LAPORAN TUGAS BESAR IF2110

Algoritma dan Struktur Data

Avatar World War: a True Dictator

Dipersiapkan oleh Kelompok 1

Muhammad Zunan Alfikri 13518019

Brandon Oktavian Pardede 13518043

Felicia Gillian Tekad Huerah 13518070

Muhammad Ayyub Abdurrahman 13518076

Muhammad Ridwan Fauzi 13518124

Hanif Muhammad Gana 13518127

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Sekolah Teknik Elektro dan Informatika ITB** | Nomor Dokumen | | Halaman |
| *IF2110-TB-1-1* | | *20* |
| *Revisi* | *1* | *19 Nov 2019* |

Daftar Isi

[1 Ringkasan 3](#_Toc244931066)

[2 Penjelasan Tambahan Spesifikasi Tugas 3](#_Toc244931067)

[2.1 <Spesifikasi Fitur Tambahan 1> 3](#_Toc244931068)

[2.2 <Spesifikasi Fitur Tambahan 2> 3](#_Toc244931069)

[3 Struktur Data 3](#_Toc244931070)

[3.1 <Struktur Data 1> 3](#_Toc244931071)

[3.2 <Struktur Data 2> 3](#_Toc244931072)

[4 Program Utama 3](#_Toc244931073)

[5 Algoritma-Algoritma Menarik 4](#_Toc244931074)

[5.1 <Algoritma 1> 4](#_Toc244931075)

[5.2 <Algoritma 2> 4](#_Toc244931076)

[6 Data Test 4](#_Toc244931077)

[6.1 <Data Test 1> 4](#_Toc244931078)

[6.2 <Data Test 2> 4](#_Toc244931079)

[7 Test Script 4](#_Toc244931080)

[8 Pembagian Kerja dalam Kelompok 5](#_Toc244931081)

[9 Lampiran 5](#_Toc244931082)

[9.1 Deskripsi Tugas Besar 2 5](#_Toc244931083)

[9.2 Notulen Rapat 5](#_Toc244931084)

[9.3 Log Activity Anggota Kelompok 5](#_Toc244931085)

[9.4 <Lain-Lain> 5](#_Toc244931086)

# Ringkasan

Avatar World War: a True Dictator adalah game turn-based strategy yang mensimulasikan perang antar pemain. Permainan ini terdiri dari dua orang player yang bermain bergantian. Di awal game, pemain dapat memilih untuk memulai permainan baru atau load game yang sudah disimpan. Konfigurasi awal permainan dibaca dari file eksternal dengan format yang telah ditentukan (dapat dilihat pada lampiran). Permainan berakhir apabila salah satu pemain tidak memiliki bangunan lagi atau perintah exit digunakan.

Ada beberapa atribut di dalam permainan, seperti Peta dan Bangunan. Peta merupakan ilustrasi dari posisi geografis bangunan yang ada di dunia permainan ini. Peta dapat berukuran 10x10 hingga 20x20. Bangunan milik suatu player pada peta ditandai dengan warna dari player tersebut. Peta ditampilkan jika player memasukkan command ‘STATUS’ atau berada di awal giliran suatu player. Bangunan merupakan atribut utama di dalam game ini. Bangunan bisa dimiliki oleh suatu player ataupun tidak dimiliki player manapun. Untuk memiliki suatu bangunan, pemain harus mengalahkan sejumlah pasukan yang ada dalam bangunan tersebut. Tiap bangunan memiliki sejumlah pasukan yang jumlahnya tidak melebihi 1 juta. Apabila telah dimiliki, pasukan didalam bangunan akan bertambah sejumlah tertentu (A) setiap perubahan giliran. Apabila telah mencapai sejumlah pasukan tertentu(M), penambahan tidak akan dilakukan. Pasukan dalam bangunan dapat menyerang pasukan pada bangunan lain yang terhubung dengan bangunan tersebut. Pasukan yang ada didalam bangunan dapat dipindahkan ke bangunan lain. Ada 4 jenis bangunan, yakni Castle,Tower, Fort, dan Village. Tiap jenis bangunan dibedakan oleh jumlah pasukan awal((U), nilai perubahan pasukan(A), kapasitas bangunan (M) dan keberadaan pertahanan(P) (detailnya dapat dilihat pada lampiran) Bangunan memiliki 4 level yang dapat dinaikkan levelnya dengan menukarkan sejumlah pasukan yang ada didalam bangunan tersebut.

