

LAPORAN TUGAS BESAR

IF2110/Algoritma dan Struktur Data

BATTLE OF OLYMPIA

Dipersiapkan oleh:

Kelompok 01 – K02

13516035 – M. Sulthan Adhipradhana

13516062 – Yusuf Rahmat Pratama

13516083 – Abram Perdanaputra


13516086 – Dandy Arif Rahman

13516095 – Faza Fahleraz

13516116 – Ahmad Izzan

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

	Sekolah Teknik Elektro dan Informatika ITB	Nomor Dokumen		Halaman
		<i>IF2110-TB-1-2</i>		27
		<i>Revisi</i>	1	5/11/2017

Daftar Isi

1. Ringkasan	4
2. Penjelasan Tambahan Spesifikasi Tugas	5
2.1. Unit White Mage	5
2.2. Probabilitas Serangan	5
2.3. Command Next Unit	5
2.4. Pergerakan Unit Secara Diagonal	5
2.5. Save dan Load Game	5
3. Struktur Data (ADT)	6
3.1. ADT Jam	6
3.2. ADT Point	6
3.3. ADT Matriks	6
3.4. ADT Mesin Karakter + Mesin Kata	6
3.5. ADT Queue	6
3.6. ADT Stack	7
3.7. ADT List Linier (Double Pointer)	7
3.8. ADT Castle	7
3.9. ADT Map	7
3.10. ADT Player	7
3.11. ADT Tower	8
3.12. ADT Unit	8
3.13. ADT Village	8
4. Program Utama	9
5. Algoritma-Algoritma Menarik	10
5.1 Algoritma BFS	10
6. Data Test	11
6.1. Main Menu Program	11
6.2. New Game	12
6.3. Command Recruit	13
6.4. Command Move	14
6.5. Command Attack	15
6.6. Command Info	17
6.7. Kondisi Menang	18
6.8. Command Change Unit	19
6.9. Next Unit	20
6.10. Command Undo	21
7. Test Script	25
8. Pembagian Kerja dalam Kelompok	26
9. Lampiran	27
9.1. Deskripsi Tugas Besar 2	27

9.2. Notulen Rapat.....	27
9.3. Log Activity Anggota Kelompok	28

1. Ringkasan

Pada tugas besar ini, dibuat sebuah program berupa game ber-*genre turn-based strategy* menggunakan bahasa C. Game ini dieksekusi menggunakan *command line* dan tanpa menggunakan GUI. Game ini memiliki mekanika sesuai algoritma yang sudah diberikan dalam spesifikasi, dan menggunakan berbagai tipe struktur data yang berbeda.

Dalam laporan ini akan dibahas lebih lengkap spesifikasi dari game yang dibuat, berbagai tipe struktur data yang digunakan, penjelasan algoritma game yang telah dibuat, contoh implementasi program, pembagian tugas, dan lampiran-lampiran yang diperlukan.

Secara keseluruhan, pengerjaan tugas besar ini dilakukan berkelompok dan bersifat modular, dengan pengaplikasian beberapa struktur data yang dibutuhkan sesuai spek. Selain itu, tugas besar ini bermanfaat sebagai salah satu implementasi terhadap berbagai algoritma dan struktur data yang telah dipelajari di mata kuliah Algoritma & Struktur Data.

Deskripsi umum persoalan, dalam tugas besar ini, kami membuat game yang mengadopsi dari *Battle For Wesnoth*. Game ini terdiri dari 2 pemain, dimana masing-masing pemain memiliki *king, castle, dan tower*. Objektif dari game ini adalah membunuh king dari lawan untuk menang. Untuk mencapai objektif dari game ini, pemain bisa merekrut pasukan / unit, menduduki village, menyerang unit & pasukan musuh.

Secara singkat laporan ini terdiri dari ringkasan, penjelasan tambahan spesifikasi tugas, struktur data yang digunakan, program utama, algoritma-algoritma menarik, data test, test script, pembagian kerja kelompok, dan lampiran.

Kesimpulan dari tugas besar ini, game berhasil dibuat dengan spesifikasi-spesifikasi yang wajib, ditambah spesifikasi-spesifikasi bonus diantaranya fitur untuk menyimpan game, melanjutkan game, next unit, unit white mage, kemungkinan serangan, pergerakan unit secara diagonal dan kami membuat user interface sederhana yang membuat user lebih mudah dalam memainkan game ini.

2. Penjelasan Tambahan Spesifikasi Tugas

2.1. Unit White Mage

White Mage merupakan unit yang dapat melakukan heal kepada unit yang berlokasi di kiri, kanan, atas, bawah. White Mage dapat menaikan nyawa dari unit sebesar n pada setiap turn. White Mage tidak dapat mengobati diri sendiri.

2.2. Probabilitas Serangan

Probabilitas serangan merupakan mekanisme penyerangan sehingga setiap serangan mempunyai kesempatan menyerang sebesar n persen. Sehingga musuh mempunyai kemungkinan terkena atau tidak terkena serangan sebesar n %.

2.3. Command Next Unit

Command Next Unit merupakan mekanisme game sehingga pemain dapat mengganti unit secara otomatis jika terdapat unit yang masih memiliki movement points atau masih mempunyai kesempatan serangan.

2.4. Pergerakan Unit Secara Diagonal

Secara otomatis, ketika Command Move digunakan, pemain dapat bergerak secara diagonal. Sistem akan menampilkan seluruh kemungkinan gerakan dengan karakter pager. Pada lokasi diagonal, karakter tetap menggunakan 2 movement points.

2.5. Save dan Load Game

Pemain dapat menyimpan dan me-load game yang sedang berjalan ke dalam file eksternal. Command save digunakan untuk menyimpan game yang sedang berjalan ke dalam file eksternal. Command load berguna untuk meneruskan game yang sudah disimpan ke file eksternal.

3.Struktur Data (ADT)

3.1. ADT Jam

ADT Jam ini digunakan untuk menyimpan waktu dalam sebuah tipe yang terstruktur khusus untuk waktu. Dalam tipe jam, terdapat 3 *integer* berupa jam, menit, dan detik yang secara keseluruhan merepresentasikan waktu yang disimpan. ADT Jam ini disimpan dalam file "Jam.h" dan "Jam.c". ADT ini digunakan untuk menandakan waktu dari command save.

3.2. ADT Point

ADT Point ini digunakan sebagai penanda lokasi objek (unit dan bangunan) dalam peta. Point ini bertindak sebagai koordinat, dimana dalam tipe point terdiri dari 2 variabel *integer* berupa absis dan ordinat. ADT Point ini disimpan dalam file "Point.h" dan "Point.c". ADT ini digunakan sebagai koordinat dari peta.

3.3. ADT Matriks

ADT Matriks digunakan untuk menyimpan data untuk peta dunia game ini, sekaligus indeks dari matriks ini dijadikan representasi koordinat yang ditunjuk menggunakan ADT Point. Dalam tipe Matriks ini terdiri dari matriks itu sendiri, yang didalamnya terdapat data jenis unit dan jenis bangunan yang berada pada setiap koordinat dalam dunia, serta disimpan nilai efektif matriks baris dan kolomnya supaya tidak mengakses indeks kosong. ADT ini disimpan dalam file "map.h" dan "map.c". ADT ini digunakan sebagai representasi peta.

