) atau tidak. Lalu seperti Fitness diatas dikuadrat dan dikali weight untuk Fitness bagus dan pengaturan dampak fitness.

1. Jumlah bola spesial yang bisa diambil

(5.8)

F = Nilai Fitness

pakses = Bola spesial yang bisa diakses

ptotal = Bola spesial dalam sebuah level

W = Weight yang ditetapkan

Fitness ini serupa dengan Fitness Luas area memberitahu Genetic Algorithm bila kita ingin seluruh bola spesial level bisa diambil oleh bot atau player. Dalam rumus 5.8, akan dicari jumlah dari bola spesial yang bisa diambil, dan jumlah tersebut akan dibagi dengan jumlah semua bola spesial dalam level. Setelah nilai tersebut ditemukan, dikuadrat dan di kali weight dan fitness selesai.

1. Rasio antara bola spesial dan luas area level

(5.9)

m = Nilai minus

P = Jumlah bola spesial dalam sebuah level

MinP = Minimal batu yang ditentukan

MaxP = Maximal batu yang ditentukan

(5.10)

M = Nilai minus maximal

A = Luas dari sebuah Level

(5.11)

F = Nilai Fitness

W = Weight yang ditetapkan

Fitness ini menggunakan rumus yang persis dengan Fitness jumlah batu, dengan perbedaan utama daripada menghitung jumlah batu, Fitness ini menghitung jumlah bola spesial yang ada di dalam map. Sama seperti Fitness jumlah batu, semakin jauh jumlah bola spesial dari range, semakin buruk fitnessnya.

1. Variasi template dalam level (Template Generation)

(5.12)

(5.13)

F = Nilai Fitness

t = Jumlah Template unik yang dipakai

Ti = Jumlah template ke – i yang sama

tl = Toleransi jumlah template yang sama

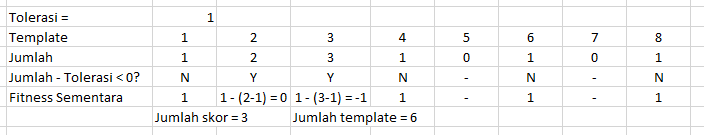
Untuk fitness terakhir yang hanya digunakan oleh Template Generation, ini digunakan untuk mengurangi template yang kembar di dalam level. Untuk menemukan nilai dari fitness ini, semua template yang muncul lebih dari 1 kali akan dihitung jumlahnya, semisal ada level dengan template : 1 3 4 2 2 3 8 6 3, maka akan didapat jumlah seperti gambar berikut :



Gambar 5.15

Hasil perhitungan template contoh

Setelah mendapat jumlah tersebut, maka setiap template yang lebih dari satu akan dihitung menggunakan rumus 5.13. Dimana Ti adalah jumlah dari masing masing template, dan tl adalah tolerasi yang kita tentukan, dimana semakin tinggi tolerasi yang ditentukan, semakin banyak jumlah template yang bisa kembar. Setelah menentukan toleransi, maka berikutnya adalah menentukan fitness sementara setiap Template berdasarkan rumus 5.13 :



