

Spesifikasi Tugas Besar

Battle for Olympia

Deskripsi Game

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.

Deskripsi Soal

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.

Game Mechanics

1. Main Menu

1. Pada awal program, pemain memilih untuk memulai permainan baru.
2. **(Bonus)** Selain memulai permainan baru, pemain dapat *load* game yang sudah disimpan.
3. Jika pengguna memilih permainan baru, maka pengguna harus memasukkan ukuran **peta** NxM, minimal 8x8.
4. Jumlah pemain adalah 2. Setiap pemain harus memiliki identitas **warna** unik untuk penggambaran peta pada terminal. Modul penulisan berwarna akan disediakan oleh asisten (baca bab Penggunaan Modul Berwarna)

2. Peta

- Pada permainan ini, terdapat peta Dunia Olympia yang memiliki ukuran NxM. Ukuran peta minimal 8x8. Lokasi pada peta mengikuti konvensi koordinat kartesian. Titik (i, j) merupakan sebuah **petak** pada baris ke-i kolom ke-j.
- Berikut adalah properti-properti yang dimiliki oleh sebuah petak peta
 - Koordinat (Constant)**
Koordinat memiliki absis dengan range nilai (0 .. N-1) dan ordinat dengan range nilai (0 .. M-1)
 - Jenis Petak (Constant)**
Jenis petak yang mungkin adalah **Tower (T)**, **Castle (C)**, **Village (V)**, dan **Normal**. Petak dengan jenis X harus menampilkan jenis petak secara eksplisit pada peta, kecuali petak Normal. Kegunaan jenis petak selain Normal akan dijelaskan di subbab-subbab selanjutnya.
 - Kepemilikan Petak (Variabel)**
Tower dan Castle memiliki kepemilikan konstan, sedangkan Village memiliki nilai kepemilikan Variabel (dapat direbut, dan dapat juga tidak dimiliki pemain siapapun). Petak Normal tidak memiliki kepemilikan. Kepemilikan kemudian akan ditampilkan dengan warna pada peta.
 - Unit pada Petak (Variabel)**
Petak juga dapat menyimpan **unit** yang sedang menempati petak peta tersebut. Satu petak hanya dapat ditempati oleh satu unit.
- Berikut adalah ilustrasi contoh petak peta.

```
*****
*  C  *
*  A  *
*    *
*****
```

Baris pertama menunjukkan **jenis peta** (kecuali petak Normal yang tidak perlu dituliskan), dan baris kedua menunjukkan **unit yang sedang menempati** petak. Petak pada ilustrasi merupakan petak dengan jenis **Castle** yang dimiliki pemain merah, dan ditempati oleh **unit Archer** milik pemain merah.

- Berikut adalah ilustrasi peta yang ditampilkan kepada pemain.

```

      0   1   2   3   4   5   6   7   8   9   10  11  12  13  14
*****
*  *  *  *  *  *  *  V  *  *  *  *  *  *  *
0 *  *  *  *  *  *  *  *  *  *  *  *  *  *
*  *  *  *  *  *  *  *  *  *  *  *  *  *
*****
*  *  C  *  *  *  *  *  *  *  *  *  *  C  *  T  *  C  *
```

```

1 *   *   * A *   *   *   *   *   *   *   *   *   *   *   *   * K *   *
  *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
  *****
  * C * T * C *   *   *   *   *   *   *   *   *   *   *   *   * C *   *
2 *   * K * A *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
  *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
  *****
  *   * C *   *   *   * V *   *   *   *   *   *   *   *   *   * V *   *
3 *   *   *   *   *   * S *   *   *   *   *   *   *   *   *   *   *   *
  *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
  *****

```

Nomor baris menyatakan absis, sedangkan nomor kolom menyatakan ordinat. Terdapat 2 pemain (biru dan merah), serta unit yang sedang digunakan (hijau).