3.4. ADT Mesin Karakter + Mesin Kata

ADT Mesin Karakter dan Mesin Kata digunakan untuk menyimpan data ke file eksternal, dan membaca data dari file eksternal. Mesin Kata digunakan untuk *parsing string* yang terdapat pada file eksternal dan menyimpan *string* tersebut dalam struktur data internal yang berupa Point, Matriks, Queue, Stack, dan struktur data lainnya. ADT ini digunakan untuk memarsing file eksternal.

3.5. ADT Queue

ADT Queue digunakan untuk menyimpan urutan pemain dalam sebuah permainan. Dengan aturan FIFO dalam Queue, maka struktur data ini cocok digunakan untuk *swap turn* setiap pemain di dalam *game*, dengan cara menghapus elemen pertama untuk menentukan *current player*. ADT ini disimpan dalam file "queue.h" dan "queue.c". ADT ini digunakan untuk membuat sistem next turn.

3.6. ADT Stack

ADT Stack digunakan untuk mengimplementasikan UNDO dalam permainan, yaitu command untuk mengembalikan posisi awal unit sebelum unit tersebut dilakukan command MOVE. Dalam element stack ini disimpan id unit dan posisi awal unit tersebut. ADT ini disimpan dalam file "stack.h" dan "stack.c". ADT ini digunakan untuk fitur undo.

3.7. ADT List Linier (Double Pointer)

ADT List Linier dengan variasi Double Pointer digunakan dalam banyak hal dalam tugas besar ini. Sebagian besar struktur data ini digunakan sebagai database untuk permainan. Beberapa contoh adalah ListPlayer, ListUnit, dll. ADT List Linier ini disimpan dalam file "player.c", "player.h", ADT ini digunakan untuk penyimpanan unit dan player.

3.8. ADT Castle

ADT Castle ini digunakan untuk mendefinisikan bangunan castle yang terdiri dari id yang menyatakan unit apa yang terdapat disana, player id yang menyatakan kepemilikan dari castle tersebut, position yang menyatakan koordinat dari castle tersebut, has_unit yang menyatakan apakah terdapat unit di dalam castle tersebut. ADT castle ini disimpan dalam file "castle.h" dan "castle.c". ADT ini digunakan untuk representasi bangunan castle yang ada di map.

3.9. ADT Map

ADT Map ini digunakan mendefinisikan map yang diimplementasikan sebagai matriks 2 dimensi, jumlah baris efektif, dan jumlah kolom efektif. ADT Map ini disimpan dalam file "map.h" dan "map.c". ADT ini digunakan untuk merepresentasikan map.

3.10. ADT Player

ADT Player ini digunakan untuk mendefinisikan player, struktur dari player adalah id yang mendefinisikan id dari player tersebut, alive yang menyatakan apakah player masih hidup, cash yang menyatakan uang yang dimiliki player, income yang menyatakan pendapatan yang diterima player setiap turn, upkeep yang menyatakan pengeluaran yang harus dikeluarkan player setiap turn, king_id yang menyatakan id dari king player, array of units yang berisi id unit yang dimiliki player, unit_count yang menyatakan banyaknya unit yang dimiliki oleh player, array of villages yang berisi id village yang dimiliki player, village_count yang menyatakan jumlah village yang dimiliki player. ADT player ini disimpan dalam file "player.h" dan "player.c" ADT ini digunakan untuk merepresentasikan player.

3.11. ADT Tower

ADT Tower ini digunakan untuk mendefinisikan tower, struktur dari tower adalah id yang menyatakan id unit dari tower tersebut, `player_id` yang menyatakan id dari player pemilik tower tersebut. ADT tower ini disimpan dalam file "tower.h" dan "tower.c". ADT ini digunakan untuk merepresentasikan bangunan tower yang ada di map.

3.12. ADT Unit

ADT Unit ini digunakan untuk mendefinisikan unit, struktur dari unit adalah id yang menyatakan id dari unit tersebut, `player_id` yang menyatakan id player pemilik unit tersebut, `attack_type` yang menyatakan tipe serangan dari unit tersebut, `max_health` yang menyatakan darah maksimum dari unit tersebut, `health` yang menyatakan darah sekarang dari unit tersebut, `attack` yang menyatakan besaran serangan, `max_move_points` yang menyatakan nilai maksimum dari move point unit tersebut, `move_points` yang menyatakan nilai move point yang tersisa dari unit tersebut, `can_attack` yang menyatakan apakah unit tersebut memiliki kesempatan untuk menyerang, `location` yang menyatakan koordinat dari unit tersebut, `price` yang menyatakan harga dari unit tersebut, `heal` yang menyatakan besaran heal yang bisa dilakukan oleh unit tersebut. ADT unit ini disimpan dalam file "unit.h" dan "unit.c". ADT ini digunakan untuk merepresentasikan unit-unit yang terdapat di game.

3.13. ADT Village

ADT Village ini digunakan untuk mendefinisikan village, struktur dari village adalah id yang menyatakan id dari unit tersebut dan `player_id` yang menyatakan id player pemilik unit tersebut. ADT village ini disimpan dalam file "village.h" dan "village.c". ADT ini digunakan untuk merepresentasikan bangunan village yang ada di map.

4.Program Utama

Program utama ini terdiri dari dua fungsi yaitu InitializeGame dan PlayGame. Fungsi InitializeGame berfungsi untuk memunculkan intro dari game ini, kemudian meminta user untuk memulai game baru atau melanjutkan game. Jika user memilih game baru maka akan dibuat data baru ke database, sedangkan jika memilih melanjutkan game maka akan memakai data game yang sudah disimpan di file eksternal. Kemudian menjalankan fungsi PlayGame yang berfungsi untuk menjalankan permainan, fungsi ini merupakan fungsi looping yang disetiap loop nya akan mengganti turn dari player dan menambah efek-efek terhadap status dari player tersebut, misal menambah income, mengurangi gold sebesar upkeep, menambah health unit dalam kondisi tertentu jika mempunyai unit white mage tau unit beard di village.

