**LAPORAN HASIL TUGAS BESAR**

**A PROGRAMMER IS BORN IN NEW ISEKAI (GENSHIN SEKAI)**

**ThryLos: The Prologue**

**DIAJUKAN UNTUK MEMENUHI TUGAS MATA KULIAH**

**IF2121 – Logika Komputasional**

**SEMESTER I TAHUN 2020/2021**

**Disusun oleh :**

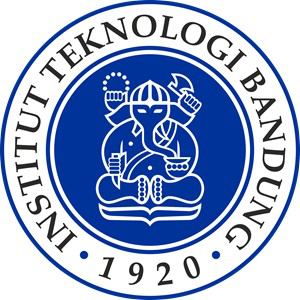
**Kelompok 07 / K-01**

**Michael Philip Gunadi 13519121**

**Maximillian Lukman 13519153**

**David Owen Adiwiguna 13519169**

**Richard Rivaldo 13519185**



**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**BANDUNG**

**2020**

Pada permainan ini, umumnya *commands* yang bisa digunakan berada di file **main.pl.** Hal ini dimaksudkan untuk mempermudah pengelompokkan *commands* yang bisa dipanggil oleh pemain dan juga yang tidak. Pemanggilan *commands* ini dilakukan dengan menggunakan pemanggilan dari gabungan predikat-predikat yang terdapat dari file lainnya. Berikut merupakan file-file tersebut.

1. **character.pl**

Di dalam file ini memuat fakta-fakta **isClass(Class)** untuk setiap **Class** yang didefinisikan di dalam permainan. Untuk setiap *class* tersebut, kemudian didefinisikan predikat **createClass(Username)** untuk melakukan pembuatan karakter dan *stats* awal yang akan diberikan apabila pemain memilih atau mendapatkan sebuah kelas tertentu. *Stats* yang ada dalam permainan ini berupa *health, attack, defense, magic,* dan juga *speed*. Pemain juga diinisiasi dengan sejumlah *gold* sebagai *currency* di dalam game dan juga level yang dimulai dari 1 dengan jumlah *experience point* awal adalah 0.

Pada saat pemain melakukan pemilihan kelas, akan ditampilkan 5 kelas biasa yang memiliki *stats* reguler. Untuk setiap kelas tersebut akan berpasangan dengan sebuah ***hidden class***yang bisa didapatdengan menginput pilihan `0` dan meng-*invoke* predikat **surprise(Username)** pada saat pemilihan. Pemain tidak bisa memilih *hidden class* dan setelah dilakukan pengacakan, pemain akan langsung di-*assign* ke dalam kelas tersebut.

Selain itu, ada predikat-predikat yang digunakan untuk mengecek *stats*, *equipments*, dan *inventory* dari pemain. Akan tetapi, predikat ini menjadi penyusun dari predikat di file lain yang nantinya akan menjadi *command* yang bisa diinput oleh pemain. Untuk setiap kelas juga diberikan sebuah **ASCII Art** dari kelas yang dipilih atau didapat.

1. **inventory.pl**

Di dalam inventory.pl didefinisikan beberapa predikat yang akan digunakan untuk mencatat jumlah tiap item yang ada di dalam *inventory* kita (**countItem(Item, Count)**), serta yang digunakan untuk mencatat penggunaan *space* dari *inventory* tersebut (**inventory(X)** dan **usedSpace(X)**). Ada juga predikat **IsInInvent(Items)** yang digunakan untuk mengecek apakah suatu item ada di dalam *inventory* kita dengan mengecek jumlahnya di dalam inventori tersebut.

Di dalam file ini juga akan didefinisikan predikat **initInventClass** yang dipanggil dalam createClass sebelumnya, untuk menginisiasi inventori dari setiap pasang kelas dengan masing-masing 5 buah *health potion* serta senjata awalnya masing-masing. Adapun predikat **SaveItem(Items)** digunakan sebagai fungsi perantara yang akan digunakan untuk memasukkan item ke dalam inventori serta melakukan *update* penggunaan ruang dalam inventori.

inventory.pl ini juga digunakan untuk mendefinisikan **WikiLog** yang fungsinya akan menggantikan fungsi *help* yang ada dalam spesifikasi. Serupa dengan *help*, WikiLog juga akan memberikan petunjuk mengenai mekanisme permainan serta *commands* yang bisa digunakan dalam game. Pemain dapat memasukkan navigasi ke menu *help* yang diinginkan, seperti cerita permainan, ataupun mekanisme permainan seperti *inventory, equipments, potions, map, store, enemy, quest,* dan juga *battle*.

1. **items.pl**

Di items.pl dideklarasikan fakta-fakta yang mengelompokkan *equipments* berdasarkan jenisnya (*weapon*, *armor*, dan *accessory*) serta *grade* dari tiap item (diurutkan dari yang terkecil: *Ghost, Demon, Mamon, Lucifer, dan God*). Di dalam permainan juga didefinisikan 5 jenis *potion* untuk setiap *stats* yang ada. Selain itu, juga ada pendefinisian predikat yang digunakan untuk memakai dan melepas *equipment* tersebut, serta meminum *potion*.

Pemain dapat melakukan pengecekan terhadap nama serta restriksi kelas dari setiap *equipments*. Dengan menggunakan *equipments* yang ada di dalam permainan, maka pemain akan mendapatkan tambahan *stats*, dan sebaliknya apabila dilepas. Untuk setiap kelas sendiri, tiap item yang masuk ke dalam *weapon* hanya akan menambah sebuah atribut saja sehingga atribut-atribut lain hanya bisa ditingkatkan melalui *potion* atau dengan meningkatkan level pemain.

Penggunaan *potion* sendiri akan menambah *stats* dari pemain secara permanen dan *stats* yang akan ditambahkan berkorespondensi dengan nama dari *potion* yang digunakan. Selain itu, pemain bisa membuang item yang berada di inventori dengan jumlah yang diinginkan.

1. **leveling.pl**

Sistem *leveling* dalam permainan ini dilakukan dengan menggunakan predikat perantara yang digunakan untuk menambah *experience point*, yaitu **addExp(User, X)**. Predikat ini akan menambah *experience* dari User sebanyak X poin. Predikat ini digunakan sekaligus untuk mendeteksi serta mengecek apabila pemain berhasil naik ke level berikutnya atau tidak.

Jika pemain berhasil naik level, maka akan dipanggil predikat lain untuk menambah *stats* dari pemain sesuai dengan level yang berhasil dicapai. Dalam hal ini, semua *stats* dari pemain akan meningkat sesuai dengan *growth rate* masing-masing *stats*. Jika pemain sudah mencapai level maksimum, maka *stats* pemain tidak akan mengalami penambahan lagi.

1. **map.pl**

Sistem map yang didefinisikan dalam file ini dilakukan dengan mendefinisikan koordinat dari tempat-tempat yang ada dalam peta. Koordinat ini didefinisikan dengan menggunakan generator secara rekursif sehingga dihasilkan kombinasi koordinat dari (0,0) sampai dengan (15,15). Di dalam peta didefinisikan predikat untuk menentukan apakah koordinat tertentu adalah dinding atau bukan. Di dalam peta sendiri juga didefinisikan dinding yang ada di dalam area peta sehingga terdapat dua jenis dinding.

Untuk mencetak peta, akan dilakukan iterasi secara rekursif dengan menggunakan predikat **map(StartX, StartY)**, dengan StartX dan StartY adalah titik awal predikat ketika melakukan pencetakan peta. Tiap lokasi khusus di peta akan ditandai dengan penggunaan predikat seperti **store(X, Y)**, **quest(X, Y)**, serta **isTaken(X, Y)** untuk menandakan apabila koordinat tersebut sudah dimiliki oleh lokasi tersebut.

Di dalam file ini juga akan didefinisikan *commands* gerakan yang bisa digunakan oleh pemain untuk berpindah tempat di dalam peta. Dalam hal ini, juga akan didefinisikan predikat untuk mengecek apabila gerakan yang dilakukan oleh pemain itu valid atau tidak. Umumnya, gerakan yang valid adalah gerakan pemain yang posisi barunya setelah berpindah tempat tidak merupakan dinding.