Di dalam permainan, pemain dapat memasukkan command yang dapat diberikan oleh pengguna. Pengguna dapat memasukkan 8 command, yakni Level\_Up, Undo, End\_Turn, Save, Move, Exit, Skill dan Attack. Level\_Up digunakan untuk menaikkan level dari suatu bangunan. Level up dapat digunakan apabila mencapai jumlah M/2. Apabila berhasil dinaikkan, maka pasukan sejumlah M/2 akan berkurang. Undo digunakan untuk membatalkan masukan sebelumnya dan hanya dapat digunakan sebelum giliran berakhir atau pemain menggunakan skill. End\_Turn digunakan untuk mengakhiri giliran dari suatu pemain. Save digunakan untuk menyimpan permainan ke file eksternal. Move digunakan untuk memindahkan pasukan ke bangunan lain.Move hanya dapat dilakukan sekali pada tiap bangunan pada gilirannya dan Exit digunakan untuk mengakhiri permainan. Diasumsikan exit dapat digunakan walaupun tidak melakukan Save terlebih dahulu. Sedangkan komando Skill digunakan untuk mengaktifkan skill.

Skill merupakan komando yang dapat mengubah hal-hal tertentu. Ada 7 Skill yang dapat digunaka, yakni Instant Upgrade, Shield, Extra Turn, Attack Up, Critical Hit, Instant Reinforcement, dan Barrage. Instant Upgrade didapatkan diawal permainan dan pemain yang menggunakan skill ini seluruh bangunannya akan naik satu level. Shield didapatkan jika setelah sebuah lawan menyerang, bangunan pemain berkurang 1 menjadi sisa 2. Apabila digunakan Seluruh bangunan yang dimiliki oleh pemain akan memiliki pertahanan selama 2 turn lawan. Apabila skill ini digunakan 2 kali berturut-turut, durasi tidak akan bertambah, namun menjadi nilai maksimum. Extra turn didapatkan jika Fort pemain tersebut direbut lawan. Setelah giliran pengaktifan skill ini berakhir, pemain selanjutnya tetap pemain yang sama. Arrack Up digunakan agar pertahanan bangunan musuh (termasuk Shield) tidak akan mempengaruhi penyerangan. Pemain mendapat skill ini jika pemain baru saja menyerang Tower lawan dan jumlah towernya menjadi 3. Critical Hit digunakan agar jumlah pasukan pada bangunan yang melakukan serangan tepat selanjutnya (hanya berlaku 1 serangan) efektif sebanyak 2 kali lipat pasukan. Skill ini akan menonaktifkan Shield maupun pertahanan bangunan, seperti Attack Up. Pemain mendapat skill ini jika lawan baru saja mengaktifkan skill Extra Turn. Instant Reinforcement memiliki efek menaikkan jumlah pasukan sebanyak 5 pasukan pada seluruh bangunan miliki pemain tersebut. Pemain mendapat skill ini di akhir gilirannya bila semua bangunan yang ia miliki memiliki level 4. Efek dari skill Barrage justru sebaliknya. Dengan Skill ini. Jumlah pasukan pada seluruh bangunan musuh akan berkurang sebanyak 10 pasukan. Pemain mendapat skill ini jika lawan baru saja bertambah bangunannya menjadi 10 bangunan. Pemain hanya dapat memiliki 10 Skill pada satu waktu. Dalam satu giliran, pemain dapat menggunakan banyak skill sekaligus.

Sedangkan Attack merupakan komando utama dari permainan ini. Setiap bangunan hanya dapat menyerang bangunan lainnya yang saling terhubung hanya satu kali. Awalnya, pemain dapat melihat bangunan mana yang dapat menyerang. Setelah bangunan dipilih, akan ditampilkan juga bangunan yang akan diserang dan pemain diminta mengerahkan sejumlah pasukan dari bangunan yang dipilihnya. Bangunan hanya dapat mengerahkan maksimal seluruh pasukan didalam bangunan tersebut. Jika pasukan yang dikerahkan kurang dari pasukan lawan, maka bangunan lawan tidak bisa direbut. Akan tetapi, jika pasukan sama atau lebih, maka bangunan tersebut dapat direbut. Dalam kasus dimana bangunan yang diserang memiliki pertahanan atau skill seperti yang telah disebutkan diatas aktif, nilai serang akan menjadi kurang dari pasukan yang dikerahkan. Sebaliknya, apabila skill Critical Hit digunakan, pasukan pemain dapat mengalahkan 2 pasukan musuh. Apabila bangunan telah berhasil direbut, bangunan akan kembali ke level 1. Diasumsikan bahwa jika masukan salah akan diberikan pemberitahuan dari program.

Persoalan utama dari Tugas besar ini adalah pengimplementasian permainan yang telah dideskripsikan sebelumnya ke dalam bahasa pemrograman C. Ada beberapa struktur data abstrak yang wajib digunakan untuk membuat implementasi permainan ini, seperti ADT point, Array Dinamis, Matriks, Mesin Karakter dan Mesin Kata, Queue, Stack, List, dan Graph.