5.Algoritma-Algoritma Menarik

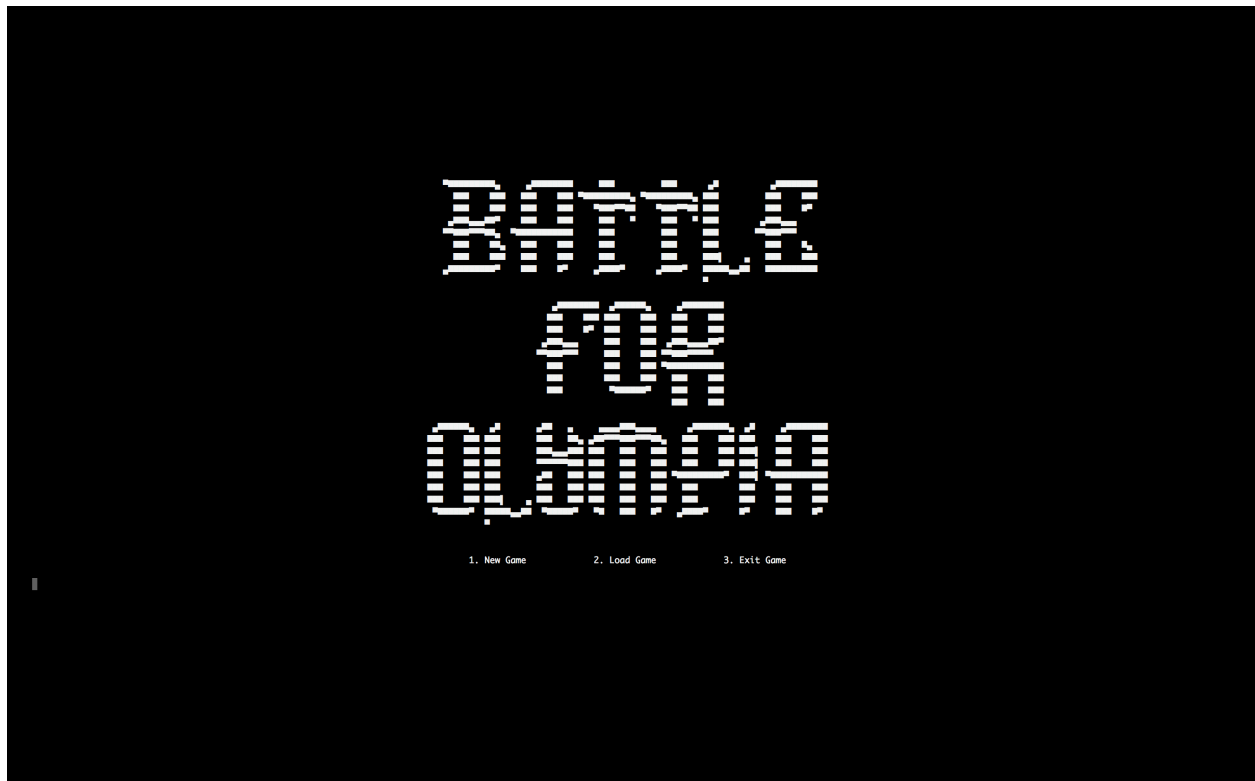
5.1 Algoritma BFS

Pada command move, digunakan algorithm BFS (Breadth First Search) agar dapat menampilkan seluruh kemungkinan gerakan yang pemain dapat lakukan. Breadth First Search adalah suatu algoritma yang mencari seluruh tetangga lebih dulu sebelum mendalami level selanjutnya.

6.Data Test

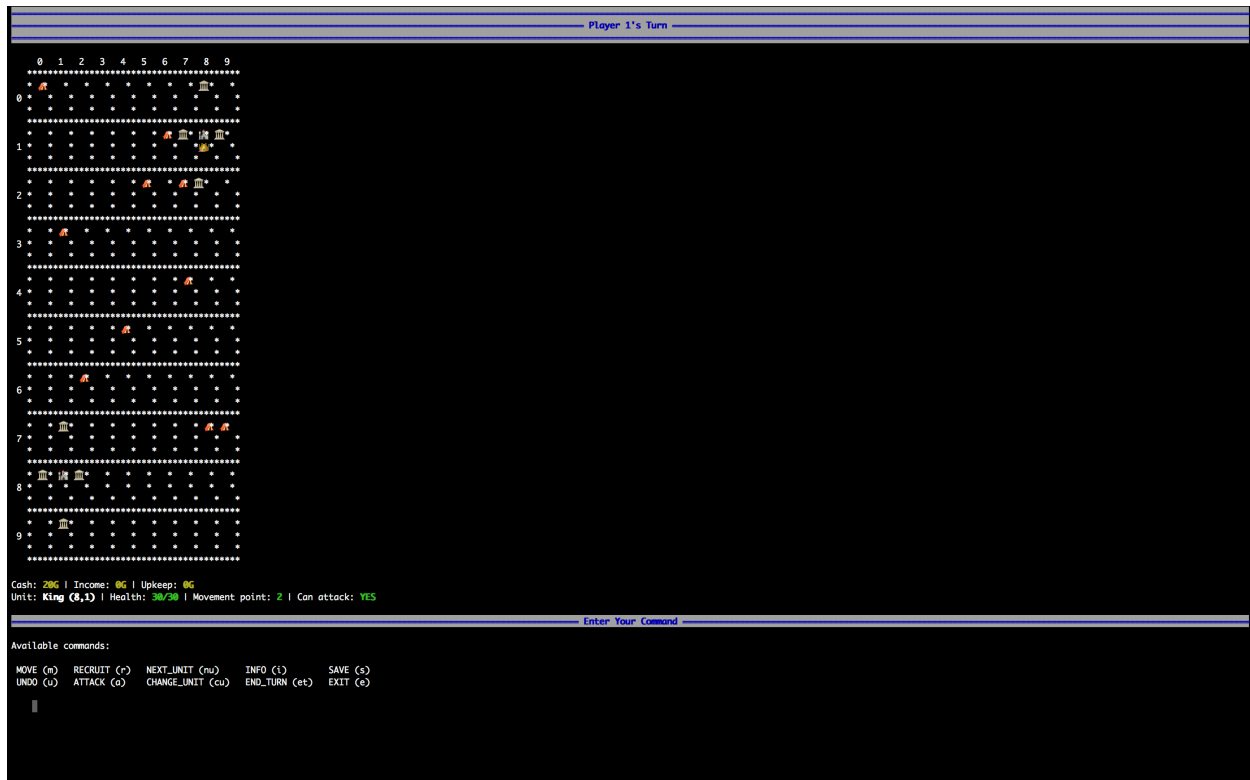
6.1. Main Menu Program

Pengetesan saat pertama kali memulai permainan. Program data mengeluarkan logo “Battle of Olympia” dan mengeluarkan main menu.



6.2. New Game

Pengetesan saat memilih menu new game. Program akan mengeluarkan output berupa peta beserta unit, bangunan, dan status dari pemain, serta command yang dapat diakses.