Adapun sisa predikat yang ada dalam file ini merupakan pendefinisian bonus yang akan dijelaskan di bagian lain.

1. **store.pl**

Dalam file store.pl terdapat mekanisme pembelian barang, juga sistem gacha. Terdapat predikat **visitStore** untuk mengunjungi store dalam game. Ketika pemain pertama kali masuk ke store, mereka akan disajikan dua buah pilihan berupa opsi untuk membeli potion, ataupun opsi untuk membeli tiket gacha untuk mendapatkan item dengan berbagai macam grade. Potion dapat dibeli dengan harga tetap dengan memilih opsi [1] pada menu store, dan menu [1] sampai [5] untuk pembelian potion dari health potion hingga speed potion. Untuk setiap pembelian potion, akan terlebih dahulu diperiksa apakah uang pemain cukup untuk membeli potion yang ia pilih, jika tidak maka akan ditampilkan pesan default yang disimpan dalam predikat **unsufficientGold**, jika cukup, maka pemain akan mendapatkan potion yang bersangkutan dalam inventorynya dan uangnya akan terpotong sejumlah harga potion yang bersangkutan.

Untuk sistem gacha, terdapat 4 tipe gacha, dimulai dari yang standard, silver, gold, hingga gambler, dengan masing-masing memiliki spesifikasi berupa probabilitas yang berbeda-beda untuk mendapatkan item dengan grade tertentu. Setelah pemain memilih tipe gacha yang ia ingin beli, pemain akan diperiksa apakah gold/uang yang mereka miliki cukup untuk membeli tiket gacha yang bersangkutan, jika iya, maka akan terpanggil predikat untuk mengambil hasil gacha sesuai dengan gacha yang mereka pilih. Misalkan ketika player membeli tiket untuk gacha standard, maka akan terpanggil predikat **getGachaResultStandard(R)** dimana R adalah angka random dari 1-100. Spesifikasi gacha standard adalah 67% ghost grade, 18% demon grade, 9% mamon grade 5% lucifer grade, 1% god grade. Dalam predikat **getGachaResultStandard(R)** akan dipanggil predikat lain yang bersesuaian dengan spesifikasi gacha tersebut, misalkan angka R adalah 82, maka akan dipanggil predikat **randomDemonGrade**, untuk memberikan player salah satu dari item dengan grade demon.

1. **battle.pl**

Dalam battle.pl terdapat beberapa predikat, yang terutama adalah **playerTurn** dan **enemyTurn**. **playerTurn** berisikan berbagai opsi yang player dapat lakukan ketika turnnya tiba, entah itu untuk melancarkan serangan, serangan spesial, memakai potion, ataupun lari. Sedangkan predikat **enemyTurn**, berfungsi untuk menampilkan apa yang musuh lakukan pada gilirannya, entah itu melancarkan serangan biasa ataupun srangan spesialnya. Kemudian ada predikat **attackActPlayer** yang me-random apakah serangan yang player lancarkan mengenai musuh atau miss. lalu ada predikat **specialActPlayer** yang menentukan apakah player dapat meluncurkan serangan spesialnya atau tidak. Kemudian sesuai dengan menu yang dapat dipilih player, ada predikat **drinkAct** dan **runAct**, **drinkAct** berfungsi agar player dapat meminum potion yang ada dalam tasnya, sedangkan **runAct** berfungsi untuk menentukan apakah player dapat lari dari musuh yang sedang ia hadapi. Kemudian untuk pelengkap **specialActPlayer,** terdapat predikat updateCD yang mengupdate cooldown special attack menjadi cooldown-1 setiap turnnya, lalu terdapat juga predikat resetCD yang mengembalikan nilai awal cooldown. Kemudian ada 4 predikat yang menghandle permasalahan transfer damage, yaitu **applyDmgPlayer(Damage)** untuk mengaplikasikan serangan pemain kepada musuh, **applyDmgEnemy(Damage)** untuk mengaplikasikan serangan musuh kepada pemain, **applySpPlayer(Damage)** untuk mengaplikasikan serangan spesial pemain kepada musuh, **applySpEnemy(Damage**) untuk mengaplikasikan serangan spesial musuh kepada pemain.

Untuk sistem battle, pertama ditentukan dulu siapa yang akan menyerang duluan berdasarkan speed enemy dan speed player. Jika speed player lebih besar dari speed enemy, maka player menyerang duluan, jika tidak, maka enemy akan menyerang duluan, serang-serangan yang berdasarkan giliran ini akan terus berlanjut hingga ada salah satu yang mati. Ketika player mati, maka program akan diberhentikan, jika enemy mati maka player akan mendapat EXP dan gold sesuai dengan exp dan gold yang telah ditentukan sebelumnya di enemy.pl. Kondisi player mati dapat diperoleh dari predikat **isPlayerDead**, kondisi enemy mati dapat diperoleh dari predikat **isEnemyDead**, kondisi battle selesai atau tidak diperoleh dari predikat **isBattle().** Penambahan gold dilakukan dengan predikat **addGold(X,Add)** dengan **X** adalah username dari player, dan **Add** adalah jumlah gold yang akan ditambahkan. Terakhir ada predikat **winningBattle** yang berisi prosedur-prosedur untuk mengakhiri suatu battle dengan kondisi player menang.

1. **enemy.pl**

File enemy.pl mengurus semua hal yang berkaitan dengan permusuhan yang terjadi di dunia game ini. File ini dimulai dengan deklarasi fakta **isEnemy(<nama musuh>)** yang menyatakan nama-nama musuh yang ada. Kemudian untuk pembentukan setiap enemy terdapat predikatnya masing-masing, seperti **createSlime(<tipe>)**, dengan tipe akan mempengaruhi spesifikasi musuh (dalam kasus ini slime) yang dibentuk. Ada 6 jenis musuh yang tersedia yaitu slime, goblin, wolf, spider, zool, apex. Lalu ada predikat **generateSlime** untuk meng-generate tampilan awal ketika player bertemu dengan musuh yang bersesuaian (dalam kasus ini slime). Tampilan awal berupa gambar yang merepresentasikan musuh yang bersangkutan dan properti yang ia miliki (health, attack, defense, speed).

Untuk meng-generate enemy, digunakan predikat **generateEnemy(C)**. Sistem generate enemy ini akan bergantung pada zona tempat player tersebut berada. Ketika player ada di zona 1, maka hanya slime dan goblin yang mungkin ditemukan oleh player tersebut. Kemudian ada predikat **isEncounter** yang menentukan apakah player menemukan musuh pada langkah yang ia baru saja ambil. Kemudian ada predikat dungeon yang menginisiasi predikat isBattle dengan musuh Apex. Terakhir ada predikat **isPlayerDead** dan **isEnemyDead** yang dipakai di **battle.pl**

1. **quest.pl**

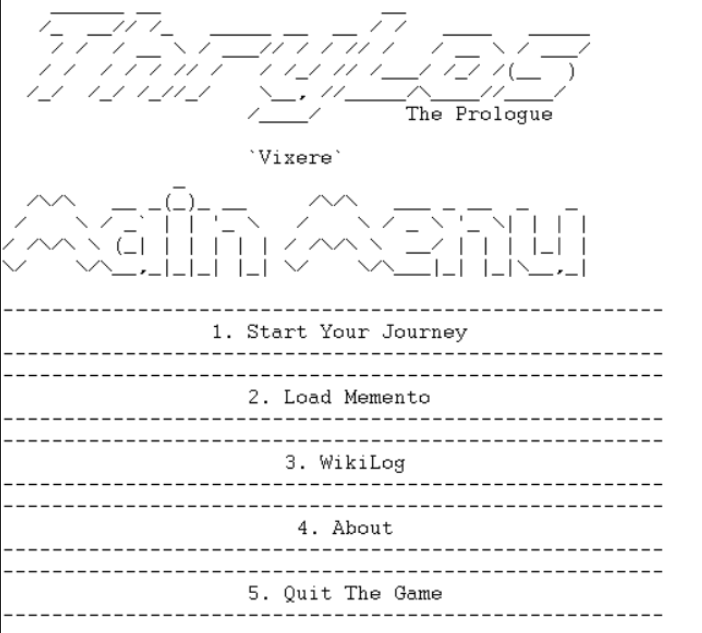
Pada quest.pl, terdapat semua fakta dan rules yang berkaitan dengan quest, quest hanya dapat diambil ketika player mencapai titik Q pada map. Hanya ada 1 quest yang bisa aktif pada 1 waktu. Quest mengharuskan player untuk membunuh 3 jenis monster tertentu, bergantung pada tingkat kesulitan quest, dan level player sekarang, reward dari quest juga menyesuaikan hal-hal tersebut (reward berupa Exp dan Gold yang jumlahnya menyesuaikan).