Laporan ini secara khusus akan menceritakan mengenai implementasi permainan ke dalam bahasa C, mulai dari spesifikasi fitur-fitur yang digunakan, penjelasan struktur data yang digunakan, dan program utama. Kami menemukan beberapa algoritma menarik yang akan kami jelaskan pada bagian 4. Kami telah mengetes program kami dengan beberapa data set yang hasilnya akan kami sertakan kedalam laporan ini. Kami juga menyertakan pembagian kerja, dan melampirkan log activity, deskripsi spesifik dari tugas besar serta notulensi rapat dari tugas besar ini.

Secara umum, tugas besar ini telah dikerjakan secara baik. Hasil implementasi dari data test yang ditest pada program menunjukkan hasil yang positif.

# Penjelasan Tambahan Spesifikasi Tugas

Ada beberapa fitur yang spesifikasinya belum jelas atau tidak dijelaskan pada dari deskripsi yang telah diberikan sebelumnya. Pada subbab ini akan dijelaskan spesifikasi dari fitur-fitur tersebut.

## Skill ’Shield’

Seluruh bangunan yang dimiliki oleh pemain akan memiliki pertahanan selama 2 turn lawan. Apabila skill ini digunakan 2 kali berturut-turut, durasi tidak akan bertambah, namun menjadi nilai maksimum. Pemain mendapat skill ini jika setelah sebuah lawan menyerang, bangunan pemain berkurang 1 menjadi sisa 2.

## Skill ’Attack Up’

Pada giliran ini, setelah skill ini diaktifkan, pertahanan bangunan musuh (termasuk Shield) tidak akan mempengaruhi penyerangan. Pemain mendapat skill ini jika pemain baru saja menyerang Tower lawan dan jumlah towernya menjadi 3.

## Skill ’Critical Hit’

Setelah skill diaktifkan, jumlah pasukan pada bangunan yang melakukan serangan tepat selanjutnya (hanya berlaku 1 serangan) hanya efektif sebanyak 2 kali lipat pasukan. Skill ini akan menonaktifkan Shield maupun pertahanan bangunan, seperti Attack Up. Pemain mendapat skill ini jika lawan baru saja mengaktifkan skill Extra Turn.

## Command ’Save’

Digunakan untuk menyimpan status permainan sekarang ke dalam file eksternal berupa file txt.

## Command ’Load’

Digunakan untuk mengunggah status permainan dari file eksternal berupa file txt.

# Struktur Data (ADT)

Pada pembuatan program ,kami membuat beberapa ADT dengan rincian sebagai berikut.

## ADT ’Bangunan’

ADT ini memiliki type bangunan yang terdiri dari properti propertinya, seperti tipe, jumlah pasukan, level, nilai penambahan pasukan, maksimum penambahan pasukan, pertahanan, posisi dan Serangan telah dikakukan atau belum. ADT ini memiliki berbagai prmitif yang dapat membuat bangunan level 1, mengecek kesiapan naik level suatu bangunan, menaikkanlevel suatu bangunan, menambah jumlah pasukan dari suatu bangunan, menyalin bangunan, mencetak properti banguna ke layar, menambah level hingga level berikutnya, ,mencetak tipe bangunan dan mengebalikan level bangunan ke level 1

ADT ini merupakan perwujudan tipe bangunan itu sendiri. Dengan primitif serta tipe yang diberikan, ADT ini mampu berperan di beberapa command seperti attack, level\_up, print\_all\_bangunan, dan beberapa skill seperti critical hit, attack up dan sebagainya. Pemilihan bentuk ADT ini dikarenakan sederhana dan praktis untuk digunakan. ADT ini diimpkementasikan sebagai ADT dengan namafile ’bangunan.c’ dan ’bangunan.h’.

## ADT ’Daftar Bangunan’

ADT ini menyimpan tipe daftar bangunan yang terdiri dari Array yang mnyimpan bangunan, nilai efektif daftar bangunan serta jumlah maksimum dari bangunan. Secara umum, ADT ini mirip seperti array dinamis yang jumlah elemennya berubah-ubah. ADT ini memiliki berbagai primitif seperti MakeEmptyArray(membuat array kosong), Dealokasi(menghapus elemen dan mengembalikannya keserver), GetFirstIdx(mendapatkan elemen pertama), GetLastIdx(mendapatkan elemen terakhir), mengecek indeks yang valid, mengecek daftar penuh atau kosong, menyalin daftar, menambahkan elemen serta mencetak daftar ke layar. Elemen ditambahkan sebagai elemen terakhir daftar bangunan.