6.3. Command Recruit

Pengetesan saat memilih command recruit. Program akan mengeluarkan castle yang tersedia. Jika tidak ada castle yang tersedia, program akan mengeluarkan pesan castle penuh. Setelah player memilih castle, pemain akan memilih unit. Jika uang untuk me-recruit tidak cukup, program akan mengeluarkan pesan kesalahan.

```
----- Player 1's Turn -----
0 1 2 3 4 5 6 7 8 9
.....
0 .....
1 .....
2 .....
3 .....
4 .....
5 .....
6 .....
7 .....
8 .....
9 .....
.....

Cash: 200 | Income: 00 | Upkeep: 00
Unit: King (0,1) | Health: 30/30 | Movement point: 2 | Can attack: YES

----- Choose Empty Castle to Recruit Unit -----
1. Castle (8,2)
2. Castle (8,0)
3. Castle (9,1)
4. Castle (7,1)
>>> |
```

```
----- Player 1's Turn -----
0 1 2 3 4 5 6 7 8 9
.....
0 .....
1 .....
2 .....
3 .....
4 .....
5 .....
6 .....
7 .....
8 .....
9 .....
.....

Cash: 150 | Income: 00 | Upkeep: 10
Unit: King (8,1) | Health: 30/30 | Movement point: 2 | Can attack: YES

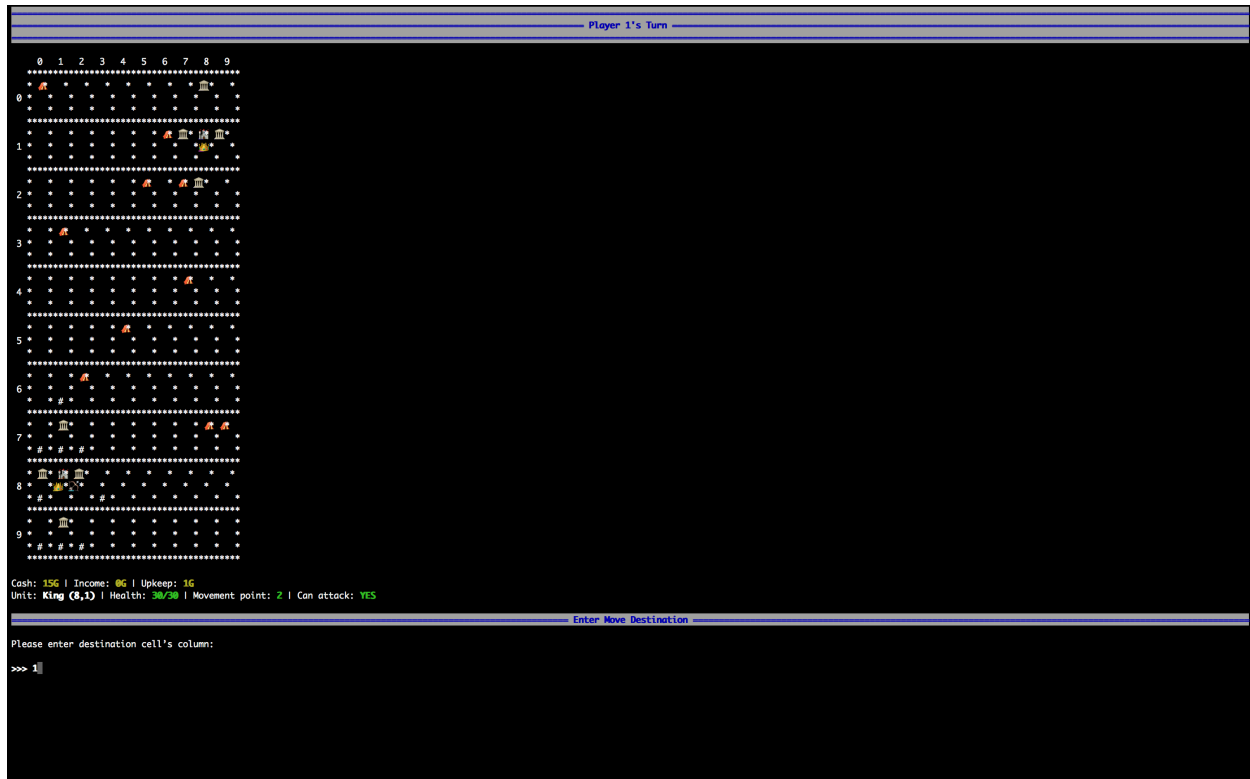
----- You have recruited an Archer! -----
----- Enter Your Command -----

Available commands:
MOVE (m)  RECRUIT (r)  NEXT_UNIT (nu)  INFO (i)  SAVE (s)
UNDO (u)  ATTACK (a)  CHANGE_UNIT (cu)  END_TURN (et)  EXIT (e)

>>> |
```