3. Unit

1. Rincian properti yang dapat dimiliki oleh sebuah unit adalah sebagai berikut. Untuk nilai properti setiap unit, Anda definisikan sendiri.
 - a. Maximum Health (Integer, Constant)
Maximum Health merupakan properti darah maksimum dari suatu unit. Ketika direkrut pertama kali, unit memiliki properti Health sejumlah Maximum Health.
 - b. Health (Integer, Variabel)
Health merupakan properti dari unit, yang mana jika health memiliki jumlah ≤ 0 , maka unit tersebut dihapus dari permainan. Health dapat berkurang ketika terkena *damage*, atau bertambah ketika disembuhkan oleh Village atau White Mage.
 - c. Attack (Integer, Constant)
Attack merupakan properti dari unit mengenai *damage* serangan yang akan ia kenakan ke karakter musuh.
 - d. Maximum Movement Points (Integer, Constant)
Maximum Movement Points merupakan properti langkah maksimum yang dapat diambil oleh seorang unit pada suatu *turn*.
 - e. Movement Points (Integer, Variabel)
Movement Points merupakan properti langkah yang sudah diambil oleh seorang unit. Nilai awal Movement Points untuk setiap giliran adalah Maximum Movement Points. Untuk memindahkan unit dari sejauh 1 petak (misal dari $\langle x, y \rangle$ ke $\langle x+1, y \rangle$), properti ini akan berkurang sebanyak 1 nilai. Unit tidak dapat bergerak lagi jika nilai Movement Points sudah bernilai 0.
 - f. Tipe Serangan (Constant)
Tipe serangan dapat berupa **melee** dan **ranged**. Guna dari tipe serangan akan dijelaskan pada subbab Mekanisme Serangan
 - g. Kesempatan Serangan (Boolean, Constant)

Kesempatan serangan bernilai True jika unit belum menyerang pada suatu *turn*. Nilainya menjadi False setelah unit menyerang pada suatu *turn*.

- h. Lokasi (Koordinat, Variabel)
Lokasi merupakan lokasi petak yang saat ini ditempati oleh unit.
 - i. Harga (Integer, Constant)
Harga dari unit ketika akan direkrut.
2. Rincian unit-unit yang mungkin ada pada permainan ini adalah sebagai berikut. Silahkan definisikan sendiri nilai properti-properti yang belum didefinisikan (Maximum Health, Maximum Movement Points, dst.)
- a. King
King merupakan unit yang pertama kali dimiliki pemain. Jika King seorang pemain mati, maka pemain tersebut berhenti dari game dan seluruh unitnya mati. King memiliki tipe serangan **melee**. King direpresentasikan dengan simbol **K** pada peta.
 - b. Archer
Archer memiliki tipe serangan **ranged**. Archer direpresentasikan dengan simbol **A** pada peta.
 - c. Swordsman
Archer memiliki tipe serangan **melee**. Swordsman direpresentasikan dengan simbol **S** pada peta.
 - d. White Mage (**Bonus**)
White Mage memiliki tipe serangan **melee**. Selain memiliki tipe serangan melee, untuk setiap awal *turn*, unit-unit yang berada di petak *adjacent* dengan petak White Mage akan di-*heal* secara otomatis. White Mage direpresentasikan dengan simbol **W** pada peta.

4. State Awal Permainan

1. Untuk setiap pemain, masing-masing pemain akan diinisiasi dengan petak-petak di bawah ini yang akan disebut **bangunan kerajaan**. Bangunan kerajaan terdiri atas 1 petak Tower, 1 unit King yang berada pada petak Tower, dan 4 petak Castle yang *adjacent* dengan petak Tower.

```
*****
*   * C *   *
*   *   *   *
*   *   *   *
*****
* C * T * C *
*   * K *   *
*   *   *   *
*****
*   * C *   *
*   *   *   *
```

* * * *

Setiap pemain akan memiliki bangunan kerajaan yang sama, namun dengan **warna berbeda**. Pemain 1 pasti memiliki bangunan kerajaan di ujung kiri-bawah peta, dan pemain 2 pasti memiliki bangunan kerajaan di ujung kanan-atas peta.