ADT ini merupakan perwujudan daftar bangunan itu sendiri. Dengan primitif serta tipe yang diberikan, ADT ini berguna sebagai bagian dari state pada permainan sehingga sering dilibatkan dalam mekanisme undo. Selain itu, ADT ini juga dilibatkan untuk menghadirkan pilihan bangunan yang akan diserang atau dinaikkan levelnya Pemilihan bentuk ADT ini dikarenakan sederhana, praktis dan hemat memori dalam penggunaannya. ADT ini diimplementasikan sebagai ADT Array Dinamis dengan namafile ’arraydin.c’ dan ’arraydin.h’.

## ADT ’Graph’

ADT ini menyimpan daftar bangunan dan keterhubungannya dengan bangunan lain. Daftar Bangunan diimplementasikan berupa list parent, dimana masing-masing bangunan menyimpan list lain (child) yang isinya daftar bangunan yang terhubung dengan bangunan tersebut. ADT ini mirip seperti ADT multilist. ADT ini memiliki berbagai primitif seperti mengecek graf kosong, membuat gaf kosong, mengalokasi parent, mendealokasi parent, mencari parent tertentu, menambahkan value parent pada awalgraf atau diakhir graf, mencetak graf kelayar dan mengecek apakah 2 buah bangunan saling terhubung atau tidak. Dengan primitif serta tipe yang diberikan, ADT ini berguna saat mekanisme penyerangan karena dalam menyerang diperlukan keterhubungan antar bangunan. ADT ini diimplementasikan sebagai ADT Array Dinamis dengan namafile ’graph.c’ dan ’graph.h’.

## ADT ’Daftar Bangunan Pemain’

ADT ini menyimpan daftar bangunan yang dimiliki oleh seorang pemain. ADT ini agak berbeda dengan ADT daftar bangunan karena pada ADT ini yang disimpan adalah indeks bangunannya saja. List sendiri terdiri dari indeks bangunan dan indeks bangunan setelahnya. Secara umum, ADT ini mirip seperti list linier yang menyimpan address dan address selanjutnya. ADT ini memiliki berbagai primitif seperti membuat list kosong, Alokasi dan Dealokasi (menghapus elemen dan mengembalikannya keserver), mencari elemen pada list dengan kriteria tertentu, menambahkan nilai dielemen pertama ataupun terakhir, menghapus nilai dielemen pertama ataupun terakhir,mencari nilai tertinggi suatu list, mencetak list kelayar maupun menggabungkan 2 list..

ADT ini merupakan perwujudan daftar bangunan pemain itu sendiri. Dengan primitif serta tipe yang diberikan, ADT ini berguna sebagai penghadir pilihanbangunan yang menyerang. Selain itu, ADT ini juga dilibatkan untuk menghadirkan pilihan bangunan yang akan diserang atau dinaikkan levelnya. Banyak skill juga membutuhkan peran dari ADT ini. Pemilihan bentuk ADT ini dikarenakan sederhana, praktis dan hemat memori dalam penggunaannya. ADT ini diimplementasikan sebagai ADT List Linier dengan namafile ’listlinier.c’ dan ’listinier.h’.

## ADT ’Matriks’

ADT ini menyimpan peta yang dimiliki oleh seorang pemain. ADT ini memiliki beberapa atribut seperti Char dan ID yang merpuakan bagian dari matriks. Matriks sendiri terdiri dari indeks dua dimensi . Secara umum, ADT ini mirip seperti list linier yang menyimpan address dan address selanjutnya. ADT ini memiliki berbagai primitif seperti membuat matriks kosong, mencari menampilkan first indeks dikolom tertentu, last indeks dikolom 2,dan mencetak peta kelayar.

ADT ini merupakan perwujudan representasi peta permainan itu sendiri. Dengan primitif serta tipe yang diberikan, ADT ini berguna sebagai representasi peta yang cukup membantu dalam penyerangan. ADT ini diimplementasikan sebagai ADT Matriks dengan namafile ’matriks.c’ dan matriks.h’.

## ADT ’Mesin Karakter’

ADT mesin karakter adalah ADT yang menerima masukan karakter, baik dari standard input maupun file eksternal. ADT ini terdiri dari pita yang berisi deret karakter dan tombol START dan ADV. Deret karakter dibaca pada jendela yang besarnya satu karakter. Terdapat juga lampu EOP yang menandakan akhir dari deret karakter. ADT ini mirip dengan ADT mesin karakter yang digunakan saat praktikum hanya saja terdapat lebih banyak fitur seperti menerima input dari file txt, file eksternal maupun dari standard input.