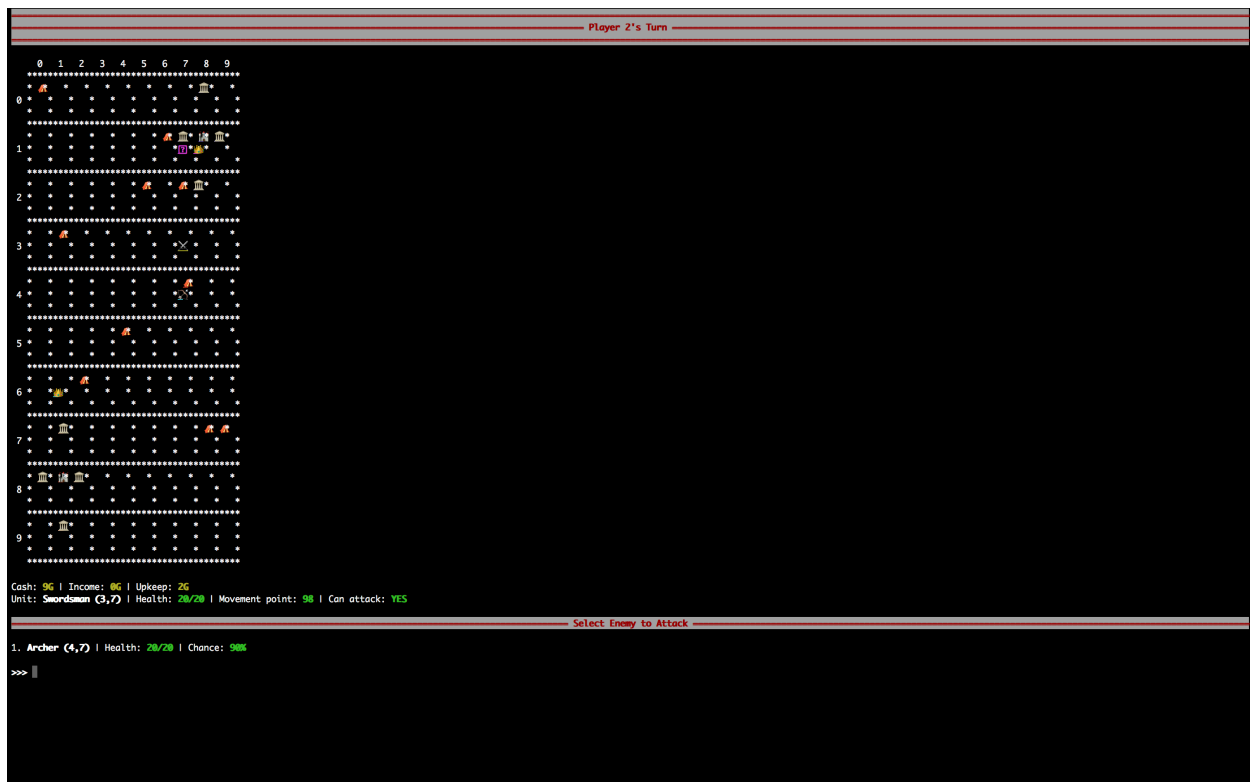
6.4. Command Move

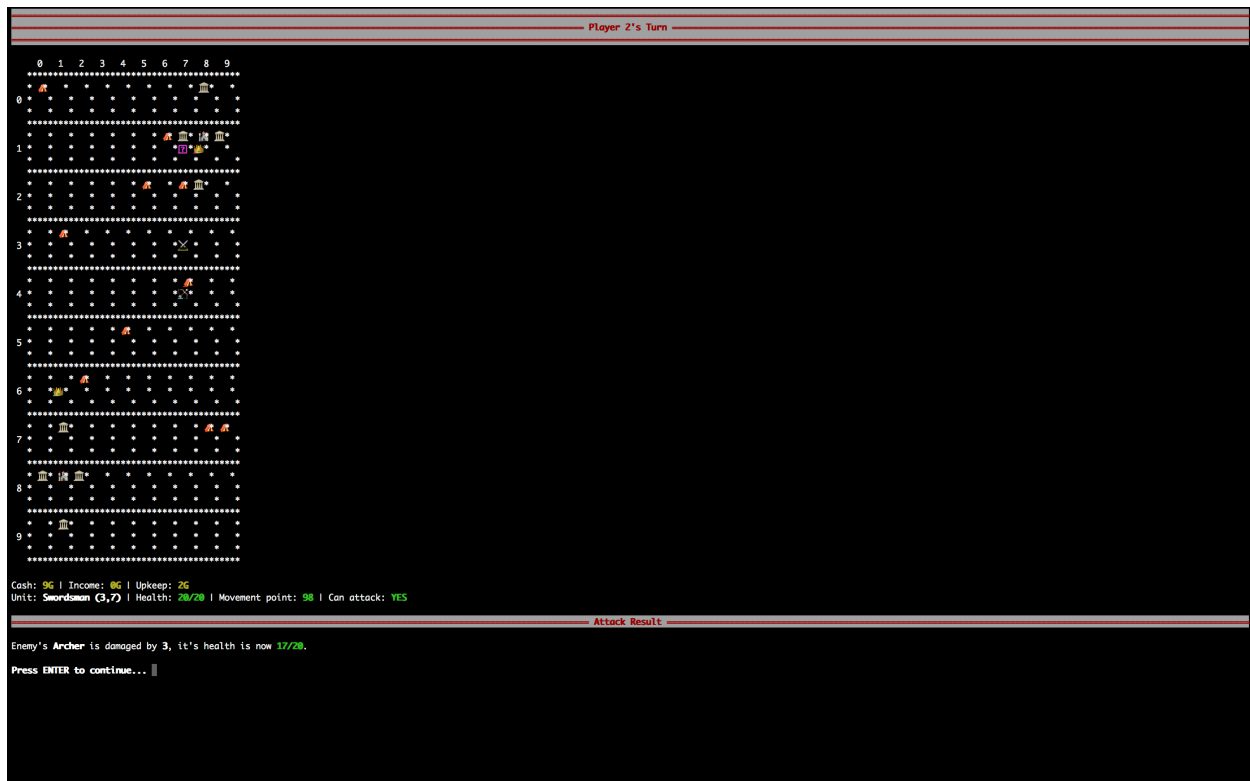
Pengetesan ini dilakukan untuk memilih command move. Program akan meminta masukan berupa kolom dan baris tujuan. Jika pemain tidak memiliki movement point yang berkurang, program akan mengeluarkan pesan kesalahan.



6.5. Command Attack

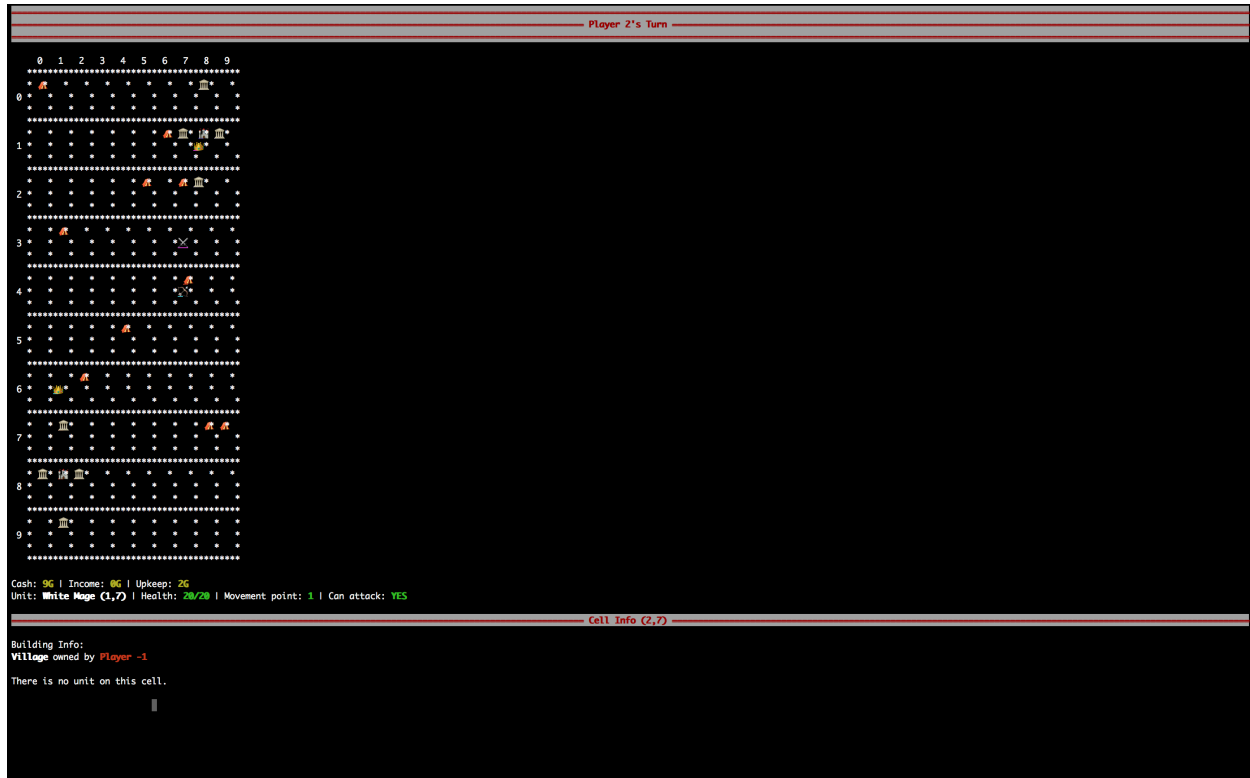
Pengetesan ini bertujuan untuk memilih command attack. Jika terdapat musuh yang bersisian dengan unit yang akan menyerang, maka program akan mengeluarkan pilihan unit musuh yang bersisian dengan unit yang akan menyerang. Jika tidak terdapat musuh yang bersisian dengan unit yang akan menyerang, maka program akan mengeluarkan pesan kesalahan. Setelah dipilih unit yang akan diserang, akan ditampilkan info tentang unit tersebut beserta kemungkinan keberhasilan serangan. Jika tipe serangan unit yang akan menyerang dan unit yang akan diserang sama, maka ada kemungkinan musuh akan menyerang balik. Jika pemain menyerang unit King lawan, maka unit King lawan akan menyerang balik tanpa memedulikan tipe serangan.





6.6. Command Info

Pengetesan ini bertujuan untuk memilih command info. Program akan meminta input berupa kolom dan baris sel yang ingin diketahui infonya. Program akan mengeluarkan output berupa informasi dari sel tersebut yang berupa info unit dan bangunan serta kepemilikannya.