3. Selain Tower dan Castle, peta akan diinisiasi dengan beberapa **Village** yang lokasinya disebarkan di peta secara random, asalkan tidak bertabrakan dengan Tower dan Castle. Village pada awalnya **belum memiliki kepemilikan**. Jumlah Village Anda tentukan sendiri.

Berikut adalah contoh peta sesuai dengan *state* awal permainan dengan jumlah pemain 2 dan terdapat 3 Village.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

	*	*	*	*	*	*	V	*	*	*	*	*	*	*	*
0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

	*	*	C	*	*	*	*	*	*	*	*	*	C	*	T
1	*	*	*	*	*	*	*	*	*	*	*	*	*	K	*
	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

	*	C	*	T	*	C	*	*	*	*	*	*	*	C	*
2	*	*	K	*	*	*	*	*	*	*	*	*	*	*	*
	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

	*	*	C	*	*	*	*	*	V	*	*	*	*	*	V
3	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

4. Setiap pemain memiliki properti sebagai berikut.
 - a. Gold (Integer, Variabel)
Gold merupakan uang yang dimiliki oleh pemain. Gold digunakan untuk merekrut **unit**.
 - b. List of Units
Pemain menyimpan daftar **unit** yang telah direkrut, termasuk **King**.
 - c. List of Villages
Pemain menyimpan daftar Village mana saja yang telah diakuisisi.
 - d. Income (Integer, Variabel)
Income merupakan pemasukan pemain setiap *turn*. Income hanya bisa didapatkan dari kepemilikan Village. Setiap *turn*, jumlah Gold yang dimiliki pemain akan bertambah sebanyak Income.

- e. Upkeep (Integer, Variabel)
Upkeep merupakan pengeluaran suatu pemain untuk setiap turn, di mana nilainya sama dengan jumlah unit yang pemain miliki di peta, **kecuali King**. Setiap *turn*, jumlah Gold yang dimiliki pemain akan berkurang sebanyak Upkeep.
- f. Warna
Warna merupakan identitas pemain untuk membedakan dengan pemain lainnya ketika digambarkan di peta.

4. Mekanisme Bertarung

1. Unit X dapat menyerang unit Y jika unit X dan Y berada pada **petak bersebelahan**. Misalkan lokasi X adalah (i1, j1) dan lokasi Y adalah (i2, j2), maka unit X dapat menyerang unit Y jika persamaan di bawah dipenuhi.
$$|i1 - i2| + |j1 - j2| = 0$$
2. Jika unit X menyerang unit Y, maka damage yang didapatkan oleh Y adalah
$$Y.HEALTH = Y.HEALTH - X.ATK$$
3. Ketika unit X menyerang unit Y, unit Y juga dapat melakukan **pembalasan/retaliation** berdasarkan **properti tipe serangan**. Berikut adalah mekanisme-mekanisme yang mungkin terjadi.
 - a. Jika X dan Y memiliki tipe serangan sama (misalnya sama-sama ranged), maka selain Y menerima damage dari X, X juga menerima damage dari Y (**retaliation**) sesuai dengan Attack Y.
 - b. Jika X dan Y memiliki tipe serangan yang berbeda, maka Y menerima damage tanpa membalas serangan (**retaliation**).
 - c. Jika Y adalah King, maka X akan selalu mendapatkan balasan.

5. Village

1. Village secara singkat adalah bangunan yang jika dimiliki pemain dapat memberikan sejumlah Gold setiap *turn*.
2. Village dapat diakuisisi oleh pemain dan pada peta. Cara mengakuisisi Village adalah dengan menggerakkan unit apapun ke petak Village. Warna Village akan menyesuaikan dengan warna pemain yang memilikinya. Apabila sebuah unit digerakkan ke Village, **movement point yang dimiliki unit akan menjadi 0**.
3. Setelah Village diakuisisi, unit dapat bergerak meninggalkan Village dan kepemilikan Village tersebut tidak berubah.