Progress dari quest dapat dicek dengan menggunakan predikat **quest**. Progress dari quest akan otomatis diupdate setelah battle selesai. Setelah battle selesai juga akan dicek apakah quest sudah beres atau belum, jika sudah reward dari quest akan otomatis didapatkan player walaupun player tidak ada di titik Q di map.

**File Utama Permainan dan *Command* yang Bisa Dilakukan Pemain**

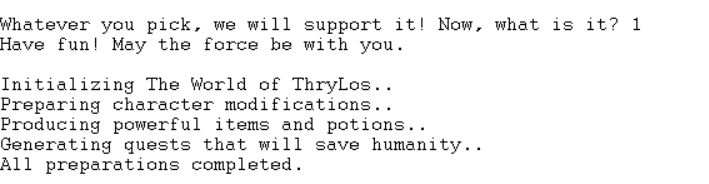
File utama yang digunakan untuk menggabungkan semua *commands* dari file-file yang telah dijelaskan sebelumnya adalah file **main.pl**. Berikut merupakan *commands* yang bisa dilakukan pemain di dalam game tersebut.

1. **thrylos.**

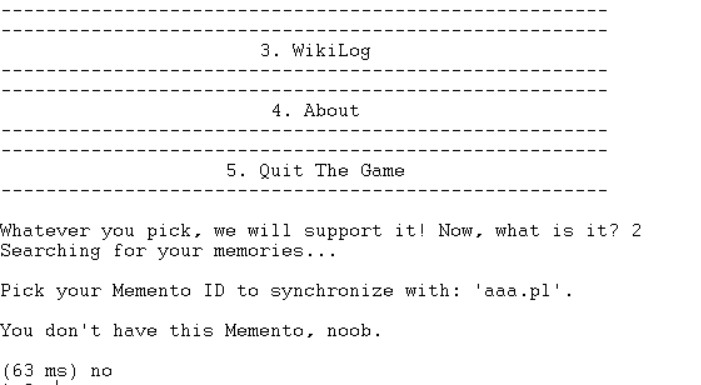
Ketika file **main.pl** telah di-*consult* melalui GNU Prolog Console, maka *command* `**thrylos.**`ini digunakan untuk memulai permainan. Setelah permainan dimulai, pemain dapat memanggil Main Menu (tanpa ada nama gamenya) dengan menggunakan *command* tersebut atau dengan memasukkan input **`mainMenu.`** ke dalam *console*.

Pada Main Menu ini, pemain dapat memilih beberapa opsi seperti di atas tetapi apabila pemain memilih selain 1-4, pemain akan otomatis *quit game* dan ditanya ingin menyimpan *progress game* atau tidak.

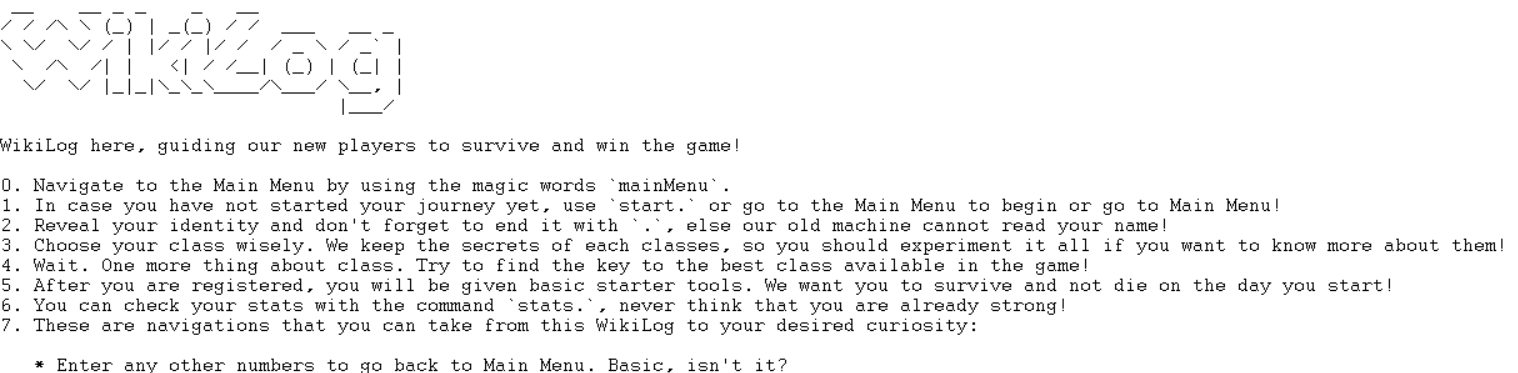
Apabila pemain memilih 1, game akan dimulai dan menampilkan beberapa kata *preparation* seperti ini. Setelah itu, akan muncul tampilan bagi pemain untuk memasukkan *username* dan *class* pilihannya yang akan dijelaskan lebih lanjut nanti..



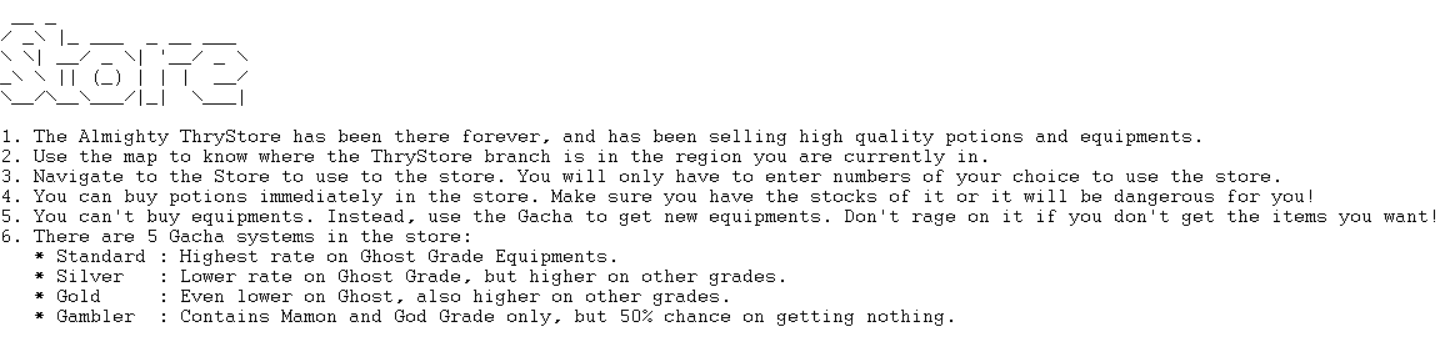
Apabila pemain memilih 2 yang adalah *load game* hasil *save* sebelumnya, pemain akan diminta untuk memasukkan nama file yang ingin dimuat dan hasilnya akan seperti ini apabila pemain tidak memiliki file save sebelumnya. Keterangan lebih lanjut akan dijelaskan pada bagian **Bonus**.