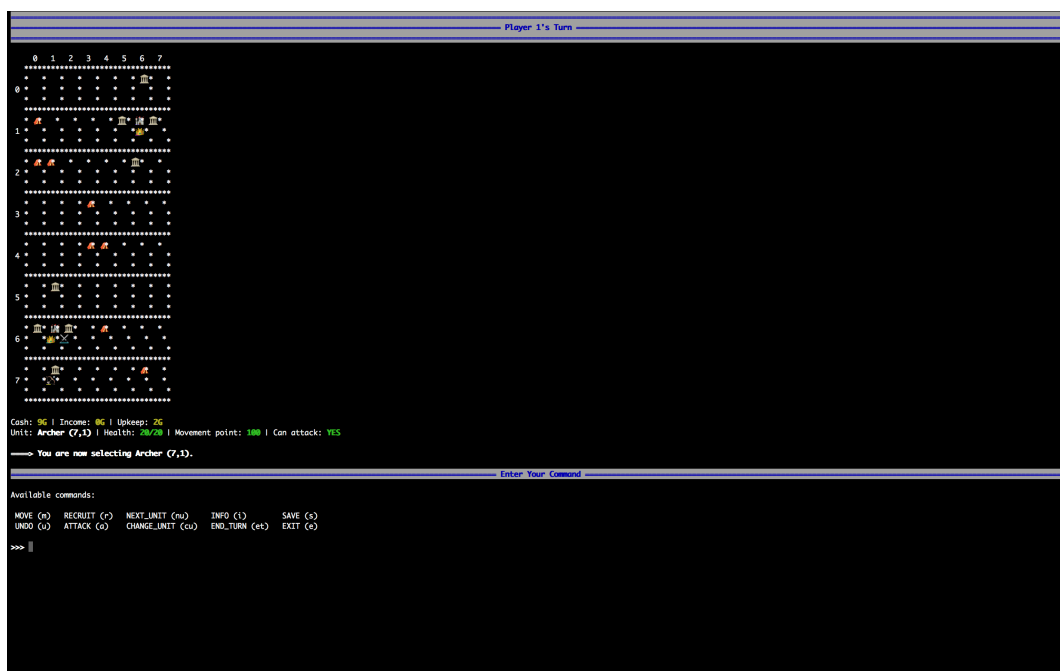
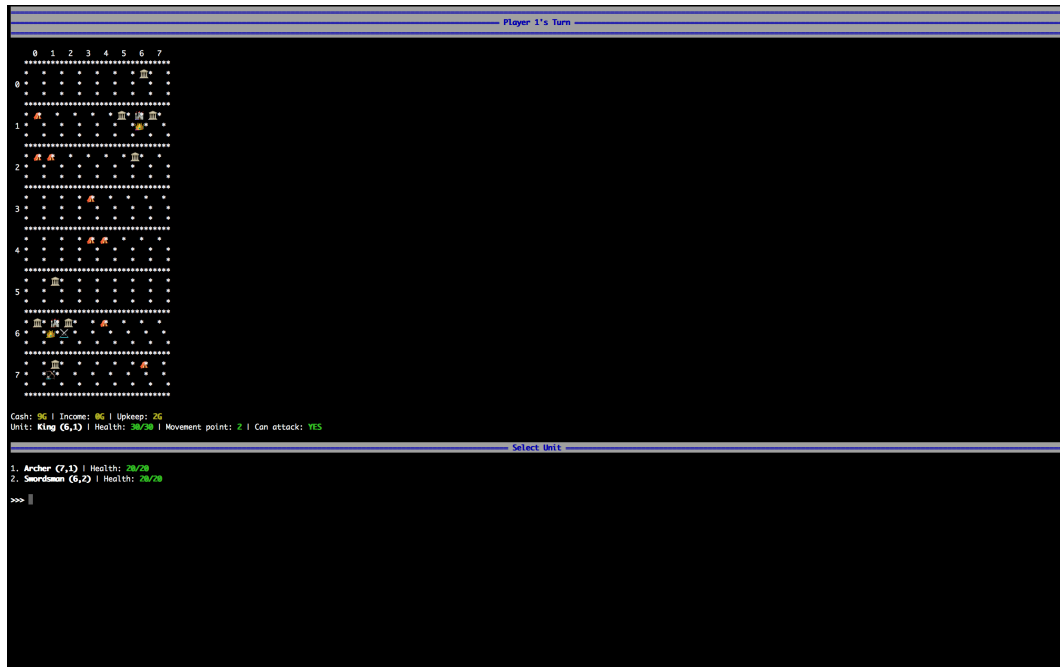
6.7. Kondisi Menang

Pengetesan ini bertujuan untuk memenangkan permainan. Kondisi ini tercapai ketika salah satu King terbunuh. Program akan mengeluarkan pesan kematian dari salah satu King dan damage yang diterima oleh King tersebut, kemudian, diberikan pesan kemenangan salah satu player. Program akan memunculkan pilihan untuk mengakhiri permainan atau memulai permainan baru.



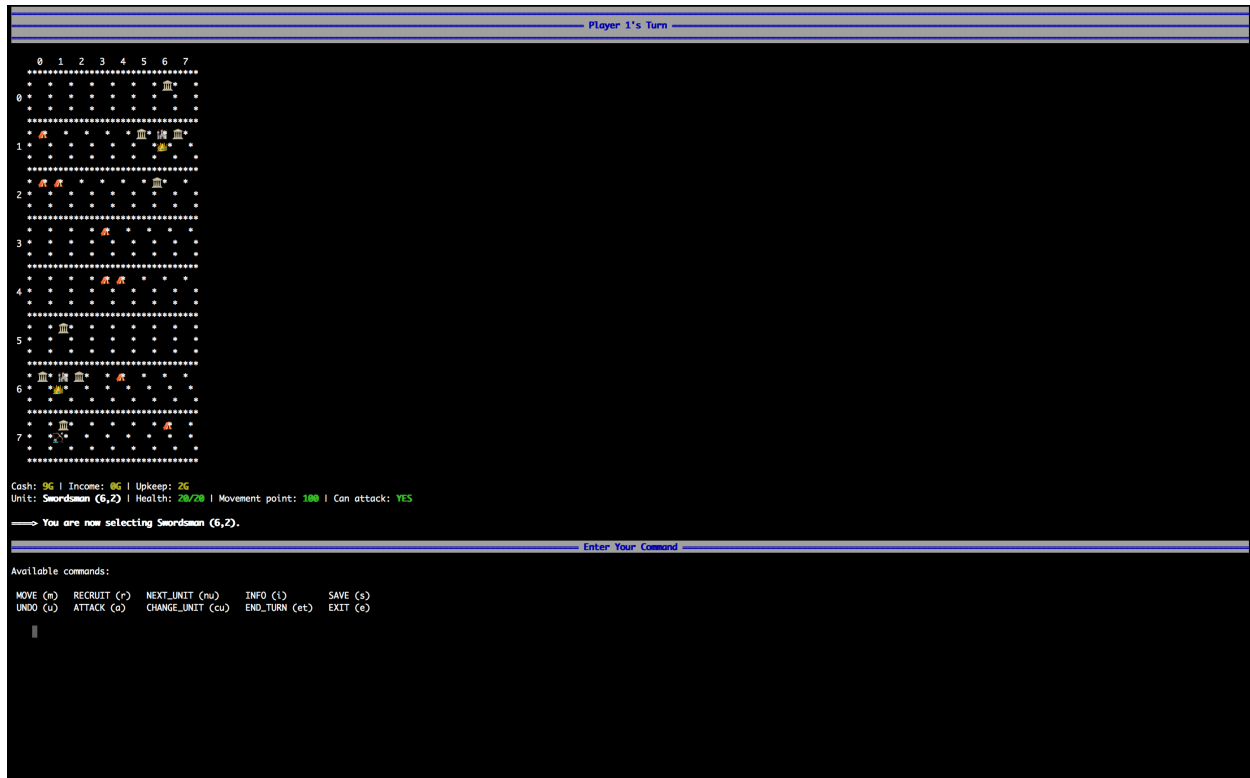
6.8. Command Change Unit

Pengetesan ini bertujuan untuk memilih command change unit. Program akan menampilkan seluruh unit pemain yang tersedia di map. Jika tidak ada pemain yang tersedia, maka program akan menampilkan pesan error.