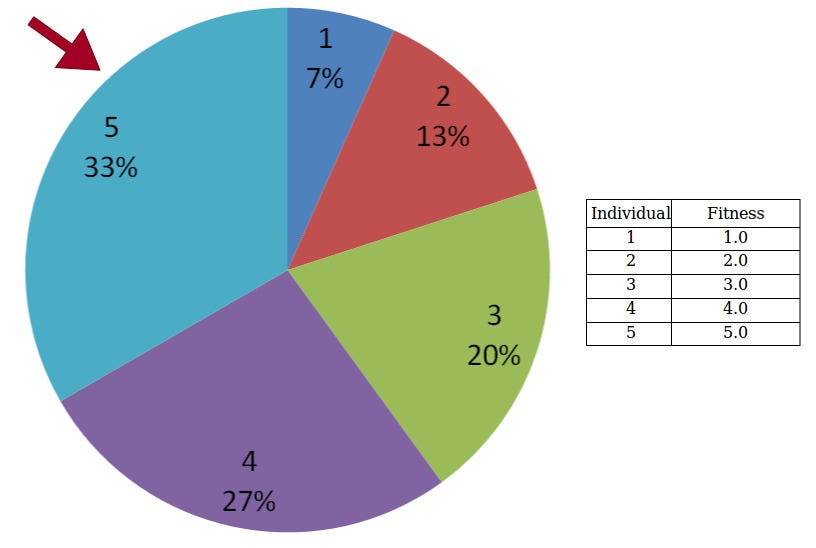
Gambar 5.16

Hasil perhitungan template contoh dengan toleransi

Diambil dari gambar 5.16 diatas, bila jumlah – toleransi < 0, maka fitness sementara template tersebut akan menjadi 1, sementara bila tidak, maka fitness sementara bisa 0 ataupun dibawah 0. Nanti jumlah dari fitness sementara akan di rata – rata dengan jumlah template yang termasuk dalam perhitungan, dan dari situ akan didapat fitness final kita, tinggal di kuadrat dan dikali weight dan fitness pun jadi, tetapi bila hasil jumlah fitness bernilai negatif fitness final kita bernilai 0.

Setelah menentukan Fitness, kita juga harus menentukan metode operasi – operasi kita yang lain, diantaranya : Selection, Crossover, Mutation dan terakhir kondisi berhenti. Berikut adalah pilihan untuk operasi – operasi yang dibutuhkan serta alasannya dipilih :

1. Selection



Gambar 5.17

Visualisasi Roulette Wheel Selection

Selection yang digunakan untuk algoritma genetik Splatted adalah Roulette Wheel Selection yang telah dijelaskan pada bab II Genetic Algorithm, tetapi sebagai pengingat, Roulette Wheel adalah sebuah seleksi yang memilih Kromosom berdasarkan fitness dari Kromosom tersebut, dimana semakin tinggi Kromosomnya, semakin tinggi kemungkinan Kromosom tersebut dipilih, seperti yang dapat dilihat pada gambar 5.17 diatas.

Pada gambar 5.17, Individual 5 dengan fitness 5 memiliki kemungkinan terbesar dipilih dikarenakan fitnessnya yang tinggi, prinsip yang sama berlaku pada Individual 1 yang memiliki kemungkinan kecil karena Fitnessnya yang kecil. Dengan metode seleksi ini Parent bisa bervariasi dengan mengorbankan sedikit kualitas rata rata dari tiap populasi.

1. Crossover



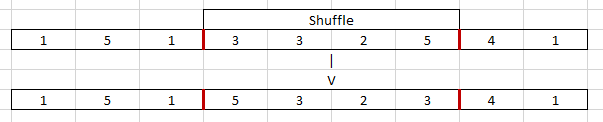
Gambar 5.18

Visualisasi Uniform Crossover

Gambar 5.18 diatas menunjukkan 2 Kromosom dimana isi Gene dalam Kromosom tertukar secara acak, karena Gene yang tertukar memang acak. Dalam Uniform Crossover, dimulai dari Gene paling kiri (atau kanan, tapi untuk penjelasan ini kiri saja), lempar sebuah koin, bila koin mendarat di kepala, maka Gene kedua parent ditukar, sementara bila mendarat di ekor, maka Gene tidak ditukar. Setelah proses tersebut, pindah ke Gene selanjutnya, dan proses ini diulang hingga Seluruh Gene dilalui.

Crossover ini berguna karena ini hanya menukar Gene secara individual dan acak, dimana setiap kromosom merupakan representasi Map yang aslinya sebuah Array 2D. Bila Gene ditukar secara bongkahan bisa membuat hasil Genetic Algorithm keluar lebih lama.

1. Mutation



Gambar 5.19

Visualisasi Partial Shuffle Mutation

Mutation ini adalah mutation yang sederhana, dimana dipilih 2 titik secara acak dalam sebuah array, lalu Gene – Gene yang berada diantara kedua titik tersebut akan diacak urutannya. Dengan ini, Partial shuffle telah dilakukan.