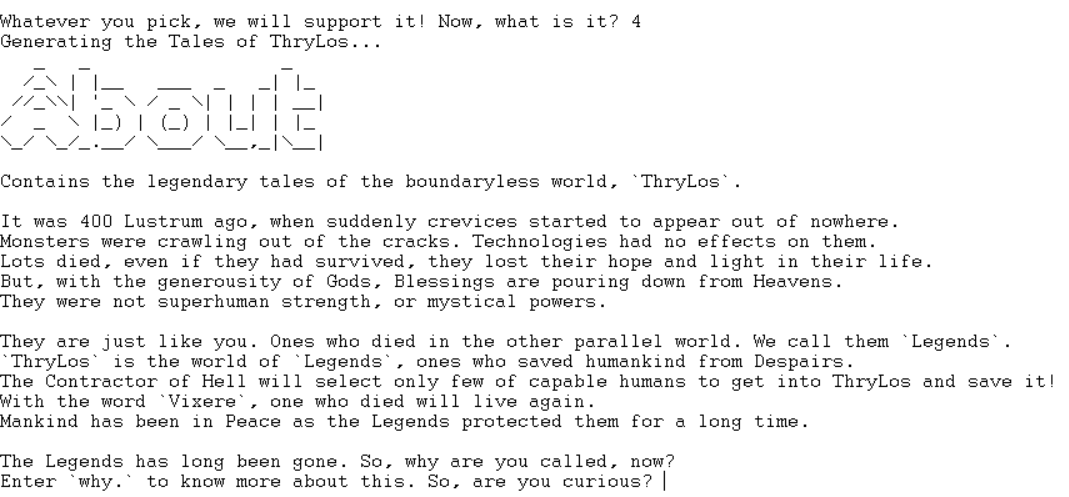
Apabila pemain memasukkan opsi 3, yaitu WikiLog, maka akan menampilkan semacam *guideline* garis besar dari game ini, terdapat beberapa pilihan Wiki yang bisa diakses.



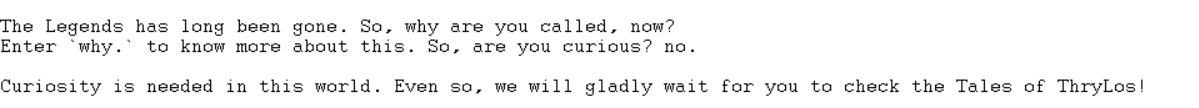
Pemain dapat memilih opsi lagi sesuai informasi-informasi yang ingin pemain dapatkan tentang game ini, beberapa contoh wiki tersebut adalah seperti dibawah saat pemain ingin mencari informasi mengenai *store*.



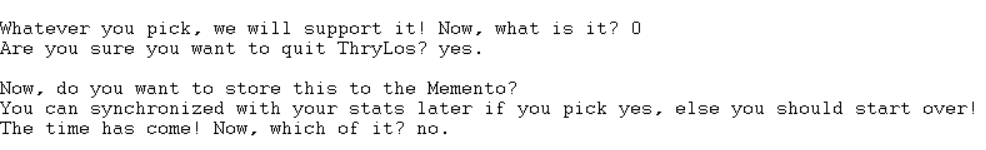
Lalu, apabila pemain memilih opsi 4 yaitu “About”, pemain dapat membaca cerita singkat tentang latar belakang judul dan permainan ini.



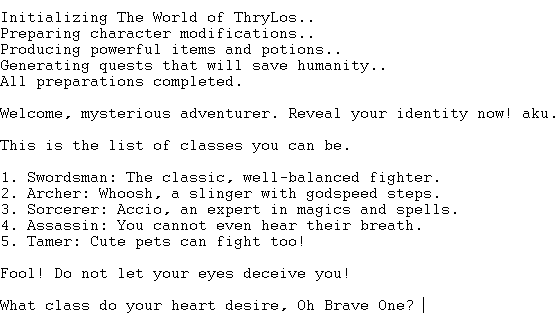
Ada fitur unik pada tampilan *About* ini dimana kita dapat memilih untuk mengetahui cerita lebih lanjut tentang game ini. Dengan memasukkan ‘why’ pada pertanyaan terakhir maka akan ditampilkan sisa cerita sedangkan pilihan lain akan menampilkan pesan berikut.



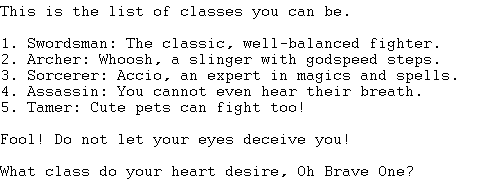
Dan terakhir apabila pemain memilih angka yang tidak ada di pilihan, maka secara otomatis pemain akan keluar dari *game*.



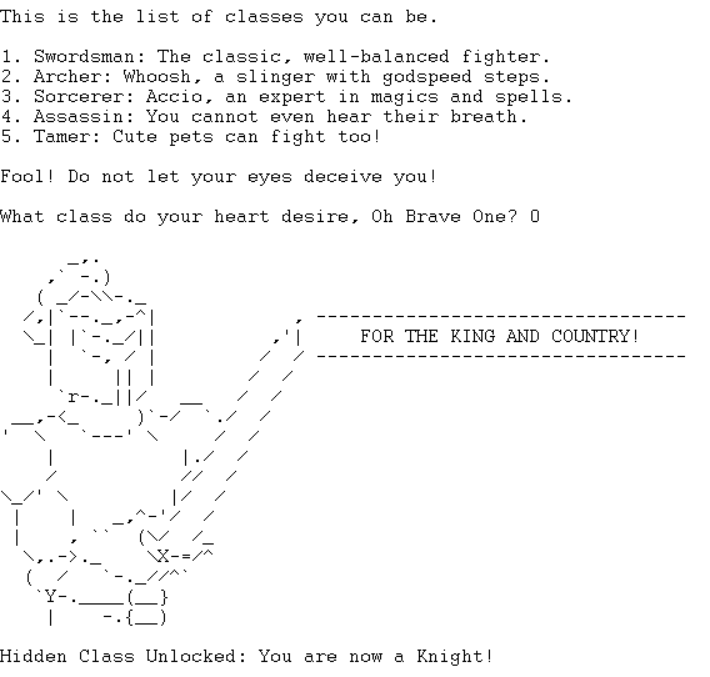
Pada saat *quit game*, pemain akan ditanya apakah ingin menyimpan hasil permainan yang sudah dimainkan pada file eksternal atau tidak. Meskipun *game* belum dimulai, sistem akan tetap menanyakan pemain untuk melakukan *save* atau tidak. Jika ternyata dipilih iya, maka tentunya file hasil penyimpanan merupakan file kosong yang tidak berisi informasi apa-apa terkait permainan.

Selanjutnya, saat pemain ingin memulai memainkan game dan memilih pilihan pertama, yaitu “Start Your Journey”, pemain akan dibawa ke tampilan memasukkan *username* seperti yang telah dijelaskan tadi.

Setelah memasukkan username, pemain akan diminta memilih *class*. Dalam game ini, terdapat 5 *class* biasa dengan memilih opsi 1-5 dan *hidden class* di mana pemain dapat memasukkan input ‘0’ dan secara random akan mendapat salah satu dari 5 *hidden job*.

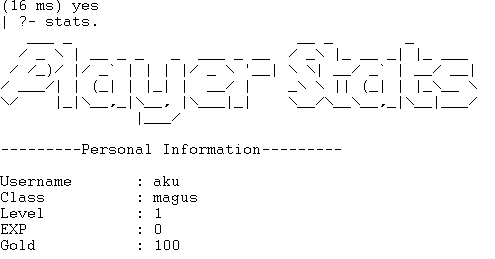


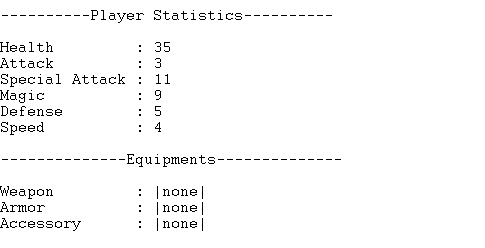
Ini adalah tampilan apabila pemain menemukan *hidden class* dengan masukkan ‘0’.



1. **stats.**

Setelah membuat karakternya, pemain dapat langsung mengecek statistik awal *character* dengan command **‘stats.’** yang juga menampilkan *equipment* yang sedang dipakai.