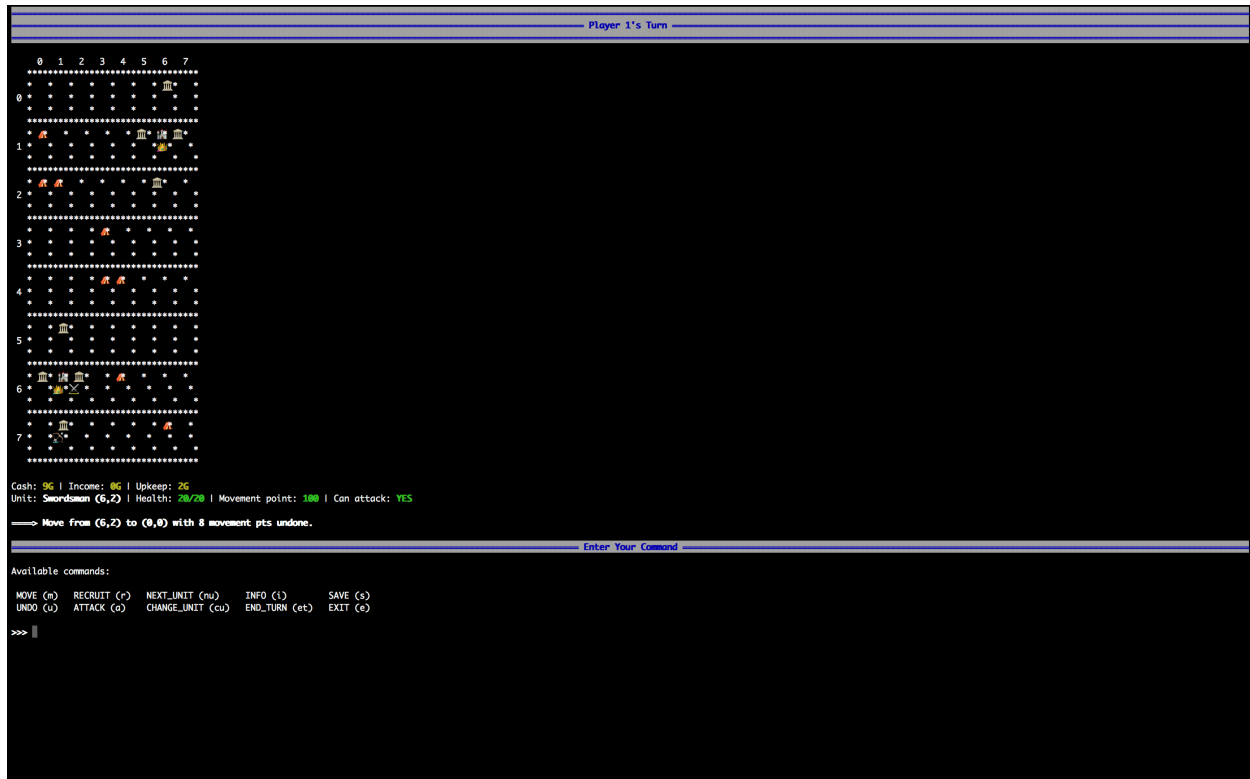
6.9. Next Unit

Pengetesan ini dilakukan untuk memilih command next unit. Program akan menampilkan unit selanjutnya yang masih aktif, yaitu masih memiliki movement points dan masih belum meny-erang.



6.10. Command Undo

Pengetesan ini dilakukan untuk memilih command undo. Undo bisa dijalankan ketika unit telah melakukan command move minimal sekali. Command undo tidak dapat digunakan jika unit telah menyerang unit lawan. Command undo akan menampilkan sel awal sebelum unit tersebut menggunakan command move.



6.11. Command Save

Pengetesan ini dilakukan untuk memilih command save. Command save digunakan untuk menyimpan data permainan di file eksternal dan dapat diload

```
===== Player 1's Turn =====
 0 1 2 3 4 5 6 7
0 + + + + + + +
1 + + + + + + +
2 + + + + + + +
3 + + + + + + +
4 + + + + + + +
5 + + + + + + +
6 + + + + + + +
7 + + + + + + +

Cash: 96 | Income: 96 | Upkeep: 26
Unit: Swordsman (6,2) | Health: 20/20 | Movement point: 100 | Can attack: YES
====> Game saved!

===== Enter Your Command =====

Available commands:
MOVE (m)  RECRUIT (r)  NEXT_UNIT (nu)  INFO (i)  SAVE (s)
UNDO (u)  ATTACK (a)  CHANGE_UNIT (cu)  END_TURN (et)  EXIT (e)

>>> |
```

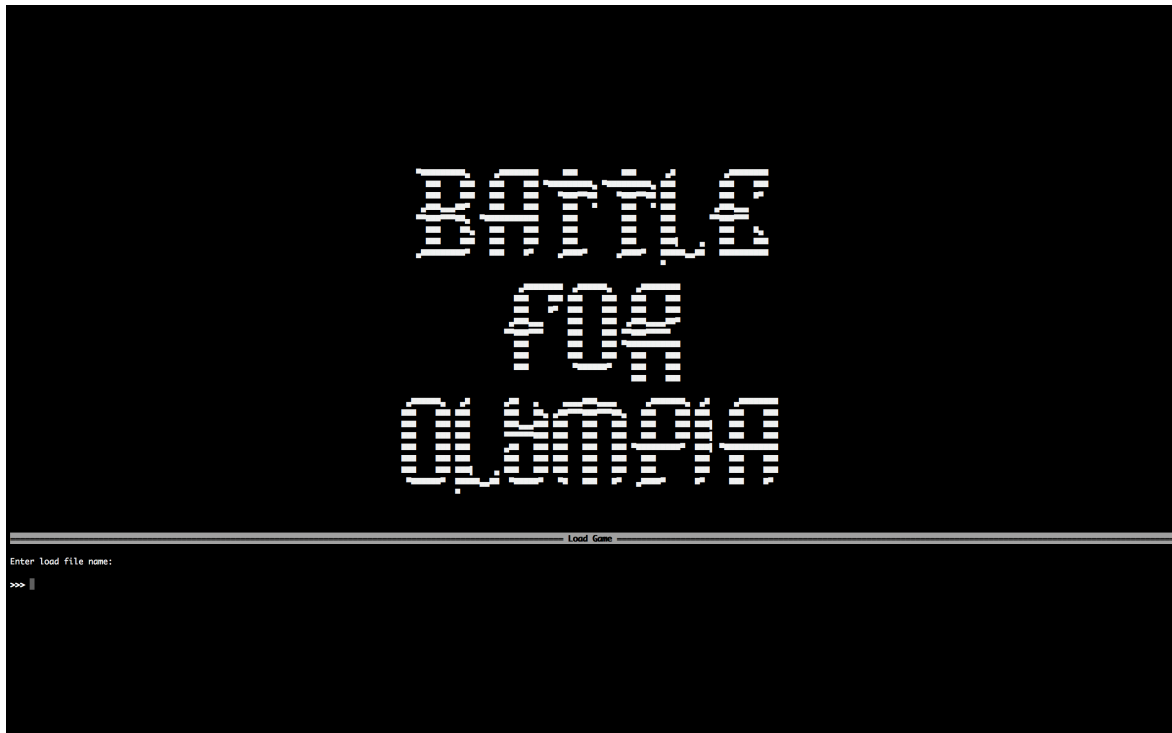
```
===== Player 1's Turn =====
 0 1 2 3 4 5 6 7
0 + + + + + + +
1 + + + + + + +
2 + + + + + + +
3 + + + + + + +
4 + + + + + + +
5 + + + + + + +
6 + + + + + + +
7 + + + + + + +

Cash: 96 | Income: 96 | Upkeep: 26
Unit: Swordsman (6,2) | Health: 20/20 | Movement point: 100 | Can attack: YES
===== Save Game =====

Enter save file name:
save.txt
```

6.12. Command Load

Pengetesan ini dilakukan untuk memilih command load. Command berguna untuk melanjutkan permainan yang sebelumnya sudah disimpan di file eksternal. Jika tidak terdapat file eksternal di folder yang sama, program akan mengeluarkan pesan kesalahan.