ADT ini sangat berguna didalam permainan. Dengan primitif serta tipe yang diberikan, ADT ini berguna membaca karakter, baik dari file konfigurasi, file eksternal maupun standard input yang nantinya akan diolah di ADT mesin kata. Bentuk ADT ini dipilih karena ADT ini sangat powerful dalam mengolah karakter. ADT ini diimplementasikan sebagai ADT Mesin Karakter dengan namafile ’mesinkar.c’ dan ’mesinkar.h’.

## ADT ’Mesin Kata’

ADT mesin kata adalah ADT yang mengolah masukan karakter, baik dari standard input maupun file eksternal dari ADT mesin karakter. ADT ini terdiri dari kumpulan karakter yang diakhiri enter atau NULL. ADT ini mirip dengan ADT mesin kata yang digunakan saat praktikum hanya saja terdapat lebih banyak fitur seperti mengabaikan enter dan spasi, memulai pembacaan kata, membaca input dari penguna, mengolah input dari file eksternal, menyalin kata, mengecek kesamaan kata dengan kata yang lain,dan mengecek inputan berupa integer tertentu.

ADT ini sangat berguna didalam permainan. Dengan primitif serta tipe yang diberikan, ADT ini berguna mengolah inputan dari eksternal program, baik dari file konfigurasi, file eksternal maupun standard input. Inputan tersebut dapat digunakan untuk membaca informasi peta dari file eksternal, membaca command dari interaksi user terhadap program, dan membaca state dari game yang sudah pernah disimpan. Bentuk ADT ini dipilih karena ADT ini sangat powerful dan multifungsi dalam mengolah input. ADT ini diimplementasikan sebagai ADT Mesin Kata dengan namafile ’mesinkata.c’ dan ’mesinkata.h’.

## ADT ’Player’

ADT ini memiliki type player yang terdiri dari properti propertinya, seperti daftar bangunan yang dimilikinya, daftar skill, giliran, dan jumlah shield yang telah dimiliki. ADT ini memiliki berbagai primitif, seperti dapat membuat player kosong, mencetak daftar bangunan dan skill yang dimiliki, menyalin data dari suatu pemain serta mendealokasi pemain dari sistem

ADT ini merupakan perwujudan tipe player itu sendiri. Dengan primitif serta tipe yang diberikan, ADT ini menjadi bagian dari state sehiingga turut dilibatkan dalam mekanisne undo. Selain itu, ADT ini sangat berperan dalam keseluruhan game,seperti mekanisme attack, level up, skill, end-turn, move, status, dll. Pemilihan bentuk ADT ini dikarenakan sederhana dan praktis untuk digunakan. ADT ini diimplementasikan sebagai ADT player dengan namafile ’player.c’ dan ’player.h’.

## ADT ’Point’

ADT ini memiliki type point yang terdiri dari koordinat x dan koordinat y. ADT ini memiliki berbagai primitif yang turut digunakan didalam game, seperti mengecek posisi, membuat koordinat, dan lain lain.ADT ini berfungsi untuk mengetahui koordinat bangunan dipeta terutama saat pembacaan file konfigurasi. Pemilihan bentuk ADT ini dikarenakan sederhana tapi multifungsi. ADT ini diimplementasikan sebagai ADT point dengan namafile ’point.c’ dan ’point.h’.

## ADT ’Skill’

ADT ini menyimpan daftar skill yang dimiliki oleh seorang pemain.. Secara umum, ADT ini mirip seperti queue hanya saja isi dari queue adalah skill para pemain yang dilambangkan sebagai ineteger. ADT ini memiliki berbagai primitif seperti mengecek queue kosong atau penuh, menghitung elemen queue, membuat queue kosong, mendealokasi elemen queue, menambah dan menghapus elemen, mengcopy queue dan mencetak daftar queue ke layar.

ADT ini merupakan perwujudan daftar skill itu sendiri. Dengan primitif serta tipe yang diberikan, ADT ini berguna sebagai daftar skill yang dilibatkan didalam penggunaan command skill. Selain itu, ADT ini juga dilibatkan dalam fitur game lainnya karena banyak kondisi yang memungkinkan skill didapatkan oleh suatu pemain.. Pemilihan bentuk ADT ini dikarenakan sederhana,serta praktis dalam penggunaannya. ADT ini diimplementasikan sebagai ADT queue dengan namafile ’queue.c’ dan ’queue.h’.

## Modul ’Read File’

Modul read file adalah modul yang berfungsi mengolah masukan dari file konfigurasi menjadi state dan bentukan dasar didalam game. Modul ini terdiri dari 2 fitur utama, yakni membaca seluruh bangunan, membaca keterhubungan antar bangunan ke dalam graf. Kedua fitur disatukan didalam prosedur readkonfigurasifile yang fungsinya membaca file konfigurasi dan mengubahnya menjadi state, matriks dan graf yang sesuai.