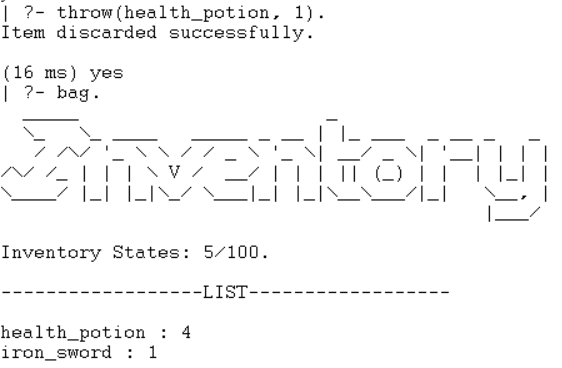
****

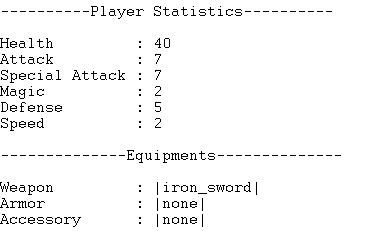
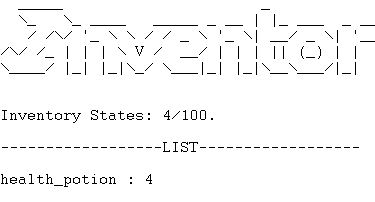
****

1. **bag.**

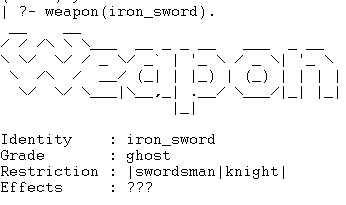
Pemain juga dapat mengecek isi *inventory* dengan mengetik command **‘bag.’** di *console* dan akan ditampilkan isi *inventory* berikut juga kapasitas *inventory* dan jumlah barang.



Pemain juga dapat membuang barang dengan command **‘throw(nama\_barang, jumlah)’**. Jika jumlah item yang dibuang melebihi jumlah barang tersebut dalam *bag*, maka semua item tersebut di dalam *bag* akan dibuang.

Pemain juga dapat menggunakan perintah **‘equipWp(nama\_wp).’** untuk menggunakan senjata, **‘unequipWp(nama\_wp)’** untuk melepas senjata yang dipakai, *armor* dan *accessories* juga dapat menggunakan command yang mirip dengan mengganti **‘Wp’** dengan **‘Arm’** atau **‘Accs’**. Di bawah ini adalah contoh memakai suatu *weapon*, *stats* akan bertambah, terlihat *equipment* apa saja yang dipakai dan *item* di *inventory* berkurang.

*Player* juga dapat mengecek informasi tentang *items* yang ada di *game* sesuai keinginan dengan menggunakan *command* **‘weapon(nama\_wp).’, ‘armor(nama\_arm).’, accessory(nama\_accs).’,** dan **‘potion(nama\_potion).’**.



Potion juga dapat digunakan di luar *battle* dengan mengetik *command* **‘drink(nama\_potion).’** dan *stats* akan langsung diperbarui jika pemain memiliki *potion* yang diminum. Jika tidak, akan ditampilkan pesan untuk mengisi ulang *stock potion*.



1. **map.**

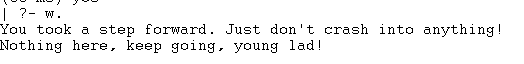
Pemain dapat menggunakan command **‘map.’** untuk melihat dan mengakses peta dari dunia ThryLos ini.

Pada map juga sudah tersedia *legend* untuk memudahkan pemain mengetahui lokasi-lokasi yang perlu diketahui seperti **P** untuk lokasi **player** saat ini, **S** adalah **store** untuk membeli *potion* dan Gacha, sedangkan **Q** adalah **quest** untuk melakukan pengambilan *quest*/misi.

1. ***Move Commands***

Untuk setiap lokasi khusus di peta akan diberikan predikat untuk menandakan apabila tempat tersebut sudah diambil atau belum, serta predikat penanda fungsionalitas dari tempat tersebut. Pemain dapat melakukan gerakan yang valid apabila tidak koordinat baru yang merupakan hasil pergerakannya bukan merupakan dinding. Untuk mobilisasi dalam peta, player dapat memasukkan komando sebagai berikut :

**w.** untuk maju ke depan/atas



**a.** untuk berjalan ke kiri



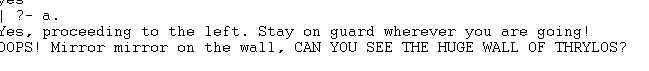
**s.** untuk berjalan ke bawah



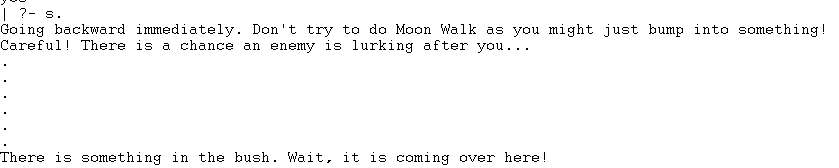
**d.** untuk berjalan ke kanan



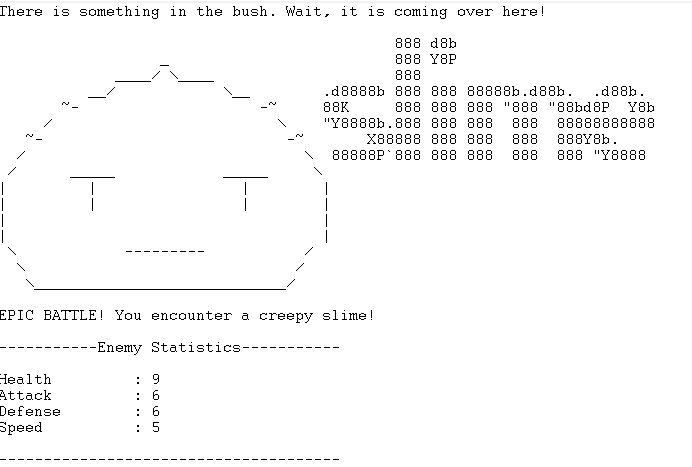
Apabila pemain bergerak kepada *border* dari peta atau area di dalam peta yang tidak dapat ditembus seperti dinding dalam, maka pesan berikut akan ditampilkan kepada pemain.



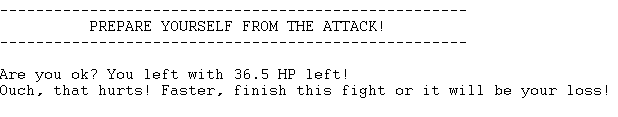
Untuk setiap pergerakan di zona tertentu, *player* memiliki kemungkinan tertentu untuk bertemu dengan *monster*/musuh. Ketika *player* bertemu dengan musuh, pesan yang diberikan akan berbeda seperti tampak pada gambar dibawah.



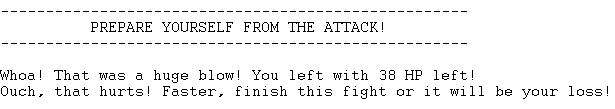
Setelah itu pemain akan diberikan tampilan yang menyatakan musuh apa yang ia hadapi beserta dengan statistik musuhnya.



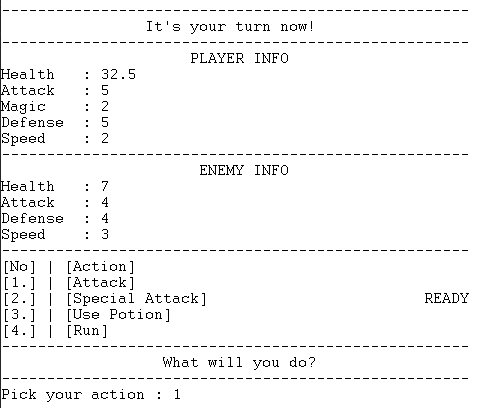
Setelah itu sistem battle-pun akan dimulai. Sejak *turn* pertama *combat* dengan musuh sudah terjadi. Apabila pemain memiliki karakter dengan *speed* yang lebih cepat daripada musuh, maka pemain akan dipersilahkan sistem untuk menyerang duluan, tetapi jika tidak (seperti dalam kasus ini), musuh yang akan diberikan kesempatan untuk menyerang duluan. Ada dua jenis serangan yang musuh dapat lakukan, yaitu serangan biasa dengan kemungkinan eksekusi 80%, dan serangan spesial dengan kemungkinan eksekusi 20%. Apabila musuh melaksanakan gilirannya, maka akan ditampilkan pesan yang bersesuaian. Jika musuh menyerang dengan serangan biasa, maka akan ditampilkan pesan sebagai berikut.