6. Commands

Setelah state awal permainan diinisiasi, maka program akan selalu menunggu masukan dari pengguna. Berikut adalah masukan *command* yang dapat diberikan oleh pengguna.

1. MOVE

- a. *Command* ini akan menampilkan peta beserta langkah yang memungkinkan yang dapat diambil oleh sebuah unit. Lihat subbab Contoh Tampilan Program untuk ilustrasi.
 - b. Langkah yang dapat diambil akan bergantung kepada jumlah langkah yang dapat dilakukan oleh unit. Jika sebuah unit memiliki 2 *movement point*, maka unit dapat bergerak maksimal 2 langkah ke atas, ke kiri, ke kanan, atau ke bawah. Tentunya, unit juga dapat bergerak hanya 1 langkah.
 - c. Unit tidak dapat bergerak diagonal (kecuali Anda ingin mengerjakan **bonus** yang dirincikan di bab Bonus).
 - d. MOVE dapat memindahkan anggota ke petak mana saja (termasuk peta Castle Tower musuh), kecuali petak tersebut ditempati oleh unit.
 - e. Suatu unit dapat bergerak “menembus” unit kerajaan yang sama (walaupun tidak dapat menempati petak yang sedang ditempati unit lain), namun tidak dapat bergerak “menembus” unit pemain lain.
2. UNDO
- a. *Command* ini dipanggil agar state suatu unit kembali ke state sebelumnya. State yang dicatat hanyalah state **perpindahan** lokasi suatu unit.
 - b. Sebagai contoh, misalkan Anda memiliki unit Archer di posisi (1,1). Dalam 1 *turn*, Anda memindahkan Archer tersebut 2x ke (1,2) dan (1,3). Jika command UNDO dilakukan, maka Archer akan kembali ke posisi (1,2), dan *movement points*nya bertambah 1. Jika Anda UNDO sekali lagi, maka Archer akan kembali ke posisi (1,1) dan *movement points*nya bertambah 1. Jika Anda UNDO sekali lagi, tampilkan pesan kegagalan.
 - c. Apabila Anda melakukan command selain MOVE, MAP, dan INFO (misalnya Anda melakukan ATTACK), maka state seluruh MOVE suatu unit dihapus sehingga sudah tidak dapat melakukan UNDO kembali.
3. CHANGE_UNIT
- a. *Command* ini akan menampilkan daftar unit pemain, rinci mulai dari jenis unit, lokasi, *movement point*, kesempatan menyerang, dsb. Dari daftar yang ditampilkan, pemain dapat memilih unit yang ingin digerakkan.
4. RECRUIT
- a. Yang dapat melakukan RECRUIT hanyalah King, dan King harus berada pada petak Tower. King suatu pemain boleh merekrut pasukan baru apabila terdapat castle yang tidak ditempati unit. Apabila Castle pemain penuh, maka raja tidak dapat merekrut pasukan baru.
 - b. *Command* ini adalah menampilkan interaksi perekrutan pasukan. Pertama, pengguna akan memilih Castle mana yang akan diisi pasukan. Lalu, akan ditampilkan pasukan-pasukan yang dapat direkrut, termasuk harga dan rincian lainnya. Ketika dibeli, Gold pemain dikurangi jumlahnya sesuai harga unit, dan unit diletakkan di Castle yang sudah dipilih
 - c. Perekrutan unit tidak membuat King kehilangan *movement point* dan kesempatan untuk menyerang.
 - d. Unit yang baru saja direkrut **memiliki movement point 0**.

5. ATTACK

- a. *Command* ini akan menampilkan unit-unit musuh apa saja yang dapat diserang. Unit-unit musuh yang dapat diserang adalah unit-unit yang *adjacent* di petak atas, kanan, kiri, atau bawah dari unit saat ini.
- b. Suatu unit hanya dapat menyerang sebanyak 1x setiap turn (properti kesempatan menyerang menjadi False setelah menyerang).
- c. Setelah melakukan *command* ATTACK, nilai *movement point* unit penyerang nilainya diubah menjadi 0, dan

6. MAP

Command ini akan menampilkan peta secara keseluruhan.