Sebelum ini sebenarnya digunakan Uniform Mutation, dimana setiap Gene memiliki kemungkinan untuk diacak isinya, tetapi karena ini digunakan dalam sebuah game, optimisasi dan kecepatan sama – sama penting, dan Uniform Mutation memakan waktu yang lama, sehingga pada akhirnya Mutation diganti menjadi partial shuffle, dikarenakan partial shuffle mutation hanya mengacak array yang ada, yang berarti setiap pemain tidak akan mencapai 1 level yang terbaik, menambah variasi dan mutasi ini tidak memakan waktu yang lama.

1. Stop Condition

Untuk stop condition kita, akan digunakan Fitness Stagnation, dimana bila Fitness kita tidak naik selama beberapa generasi, maka Genetic Algoritm akan dihentikan. Ini akan memastikan bila Genetic Algorithm akan berusaha semaksimal mungkin untuk mencari kromosom terbaik sebelum menghentikan proses GA ini.

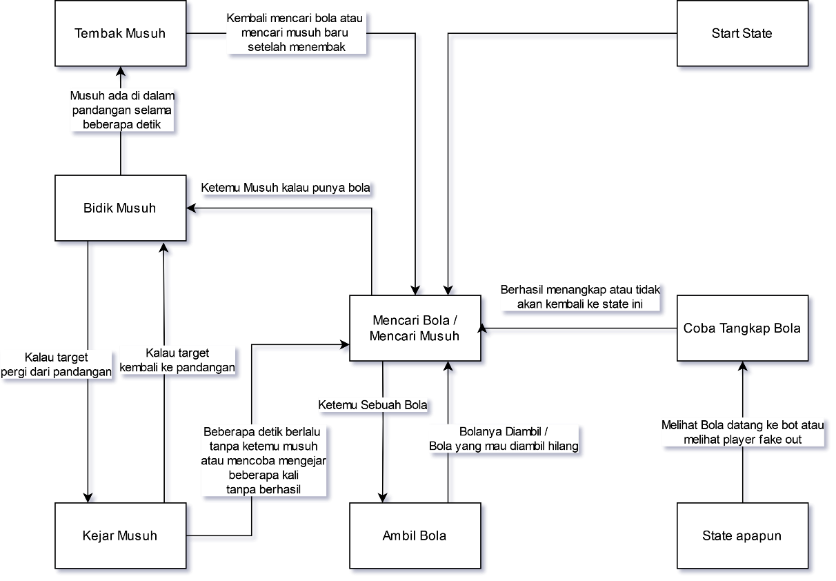
Meski kedengarannya memakan waktu yang lama, berusaha mencari fitness terbaik, berkat bentuk dari Crossover dan Mutation maka proses ini tidak selama yang dibayangkan. Ini karena setelah Genetic Algorithm telah diberi populasi inisial, maka hanya populasi tersebut yang dapat dipakai, sehingga mengurangi waktu yang diperlukan untuk mencapai fitness maksimal.

1. Variabel-Variabel pelengkap algoritma genetik

Dalam subbab ini akan disinggung mengenai nilai dari variabel-variabel pendukung yang akan digunakan pada algoritma genetik. Pertama adalah jumlah Chromosome di dalam sebuah populasi. Ditentukan dalam sebuah populasi terdapat 50 hingga 100 Kromosom. Ditentukan sebagai berikut karena variasi dari map hanya memerlukan sebanyak itu, dimana bila Chromosome terlalu sedikit maka tidak akan ada banyak variasi karena pilihan orang tua sedikit dan bila Chromosome terlalu banyak maka algoritma genetik akan berjalan untuk waktu yang jauh lebih lama dikarenakan seluruh tahap algoritma genetik selain Crossover dan Mutation akan memakan waktu yang lebih lama.

Variabel kedua dan terakhir adalah Stop Condition. Dimana untuk stop condition untuk Splatted adalah Fitness Stagnation sebanyak 100 Generasi. Artinya adalah bila Fitness terbaik yang bisa diraih oleh sebuah algoritma genetik tidak berkembang banyak selama 100 Generasi, maka algoritma genetik akan dihentikan untuk menghemat waktu dan mendapat Chromosome terbaik yang bisa didapat.