Namun, jika musuh menyerang dengan serangan spesial, maka akan ditampilkan pesan sebagai berikut.



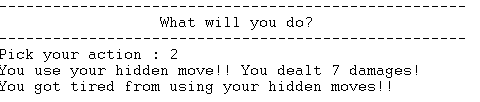
Lalu pada saat giliran *player* tiba, maka pemain akan diberikan *prompt* berikut yang menyatakan opsi apa saja yang pemain dapat lakukan.



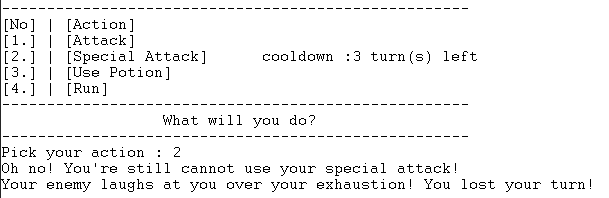
Apabila *player* memilih untuk menyerang, maka akan ada pesan yang menyatakan seberapa besar *damage* yang ia telah berikan kepada musuh yang bersangkutan.



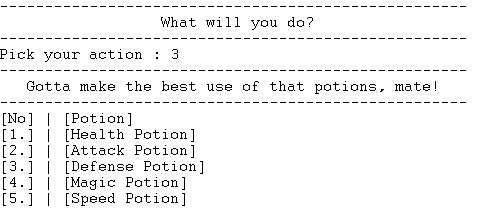
Apabila player memilih untuk menggunakan *special attack*nya, maka akan ditampilkan pesan sebagai berikut jika berhasil



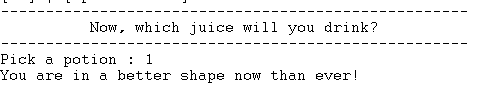
Setelah pemakaian *special attack*, player akan melihat hitungan *cooldown* untuk menandakan bahwa *special attack*nya belum siap untuk dipakai. Apabila *player* memaksakan pemakaian *special attack*nya, maka *turn*nya akan hangus begitu saja. Untuk setiap giliran yang lewat, *cooldown special attack* akan berkurang dan apabila cooldownnya sudah selesai, tampilan *special attack* akan berubah menjadi READY.



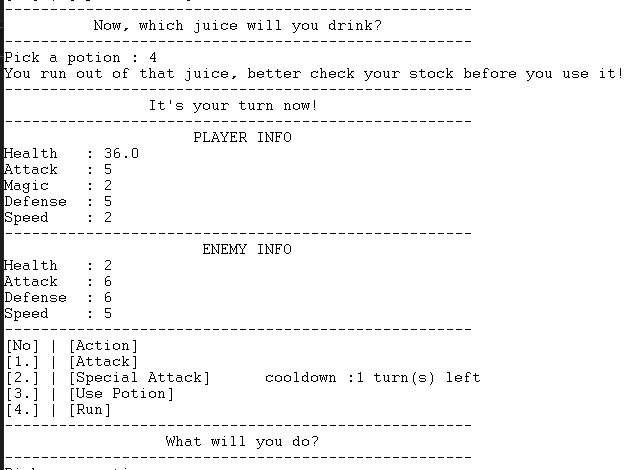
Apabila player memilih untuk menggunakan potion yang ada di tasnya, maka pesan sebagai berikut akan ditampilkan pada layarnya.



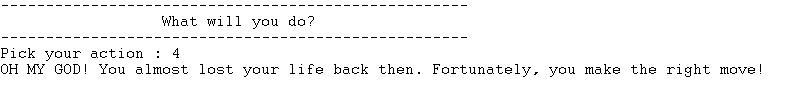
Setelah menggunakan salah satu dari *potion* yang ada di atas, apabila sukses, player akan diberikan pesan sebagai berikut, efek dari *potion* tersebut akan langsung mengubah statistik playernya, dan giliran akan berganti ke giliran lawan



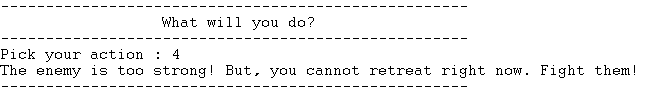
Apabila player memilih untuk memakai *potion* yang ia tidak miliki di bag nya maka pesan sebagai berikut akan ditampilkan pada player dan giliran akan tetap giliran player



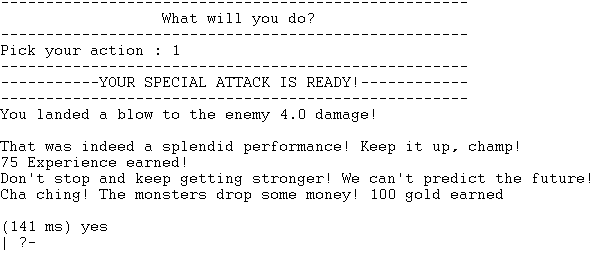
Terakhir, player memiliki kemungkinan untuk lari dari *battle* yang sedang berlangsung dengan menekan opsi 4, namun *player* tidak akan selalu berhasil untuk lari. Ketika *player* berhasil lari akan ditampilkan pesan sebagai berikut.



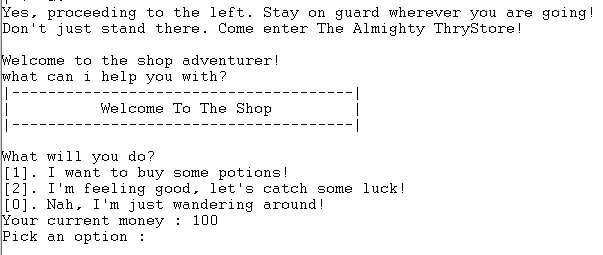
Namun, jika *run* yang dilakukan gagal, akan ditampilkan pesan berikut dan *battle* akan terus berlanjut tetapi giliran akan berganti ke giliran lawan.



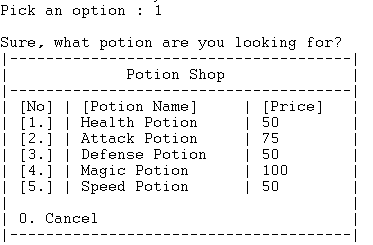
Pada akhir *battle*, terdapat dua kemungkinan: *player* akan memenangkan *battle* atau kalah. Apabila *player* kalah dan mati, permainan akan memasuki *fail state* yang sering disebut *game over*. Tetapi apabila *player* memenangkan *battle* akan keluar sedikit dialog tentang jumlah *exp* dan gold yang didapat lalu akan keluar *mode battle* seperti dibawah ini.



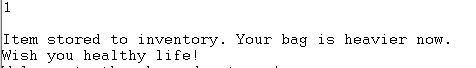
Untuk masuk dan menggunakan lokasi-lokasi dengan fungsionalitas tertentu, seperti *store* dan *quest*, maka pemain dapat langsung mengunjungi *store* dengan berjalan menuju legenda S yang ada pada *map*. Pemain akan disuguhi dengan dua buah menu utama yaitu [1] untuk membeli *potion*, dan [2] untuk membeli tiket Gacha untuk memperoleh item secara *random*, atau [0] untuk keluar dari *store*.



Ketika pemain memilih [1], maka akan ditampilkan *potion* apa saja yang tersedia di store tersebut beserta harganya.



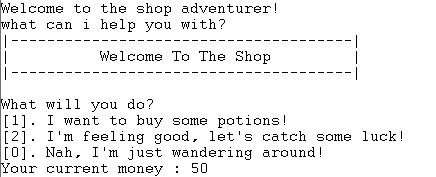
Terdapat 5 *potion* di *store* beserta dengan harganya masing-masing, pemain dapat membeli *potion* satu per satu. Pada saat pemain membeli *potion* tertentu, akan di cek terlebih dahulu apakah pemain memiliki uang yang cukup untuk membeli *potion* yang bersangkutan atau tidak. Jika pemain memiliki uang yang cukup, maka akan ditampilkan pesan seperti dibawah ini.