Modul ini berguna untuk membaca file konfigurasi. Bentuk modul ini dipilih karena modul ini sangat powerful dan multifungsi dalam mengolah input. modul ini diimplementasikan sebagai modul Read File dengan namafile ’readfile.c’ dan ’readfile.h’.

## ADT ’Daftar State’

ADT ini menyimpan daftar state yang telah dilakukan sejak mulai giliran hingga digunakan command skill atau giliran berakhir. ADT ini memiliki berbagai primitif seperti membuat daftar state kosong, mengecek daftar state kosong atau penuh, memasukkan dan menghapus state kedalam daftar state, mengcopy state serta mekanisme undo. Dengan primitif serta tipe yang diberikan, ADT ini berguna saat mekanisme undo. ADT ini diimplementasikan sebagai ADT stack dengan namafile ’stack.c’ dan ’stack.h’.

## ADT ’State’

ADT ini memiliki type state yang terdiri dari properti propertinya, seperti Daftar bangunan, player, dan Critical Hit. ADT ini memiliki berbagai primitif, seperti dapat membuat state kosong, mencetak daftar bangunan dan atribut kedua pemain yang dimiliki, menyalin data state serta mendealokasi state dari sistem.

ADT ini merupakan perwujudan state daripermainanitu sendiri. Dengan primitif serta tipe yang diberikan, ADT ini meyimpan setiap efek dari command yang dilakukan kedua pemain. Pemilihan bentuk ADT ini dikarenakan sederhana dan praktis untuk digunakan. ADT ini diimplementasikan sebagai ADT state dengan namafile ’state.c’ dan ’state.h’.

## Modul ’Save Load’

Modul read file adalah modul yang berfungsi mengolah masukan dari file eksternal menjadi state dan bentukan dasar didalam game serta menyimpan state dan bentukan dasar kedalam file eksternal. Modul ini terdiri dari 3 fitur utama, yakni menyimpan state dan bentukan dasar kedalam file eksternal, mengolah masukan dari file eksternal menjadi state dan bentukan dasar didalam game dan meminta filename tempat menyimpanan atau pengunduhan. ADT ini berguna untuk dalam prosedur command save dan load. Bentuk Modul ini dipilih karena modul ini sangat sederhana namun berfungsi dengan baik. Modul ini diimplementasikan sebagai Modul Save dan load dengan namafile ’save\_load.c’ dan ’save\_load.h’.

## Modul ’Function’

Modul function adalah modul yang berisikan semua fungsi dan prosedur yang mendukung semua komando dan skill yang digunakan didalam permainan, mulai dari skill, mekanisme attack, status, end\_turn, move dan exit. Didalam modul ini, terdapat primitif seperti yang mendukung command skill, seperti mekanisme penambahan skill, instant upgrade, shield, End turn, extra turn,attack up, critical hit, mengecek kondisi penambahan instant reinforcement, menghitung jumlah tower dan skill barrage. Terdapat juga primitif yang mendukung command lain, seperti level up dan move yang mana didalam modul ini terdapat fungsi untuk memilih bangunan yang dinaikan/dipindahkan pasukkannya dan prosedur untuk menaikkan level bangunan serta memindahkan pasukan. Didalam fungsi ini juga dimuat prosedur pendukung untuk mekanisme attack, dimana ada 3 prosedur utama seperti memilih bangunan penyerang dan diserang, mekanisme utama penyerangan dan mekanisme dasar dalam menyerang. Selain itu, didalam modul ini juga disimpan prosedur tambahan seperti menampilkan status, mencetak tampilan game dengan warna, tampilan akhir game dan help. Alasan dipilih bentuk modul agar mudah di debug dan diperlukan kolaborasi antar ADT dalam pengerjaan mekanisme ini sehingga semuanya ditulis dalam satu modul agar ringkas.

## Modul ’pcolor’

Modul pcolor adalah modul yang berfungsi untuk memberi warna pada tampilan permainan. Modul ini menyimpan primitif untuk mencetak 8 warna,yakni merah, biru, hijai, kuning, normal,magenta dan cyan. Modul ini berguna untuk menginformasikan milik siapa suatu bangunandan giliran yang ditentukan program. Bentuk modul ini dipilih karena modul ini sangat sederhana namun berfungsi dengan baik. Modul ini diimplementasikan sebagai modul pcolor dengan namafile ’pcolor.c’ dan ’pcolor.h’.

Secara umum, keterhubungan antar modul dapat dijelaskan melalui diagram berikut.

Daftar Bangunan Pemain

Point

Bangunan

Skill

Daftar Bangunan

ReadFile

Save and Load

pcolor

Player

State

Function

Daftar State

Matriks

Graf

Legends:

ADT digunakan di ADT Target

Mesin Kata

Mesin Karakter

Diagram 3.1 Diagram Ilustrasi keterhubungan antar modul dalam program.