1. Bot State Machine



Gambar 5.20

State Machine Diagram Bot

Bot merupakan inti dari banyak game single player, karena dimana player lain tidak tersedia, kita membuat lawan kita sendiri sebagai tantangan terhadap player kita. Tentu saja kita tidak bisa menyuruh bot kita untuk melakukan aksi sembarangan, bot memiliki alur perilaku mereka sendiri dan karena itu bot bisa terlihat seperti memiliki kepintaran tersendiri. State Machine di gambar 5.20 diatas adalah blueprint State Machine diagram bagi bot kita di dalam Splatted. Berikut adalah state – state yang dijalankan secara urut :

1. Start State

Start state adalah state yang paling pertama dijalankan oleh State Machine. Dari state ini kita bisa bergerak ke banyak state, entah melempar bola, menangkap, atau mencari bola. Tapi dikarenakan bot kita masih tidak memiliki apa apa, kita akan pindahkan bot ke state Mencari Bola agar bisa memiliki kesempatan bermain dalam Splat.

1. Mencari Bola

Dalam state ini, bot akan mencari sebuah bola melalui beberapa tahap. Tahap pertama, pilih sebuah titik acak dalam map yang bisa diakses, bila bot dapat melihat sebuah bola selagi berjalan, maka pindah ke state ambil bola, tetapi bila tidak menemukan bola dan bot sudah sampai tujuan, maka bot memilih lokasi baru dan mengulang state ini dari awal.

1. Ambil Bola

Setelah menemukan sebuah bola, maka aksi berikutnya adalah untuk mengambil bola tersebut. Bot akan berjalan menuju bola dan bila bot tersebut sampai di tujuan dan bola tidak hilang, maka bot akan mengambil bola tersebut dan pindah menuju state Mencari musuh, tetapi bila bot sudah sampai dan bola telah diambil, maka bot akan kembali menuju state mencari bola.

1. Mencari Musuh

State ini menggunakan state yang sama dengan mencari bola, tetapi dengan perbedaan ketimbang mencari bola, bot akan mencari anggota dari tim musuh. Bila bot menemukan anggota tim musuh, maka bot akan berpindah state menuju bidik musuh. Bila tidak menemukan musuh, maka state akan diulangi hingga menemukan musuh

1. Bidik Musuh

Bot “membidik” bola kepada musuh selama waktu yang ditentukan, dan melempar bola menuju mereka. Bila bot terlempar, maka bola akan terlempar, dan bot akan kembali ke state mencari bola. Tetapi bila musuh menghilang dari pandangan musuh, maka bot akan pindah ke state kejar musuh untuk mengejar musuh yang kabur tersebut.

1. Kejar Musuh

Bot mengejar musuh yang kabur dari pandangan bot. Menggunakan A\*, bot akan mengikuti musuh hingga musuh bisa terlihat lagi, mengembalikan bot ke state bidik musuh atau bila musuh tidak kelihatan selama beberapa detik, maka bot akan menyerah dan kembali ke state mencari musuh untuk mencari musuh bari. Bot juga akan menyerah bila bot menemukan dan kehilangan musuh yang sama beberapa kali berturut turut, dimana bot akan mencari musuh baru.

1. Coba tangkap bola

State ini dapat diakses dari state apapun dan dapat diraih bila bot melihat sebuah bola terbang ke arahnya. Bot akan melakukan lemparan dadu, dan bila angkanya cukup tinggi maka bot akan menangkap bola tersebut. Tetapi bila bot tidak beruntung, maka bot tidak akan menangkap bola yang dilihat. Setelah entah usaha tersebut, maka bot akan kembali ke mencari musuh / bola, tergantung apabila bot punya bola salju atau tidak. State ini juga dapat diraih apabila bot melihat player melakukan fakeout, dimana bot akan mencoba menangkap bola yang tidak keluar.