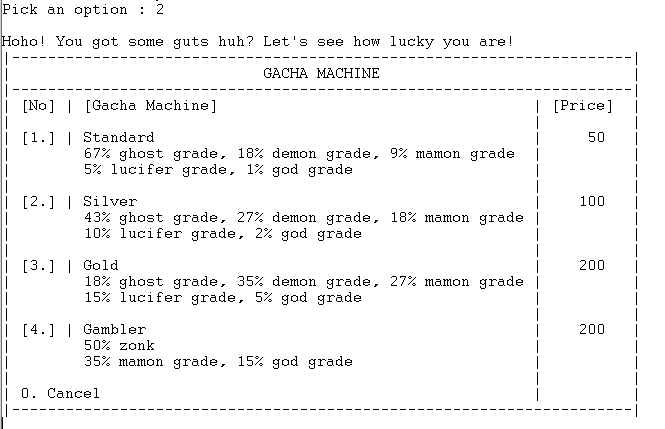
Tetapi jika pemain tidak memiliki uang yang cukup, maka pemain akan ditampilkan pesan seperti berikut.



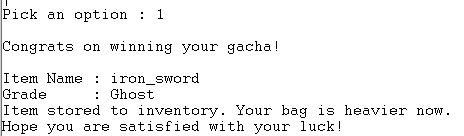
Baik pemain sukses membeli suatu barang ataupun gagal, pemain akan di *redirect* ke percakapan awalnya dengan pemilik toko.



Sedangkan untuk mengakses menu gacha, pemain dapat memilih menu [2] pada *store*, dan pemain akan mendapatkan tampilan sebagai berikut. Pada tampilan ini terdapat 4 jenis Gacha yang pemain dapat beli beserta dengan spesifikasinya masing-masing. Berikut urutan *grade items* dari yang paling biasa ke yang paling luar biasa : *ghost grade, demon grade, mamon grade, lucifer grade, god grade*.

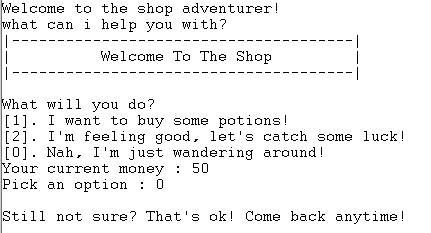


Ketika pemain memilih salah satu dari menu yang tertera di atas dan uang pemain mencukupinya, maka sistem Gacha akan jalan, tetapi jika uangnya tidak mencukupi maka pesan yang sama akan tertampil seperti ketika pemain tidak cukup uang untuk membeli potion. Berikut adalah tampilan yang akan muncul ketika pemain berhasil membeli salah satu tiket Gacha.



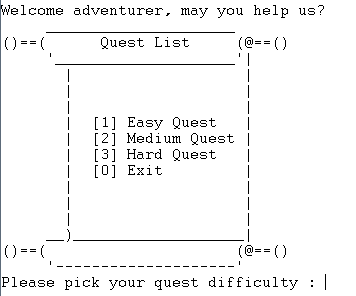
Pemain akan diberikan pesan berupa nama item yang didapatkan, gradenya, dan notifikasi yang menyatakan bahwa item yang bersangkutan berhasil disimpan dalam *bag*.

Kemudian untuk keluar dari store, pemain dapat memasukan opsi [0] pada menu *store* yang utama.



Selain itu, pemain juga dapat mengambil *quest*/misi dari tempat pengambilan *quest* yang dilambangkan dengan Q pada map. Pemain hanya dapat memiliki sebuah misi yang aktif dan apabila ingin mengambil *quest* yang lain, pemain harus menyelesaikan *quest* sebelumnya terlebih dahulu.

Ketika mencapai Q pada map, pemain akan diminta untuk memilih tingkat kesulitan *quest* dengan angka 1,2,3 dan 0 untuk keluar dan tidak jadi mengambil *quest*.



Setelah memilih *quest*, pemain akan diberitahu *monster* yang harus dibunuh dan jumlah yang harus dibunuh untuk setiap *monster* tersebut.



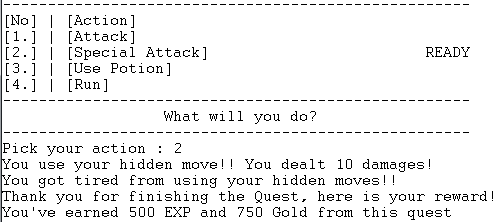
Karena hanya ada 1 quest yang boleh aktif dalam 1 waktu, pemain tidak bisa mengambil quest lagi sampai quest sebelumnya selesai



Untuk mengecek progress dari quest, bisa digunakan predikat **quest**

****

Ketika player sudah membunuh monster sesuai dengan jumlah yang dibutuhkan, quest akan otomatis selesai dan *reward* akan otomatis masuk.



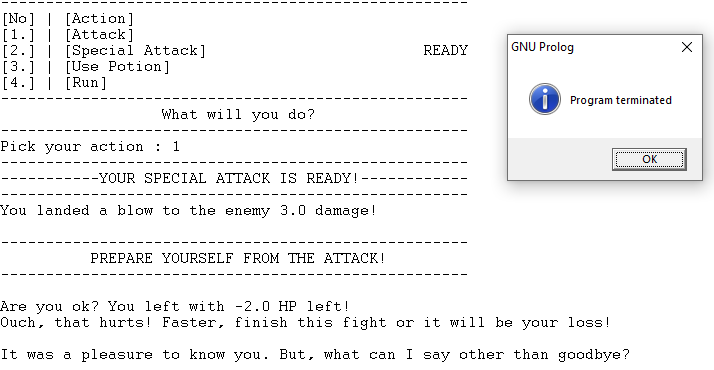
****

Ketika *experience points* yang dimiliki oleh pemain telah melewati batas jumlah tertentu, *player* secara otomatis akan naik ke *level* berikutnya. *Stats* dari pemain akan bertambah sesuai dengan *growth rate* dari masing-masing *level*. Dalam ThryLos ini sendiri terdiri dari 6 buah *level*. *Level* terakhir adalah *level* maksimum yang bisa dicapai oleh pemain dan setelah itu, *stats* dari pemain tidak akan bertambah lagi. Berikut merupakan pesan yang diberikan kepada pemain ketika mencapai *level* maksimum.

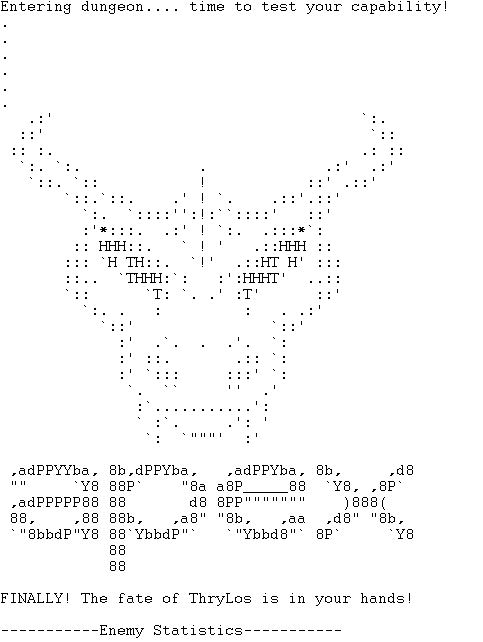
Berikut adalah tampilan *stats* yang dimiliki oleh pemain ketika telah mencapai *level* maksimum.