6.13. Command Exit

Pengetesan ini dilakukan untuk memilih command exit. Command exit berguna untuk keluar dari permainan. Program akan menampilkan opsi apakah pemain ingin menyimpan data ke file eksternal atau tidak.



7.Test Script

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
1	Attack	Testing kesesuaian program	Mencoba tes serangan ke musuh di berbagai posisi koordinat	Data Test 6.5.	Health pemain musuh berkurang	Health pemain musuh dapat berkurang
2	Move	Testing kesesuaian program	Mencoba menggerakkan unit ke sel tertentu	Data Test 6.4	Unit berpindah ke sel tersebut jika terjangkau	Unit berhasil pindah ke sel yang baru
3	Main Menu	Testing kesesuaian program	Mencoba UI awal permainan	Data Test 6.1	Muncul logo dengan benar disertai menu-menu yang bisa dipilih	Logo muncul dengan benar disertai menu-menu yang bisa dipilih
4	New Game	Testing kesesuaian UI	Mencoba inisialisasi awal permainan	Data Test 6.2	Game baru dapat terinisialisasi	Game terinisialisasi
5	Recruit	Testing perekrutan unit	Mencoba merekrut unit baru	Data Test 6.3	King dapat merekrut jika Tower kosong dan gold cukup	King dapat merekrut unit baru
6	Info	Testing command info	Menampilkan info dari sel tertentu	Data Test 6.6	Program dapat menampilkan informasi suatu sel	Program menampilkan informasi dari sel
7	Change Unit	Testing perubahan unit	Mengganti unit sekarang dengan unit lain	Data Test 6.8	Pemain dapat mengganti unit sekarang dengan unit lain	Pemain berhasil mengganti unit
8	Next Unit	Testing perubahan unit berikutnya	Mengganti unit sekarang dengan unit lain yang masih aktif secara random	Data Test 6.9	Pemain dapat mengganti ke unit lain yang masih aktif	Pemain berhasil mengganti ke unit lain

No.	Fitur yang Dites	Tujuan Testing	Langkah-Langkah Testing	Input Data Test	Hasil yang Diharapkan	Hasil yang Keluar
9	Undo	Testing stack move	Mengembalikan unit ke posisi awal sebelum melakukan move	Data Test 6.10	Pemain dapat meng-undo pergerakan yang telah terjadi	Pemain dapat meng-undo pergerakan
10	Save	Testing penyimpanan data	Menyimpan data permainan	Data Test 6.11	Pemain dapat menyimpan ke file eksternal	File eksternal baru terbuat dengan data dari permainan
11	Load	Testing load program	Meneruskan permainan dari file eksternal	Data Test 6.12	Pemain dapat meng-load game yang sebelumnya sudah disimpan di file eksternal	Berhasil meng-load game yang Sudah disimpan sebelumnya.
12	Exit	Testing keluar program	Mencoba untuk keluar dari permainan	Data Test 6.13	Program akan memunculkan menu exit	Program dapat memunculkan menu exit

8.Pembagian Kerja dalam Kelompok

NAMA	NIM	TUGAS
M. Sulthan A.	13516035	Mekanisme Game, Konstruktor Game, Laporan
Yusuf Rahmat P.	13516062	Mekanisme Game, Konstruktor Game
Abram Perdanaputra	13516083	Mekanisme Game, Testing, Konstruktor
Dandy Arif Rahman	13516086	Mekanisme Game , Laporan
Faza Fahleraz	13516095	User Interface, Mekanisme Game , Konstruktor
Ahmad Izzan	13516116	Konstruktor Game, Mekanisme Game

9.Lampiran

9.1. Deskripsi Tugas Besar 2

Battle for Olympia adalah permainan dengan *genre turn-based strategy* yang dimainkan dengan cara memasukkan perintah melalui *command line interface*. Tujuan utama dari permainan ini adalah menghancurkan kerajaan musuh dengan cara membunuh raja yang memimpin seluruh unit musuh. Seorang raja dapat merekrut beberapa pasukan, dan pasukan dapat dikerahkan untuk membunuh pasukan dan raja kerajaan musuh. Apabila raja mati, maka pemain selesai bermain dan seluruh pasukan yang tersisa ikut mati.

Permainan ini merupakan simplifikasi dari permainan Battle for Wesnoth. Permainan ini *open source* dan *setup* -nya berukuran sekitar 300 MB. Anda dapat mencoba memainkan permainan tersebut (minimal menyelesaikan tutorial) agar permainan tugas besar ini menjadi lebih intuitif.

Dari deskripsi game di atas, anda perlu membuat sebuah program yang dapat mensimulasikan *game* tersebut menggunakan bahasa C. Interaksi dari pengguna terhadap program tersebut perlu diimplementasikan menggunakan *Command Line Interface* (CLI) dengan *command* yang akan dijelaskan pada masing-masing fitur game.

9.2. Notulen Rapat

2/11/2017

Selasar Dingdong, pada malam ini kami membahas asbtraksi persoalan tugas besar.

6/11/2017

Apartement Izzan, pada malam ini kami telah membahas lebih lanjut tentang pembagian tugas dan abstraksi persoalan tugas besar, membuat list ADT yang diperlukan diantaranya *player*, *map*, *unit*, membuat beberapa fungsi yang diperlukan diantaranya *Mechanics*, *Recruit*, *Attack*, *PlayerTurn*, *ChangeActiveUnit*, membuat google drive untuk tugas besar.

12/11/2017

Apartement Faza, pada malam ini kami memutuskan untuk memindahkan fungsi-fungsi yang sebelumnya dalam bentuk ADT terpisah, kemudian dimasukkan ke *mechanics*, melanjutkan pengerjaan fungsi *recruit*, dan membuat UI untuk game tersebut.

13/11/2017

Appartement Izzan, pada malam ini kami melanjutkan menyelesaikan fungsi *NextActiveUnit*, *Attack*, *Random*, *ChangeActiveUnit*, melakukan percobaan ke program, *debugging*.

14/11/2017

9.3. Log Activity Anggota Kelompok

13516035 – M. Sulthan Adhipradhanna

13516062 – Yusuf Rahmat Pratama

13516083 – Abram Perdanaputra

13516086 – Dandy Arif Rahman

13516095 – Faza Fahleraz

13516116 – Ahmad Izzan