7. INFO

Command ini akan memberikan informasi mengenai suatu petak, baik jenis dari petak tersebut (contoh: petak biasa, petak Tower pemain merah, dll.) maupun unit yang sedang berada di petak tersebut (contoh: unit Archer, health 50, dll.).

8. END_TURN

Command ini digunakan untuk mengakhiri giliran seorang pemain. Giliran kemudian akan berpindah ke pemain selanjutnya.

9. SAVE (**Bonus**)

Command ini digunakan untuk menyimpan *state* permainan saat ini agar dapat dilanjutkan kemudian. Metode interaksi bebas. Anda dapat merancang struktur penyimpanan data dalam *file* yang dapat disimpan dan kemudian dimuat kembali. Format data yang Anda simpan dibebaskan.

10. EXIT

Command ini digunakan untuk keluar dari program.

7. Kondisi Akhir

1. Permainan berakhir apabila hanya ada satu raja yang tersisa. Pemain yang memiliki raja tersisa adalah pemenang permainan.

Contoh Tampilan Program

Berikut adalah contoh state dari permainan yang akan digunakan untuk beberapa command yang penting. Misalkan player 1 adalah player dengan warna merah. Perhatikan bahwa **Anda tidak perlu mengimplementasikan output sama persis dengan contoh.**

```

      0   1   2   3   4   5   6   7   8   9   10  11  12  13  14
*****
*   *   *   *   *   *   V   *   *   *   *   *   *   *   *
0 *   *   *   *   *   *   *   *   *   *   *   *   *   *
*   *   *   *   *   *   *   *   *   *   *   *   *   *
*****
*   *   C   *   *   *   *   *   *   *   *   *   *   C   *   T   *   C   *
1 *   *   *   A   *   *   *   *   *   *   *   *   *   *   *   K   *   *
*   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
*****
*   C   *   T   *   C   *   *   *   *   *   *   *   *   *   *   C   *   *
2 *   *   K   *   A   *   *   *   *   *   *   *   *   *   *   *   *   *
*   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
*****
*   *   C   *   *   *   *   V   *   *   *   *   *   *   *   *   V   *   *
3 *   *   *   *   *   *   *   S   *   *   *   *   *   *   *   *   *   *
*   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
*****

```

1. Command RECRUIT

Player 1's Turn

Cash: 50G | Income: 2G | Upkeep: 1G

Unit: King(2,1) | Movement Point: 2 | Can Attack: Yes

Your input: RECRUIT

Enter coordinate of empty castle: 2 2

That castle is occupied!

Enter coordinate of empty castle: 2 0

=== List of Recruits ===

1. Archer | Health 20 | ATK 4 | DEF 1 | 5G

2. Swordsman | Health 20 | ATK 3 | DEF 2 | 4G

Enter no. of unit you want to recruit: 1

You have recruited an archer!

Player 1's Turn

Cash: 45G | Income: 2G | Upkeep: 1G

Unit: King(2,1) | Movement Point: 2 | Can Attack: Yes

Your Input: MAP

```

      0   1   2   3   4   5   6   7   8   9   10  11  12  13  14
*****
*   *   *   *   *   * V *   *   *   *   *   *   *   *   *
0 *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
*   *   *   *   *   *   *   *   *   *   *   *   *   *   *
*****
*   * C *   *   *   *   *   *   *   *   *   *   * C * T * C *
1 *   *   * A *   *   *   *   *   *   *   *   *   *   * K *   *
*   *   *   *   *   *   *   *   *   *   *   *   *   *   *
*****
* C * T * C *   *   *   *   *   *   *   *   *   *   * C *   *
2 * A * K * A *   *   *   *   *   *   *   *   *   *   *   *   *
*   *   *   *   *   *   *   *   *   *   *   *   *   *   *
*****
*   * C *   *   *   * V *   *   *   *   *   *   *   * V *   *
3 *   *   *   *   *   * S *   *   *   *   *   *   *   *   *   *
*   *   *   *   *   *   *   *   *   *   *   *   *   *   *
*****