# Program Utama

Program utama dalam permainan *Avatar World War: a True Dictator* ini terletak di file dengan nama ’main.c’. Di dalam program, awalnya program mengimpor fungsi fungsi yang sudah didefinisikan pada header file lain. Hal ini dikarenakan main membutuhkan hampir semua modul yang tersedia pada file lain. Setelah itu, program mulai mendeklarasikan Tipe tipe yang dibutuhkan, seperti State, daftar State, Matriks, dan Graf. Dideklarasikan pula boolean tambahan yang mendukung keberjalanan skill yang akan digunakan didalam program. Setelah beberapa intro dari permainan, program akan membaca file konfigurasi apabila dipilih new game atau membaca file eksternal apabila pemain memilih load game. Saat ini, State, Matriks dan Graf sudah berisikan data dari konfigurasi. Pada saat yang hampir sama, Atribut Turn pada pemain 1 diaktifkan, menandakan giliran saat ini adalah milik pemain 1. Dibuat juga daftar state yang isinya baru berupa giliran pemain1 yang aktif dan ditampilkan map, daftar command, dan daftar Bangunan dari pemain 1.

Lalu, selama permainan tidak diakhiri (salah satu pemain memasukkan masukan ’EXIT’) atau salah satu pemain tidak kehabisan bangunan, pemain akan disuguhkan dengan permintaan masukan dari komputer, dengan warna sesuai dengan giliran pemain saat itu. Lalu, masukan pemain akan disesuaikan. Apabila pemain memasukkan masukan ’ATTACK’, maka program akan memanggil fungsi Attack dari modul function, yang mana isinya adalah prosedur untuk menyerang musuh,mulai dari memilih bangunan penyerang dan bertahan, mekanisme penyerangan hingga mekanisme dasar. Apabila pemain memasukkan masukan ’LEVEL\_UP’, maka program akan memanggil fungsi LevelUp dari modul function, yang berfungsi mengubah suatu level bangunan yang telah dipilih. Apabila pemain memasukkan masukan ’SKILL’, maka program akan memanggil fungsi Skill dari modul function. Fungsi Skill ini menghapus Skill dari daftar skill dan memanggil fungsi-fungsi terkait dengan skill yang dipilih sesuai masukan selanjutnya. Apabila pemain memasukkan masukan ’MOVE’, maka program akan memanggil fungsi Move dari modul function. Fungsi Move ini memanggil fungsi lain untuk mencari 2 bangunan yang menjadi target lalu mengurangi pasukan dari suatu bangunan dan menambahkan nilai yang sama pada bangunan target. Pada akhir prosedur pada semua masukan sebelumnya, state akan ditambahkan ke daftar state. Apabila pemain memasukkan masukan ’UNDO’, maka program akan menghapus sebuah state dari daftar state, dengan fungsi Undo dari ADT Stack. Namun, apabila pemain memasukkan masukan SKILL atau END\_TURN, maka program akan menghapus semua state dari daftar state, kecuali state terakhir, dengan fungsi EndTurnState dari ADT Stack. Apabila pemain memasukkan fungsi untuk menampilkan sesuatu, seperti ’STATUS’,’HELP’, ’PRINT\_ALL\_BANGUNAN’, maupun ’PRINT\_GRAPH’, maka program akan memanggil fungsi terkait dan penambahan state tidak dilakukan. Apabila pemain memasukkan masukan ’SAVE’, maka program akan memanggil fungsi save dari modul function dan menyimpan state terbaru kedalam file eksternal.

Permintaan masukan beserta hasil masukan tadi akan terus berulang hingga kondisi dimana permainan dapat dinyatakan selesai. Apabila permainan dimenangkan salah satu pihak,maka programakan memanggil fungsi GameEnd yang tujuannya mengakhiri game serta mengucapkan selamat kepada pemain yang memenangkan pertandingan.

Secara umum, alur program dapat dijelaskan melalui diagram flowchart berikut.

Panggil Fungsi

Load

Load Game

Daftar State,State Matriks,Boolean,Graphdan Tipe Pendukung lainnya terbentuk

Main Menu

Intro

Deklarasi Tipe Dasar dan Boolean Penanda Skill

Panggil Fungsi

ReadKonfigurasiFile

New Game

State ditambahkan kedaftar state

EfekFungsi Terlaksana

END??