```

2. Command CHANGE_UNIT

Player 1's Turn

Cash: 50G | Income: 2G | Upkeep: 1G

Unit: Archer (2,2) | Health 3/20 | Movement Point: 2 | Can Attack: Yes

Your input: CHANGE_UNIT

== List of Units ==

1. King (2,1) | Health 20

2. Swordsman (3,5) | Health 20

Please enter the no. of unit you want to select: 1

You are now selecting King

Player 1's Turn

Cash: 50G | Income: 2G | Upkeep: 1G

Unit: King(2,1) | Health 20/20 | Movement Point: 2 | Can Attack: Yes

Your Input: MAP

```

      0   1   2   3   4   5   6   7   8   9   10  11  12  13  14
*****
*   *   *   *   *   * V *   *   *   *   *   *   *   *   *
0 *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
*   *   *   *   *   *   *   *   *   *   *   *   *   *   *
*****
*   * C *   *   *   *   *   *   *   *   *   *   * C * T * C *
1 *   *   * A *   *   *   *   *   *   *   *   *   *   * K *   *

```

```

* * * * *
*****
* C * T * C * * * * * * * * * * C * *
2 * A * K * A * * * * * * * * * * * *
* * * * *
*****
* * C * * * * V * * * * * * * V * *
3 * * * * * S * * * * * * * * * *
* * * * *
*****

```

3. Command MOVE

Pada command ini, misalkan petak dengan '#' merupakan petak yang dapat dijangkau oleh unit tersebut.

Player 1's Turn

Cash: 50G | Income: 2G | Upkeep: 1G

Unit: Archer (2,2) | Health 3/20 | Movement Point: 2 | Can Attack:

Yes

Your input: MOVE

```

    0   1   2   3   4   5   6   7   8   9   10  11  12  13  14
*****
* * * * * V * * * * * * * * * C * *
0 * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * *
*****
* * C * * * * * * * * * * C * T * C *
1 * * * A * * * * * * * * * * K * *
* * * * * * * * * * * * * * * *
*****
* C * T * C * * * * * * * * * * C * *
2 * # * K * A * # * # * * * * * * * *
* * * * * * * * * * * * * * * *
*****
* * C * * * * V * * * * * * * V * *
3 * * * # * * * S * * * * * * * * *
* * * * * * * * * * * * * * * *
*****

```

Please enter cell's coordinate x y: 5 3

You can't move there

Please enter cell's coordinate x y: 2 4

You have successfully moved to (2, 4)

Player 1's Turn

Cash: 50G | Income: 2G | Upkeep: 1G

Unit: Archer (2,4) | Health 3/20 | Movement Point: 2 | Can Attack:
Yes

Your Input: MAP

```

      0   1   2   3   4   5   6   7   8   9   10  11  12  13  14
*****
*   *   *   *   *   *   V   *   *   *   *   *   *   *   *
0 *   *   *   *   *   *   *   *   *   *   *   *   *   *
*   *   *   *   *   *   *   *   *   *   *   *   *   *
*****
*   *   C   *   *   *   *   *   *   *   *   *   *   *   C   *   T   *   C   *
1 *   *   *   A   *   *   *   *   *   *   *   *   *   *   *   K   *   *
*   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
*****
*   C   *   T   *   C   *   *   *   *   *   *   *   *   *   *   *   C   *   *
2 *   *   K   *   *   *   A   *   *   *   *   *   *   *   *   *   *   *
*   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
*****
*   *   C   *   *   *   *   V   *   *   *   *   *   *   *   *   *   V   *   *
3 *   *   *   *   *   *   *   S   *   *   *   *   *   *   *   *   *   *   *
*   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
*****
```

4. Command INFO

Player 1's Turn

Cash: 50G | Income: 2G | Upkeep: 1G

Unit: King(2,1) | Health 20/20 | Movement Point: 2 | Can Attack: Yes

Your input: INFO

Enter the coordinate of the cell: 3 5

== Cell Info ==

Village

Owned by Player 1

== Unit Info ==

Swordsman

Owned by Player 1

Health 7/20 | ATK 3 | DEF 2

5. Command ATTACK

Player 1's Turn

Cash: 50G | Income: 2G | Upkeep: 1G

Unit: Archer (2,2) | Health 3/20 | Movement Point: 2 | Can Attack:
Yes

Your input: **ATTACK**

Please select enemy you want to attack:

1. Archer (2,1) | Health 7/20 (Retaliates)

Select enemy you want to attack: **1**

Enemy's Archer is damaged by 4.

Enemy's Archer retaliates.

Your Archer is damaged by 3.

You Archer is dead :(

Bonus

1. Terdapat properti probabilitas terkena serangan pada suatu unit. Misalkan unit X menyerang unit Y, dan unit Y memiliki probabilitas terkena serangan 20%. Maka, kemungkinan Y terkena serangan hanya 20%. Tampilkan pada program bahwa serangan tersebut terkena atau tidak. Hal yang sama juga berlaku ketika musuh melakukan *retaliation*.
2. Terdapat command NEXT_UNIT. Tidak seperti CHANGE_UNIT, NEXT_UNIT langsung memilih otomatis unit pada daftar unit pemain yang masih memiliki *movement points* atau kesempatan menyerang. Jika tidak ada unit lain yang masih memiliki *movement points* atau kesempatan menyerang, NEXT_UNIT menampilkan pesan kegagalan.
3. Unit dapat bergerak secara diagonal. Contohnya, jika *movement point* archer nilainya 2 dan saat ini berada pada (1,1), peta dapat menampilkan kemungkinan unit pindah ke (2,2). Konsumsi *movement point* tetap 2 (bergerak ke kanan, kemudian ke atas). Keberadaan pasukan musuh tetap harus dipertimbangkan.
4. Rekrutmen unit White Mage
5. Save dan load game

Penggunaan Modul Print Berwarna

Karena setiap pemain diharuskan menggunakan warna berbeda, asisten akan menyediakan sebuah modul untuk melakukan *print* berwarna pada terminal. Modul ini dapat digunakan pada Windows Powershell dan terminal UNIX. Silahkan unduh **pcolor.c** dan **pcolor.h** pada Olympia, serta lihatlah contoh pada pcolor.h untuk penggunaannya. Anda tentu dapat memodifikasi modul tersebut jika diperlukan.

Daftar Abstract Data Type (ADT) yang Digunakan

Dapat pula menggunakan ADT lain, cantumkan analisis alasan penggunaan ADT tersebut pada laporan.

1. ADT Jam

Digunakan untuk menyimpan waktu ketika melakukan save.

2. ADT Point

ADT ini digunakan untuk menunjukkan posisi unit pada setiap waktu.

3. ADT Matriks

ADT ini digunakan sebagai representasi peta.

4. ADT Mesin Karakter + Mesin Kata

ADT ini digunakan untuk:

- membaca informasi dari properti masing-masing unit dari file eksternal
- membaca *command* dari interaksi user terhadap program, dan
- membaca *state* dari *game* yang sudah pernah disimpan.

5. ADT Queue

ADT ini digunakan untuk menentukan giliran dari masing-masing pemain.

6. ADT Stack

ADT ini digunakan untuk mendukung mekanisme command UNDO.

7. ADT List Linier (Dummy element/circular/double pointer/rekursif)

ADT ini akan digunakan untuk merepresentasikan daftar pemain, daftar village, serta daftar unit yang dimiliki oleh seorang pemain.

8. ADT Lain

ADT ini dibuat dan digunakan untuk abstraksi beberapa hal seperti ADT Unit. Mahasiswa dipersilahkan untuk mendefinisikan sendiri.