Attack,Move Level Up

EfekFungsi Terlaksana

Skill, End turn

Minta Input

Isi Daftar state dihapus hingga tersisa satu state terakhir

State terakhir disimpan

Save

END

Menampilkan Sesuatu

Help, Status

Menghapus state terakhir dari daftar state, lalu mengubah state menjadi state yang berada di daftar state

Undo

Exit

Diagram 4.1 Diagram flow chart algoritma program utama

# Algoritma-Algoritma Menarik

## PrintQSkill & PrintTopSkill

Dalam keberjalanan game, dibutuhkan suatu algoritma untuk mencetak daftar skill yang telah dimiliki seorang pemain. Algoritma kami untuk melakukan hal tersebut adalah sebagai berikut.

void PrintQSkill (Queue Q){

    infotype X;

    printf("Skill Avaible | ");

    while (!IsEmpty(Q)){

        Del(&Q,&X);

        if (X == 1){

            printf("InstantUpgrade |");

        } else if (X == 2) {

            printf("Shield |");

        } else if (X == 3) {

            printf("Extra Turn |");

        } else if (X == 4){

            printf("AttackUp |");

        } else if (X == 5){

            printf ("CriticalHit |");

        } else if (X == 6) {

            printf("InstantReinforcement |");

        } else if (X == 7) {

            printf("Barrage |");

        }

    }

    printf("\n");

}

Gambar 5.1 Potongan program PrintQSkill.

Bagi pemula dalam dunia pemrograman, sekilas prosedur tersebut tampak aneh, dikarenakan untuk mencetak daftar skill, kami harus ’menghapus’ skill tersebut dari daftar skill. Akan tetapi, kenyataannya tidak. Apabila dilihat kembali, parameter yang kami gunakan pada prosedur adalah parameter input, sehingga apapun prosedur didalam program tidak akan mengganggu daftar skill yang asli. Hal ini akan berbeda hasilnya dengan prosedur yang parameternya berupa input output, dimana hasil dari prosedur akan turut mengganggu daftar skill yang ada.

# Data Test

Isi dengan data test dan penjelasan yang diperlukan. Hal-hal yang harus dijelaskan untuk tiap data test:

* Fitur apa (saja) yang dites dengan data test ini
* Hasil apa yang seharusnya diberikan
* Penjelasan lain (jika ada)
* Data test-nya sendiri.

Untuk mengetest program, kami membuat 2 data test berupa file berekstensi .txt.

## <Data Test 1>

Isi dengan penjelasan mengenai data test.

## <Data Test 2>

Isi dengan penjelasan mengenai data test.

# Test Script

Isi dengan skenario test yang dimungkinkan untuk semua fitur yang ada. Bisa dibuat dalam bentuk tabel sebagai berikut:

| **No.** | **Fitur yang Dites** | **Tujuan Testing** | **Langkah-Langkah Testing** | **Input Data Test** | **Hasil yang Diharapkan** | **Hasil yang Keluar** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | <Fitur 1> | … | … | <tinggal refer ke data test di atas> | … |  |
| 2 | <Fitur 1> | … | … | … | … |  |
| 3 |  |  |  |  |  |  |
| … |  |  |  |  |  |  |
| 5 | <Fitur 2> | … | … | … | … |  |
| 6 | <Fitur 2> | … | … | … | … |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Tabel 7.1 Test skenario untuk setiap fitur pada program.

# Pembagian Kerja dalam Kelompok

|  |  |  |
| --- | --- | --- |
| Nama | NIM | Pembagian Kerja |
| Muhammad Zunan Alfikri | 13518019 |  |
| Brandon Oktavian Pardede | 13518043 |  |
| Felicia Gillian Tekad Huerah | 13518070 |  |
| Muhammad Ayyub Abdurrahman | 13518076 |  |
| Hanif Muhammad Gana | 13518127 |  |

Tabel 8.1 Pembagian kerja dalam kelompok.

# Lampiran

## Deskripsi Tugas Besar 2

Isi dengan deskripsi Tugas besar (ambil dari file deskripsi Tugas besar).

## Notulen Rapat

Isi dengan semua notulen rapat.

## Log Activity Anggota Kelompok

|  |  |  |
| --- | --- | --- |
| NIM / Nama | Waktu | Activity |
| 13518019  Muhammad Zunan Alfikri |  |  |
|  |  |
|  |  |
|  |  |
| 13518043  Brandon Oktavian Pardede |  |  |
|  |  |
|  |  |
|  |  |
| 13518070  Felicia Gillian Tekad Huerah |  |  |
|  |  |
|  |  |
|  |  |
| 13518076  Muhammad Ayyub Abdurrahman |  |  |
|  |  |
|  |  |
|  |  |
| 13518127  Hanif Muhammad Gana |  |  |
|  |  |
|  |  |
|  |  |

Tabel 9.1 Log Activity Anggota kelompok.

## <Lain-Lain>

Hanya diisi jika memang ada hal-hal lain yang perlu dilampirkan. Jika tidak ada yang ditambahkan, hapus subbab ini.