***Fail State* (*Game Over*) dan *Goal State* (*Winning Condition*)**

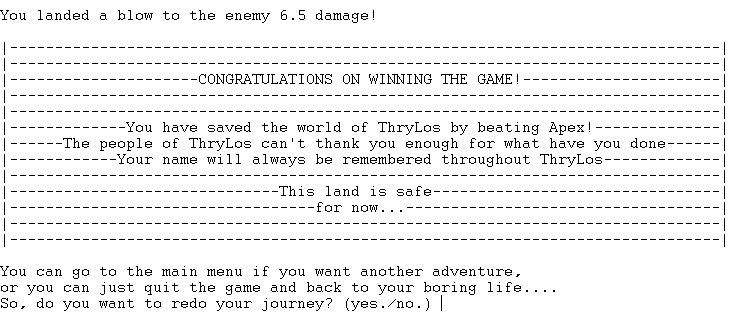
Terdapat dua kondisi yang menandakan berakhirnya game ini, yang pertama adalah saat pemain meninggal/kalah yaitu *fail state* atau disebut juga *game over*. Saat HP pemain mencapai 0 atau dibawah 0, game akan otomatis berhenti dan keluar dari console DAN pemain tidak dapat hidup kembali kecuali meneruskan dari data *save file* terakhir. Kasus *game over* adalah sebagai berikut:



Sementara itu, *winning condition*/*goal state* dari game ThryLos: The Prologue ini adalah mengalahkan sebuah *final boss* yang bernama Apex. Cara melawan *final boss* ini adalah dengan cara memasuki *dungeon* yang ada pada salah satu tempat di dalam dunia ThryLos ini. Saat memasuki dungeon tersebut, player akan langsung melawan *final boss* tersebut seperti ini.



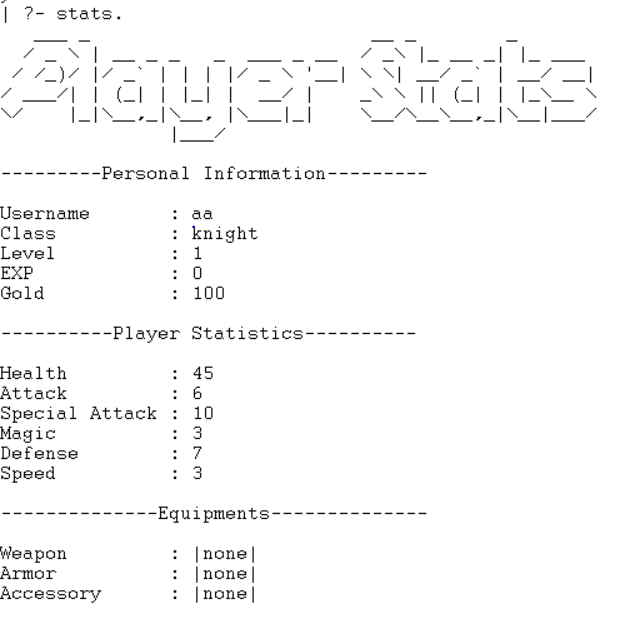
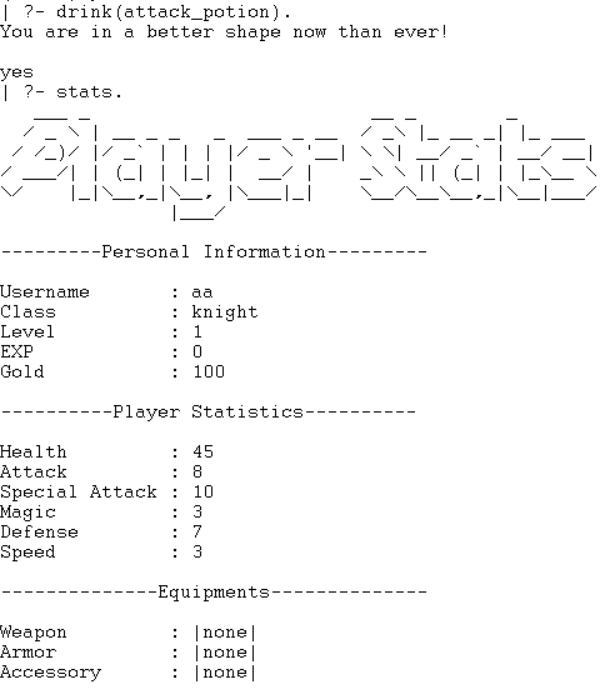
Lalu, apabila *final boss* berhasil dikalahkan maka pemain akan memasuki *goal state* yang akan menampilkan sedikit *credit* untuk pemain.



Akan ada beberapa pilihan untuk *player* apakah ingin bermain kembali dengan mengulang dari awal ke Main Menu atau ingin langsung *quit game* dan *console* akan langsung tertutup.

**Bonus**

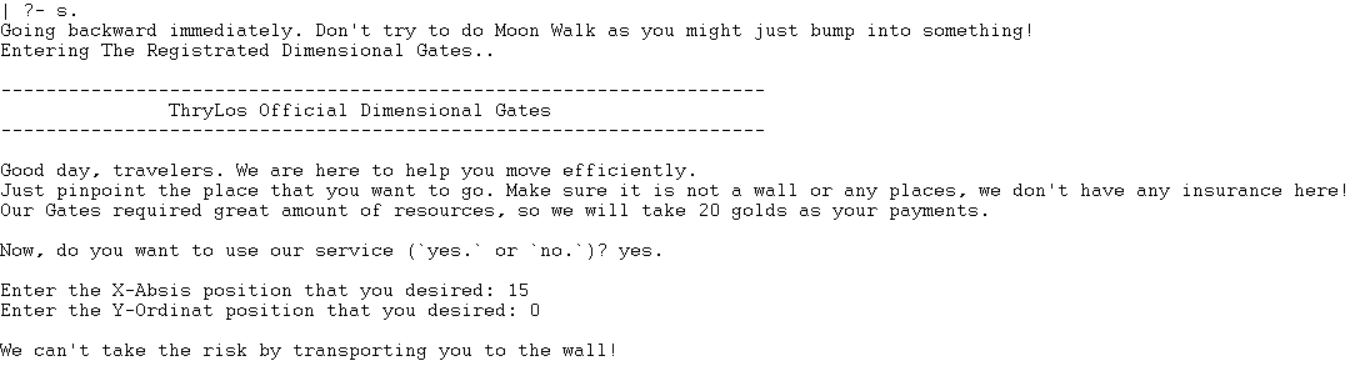
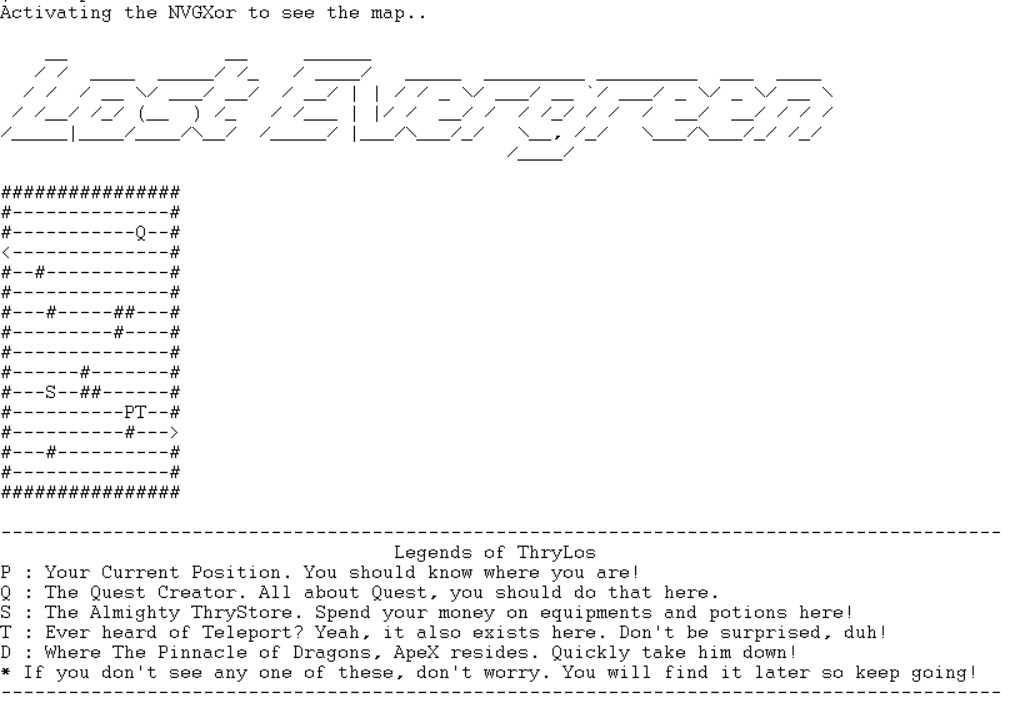
1. **Fungsionalitas *Potion***

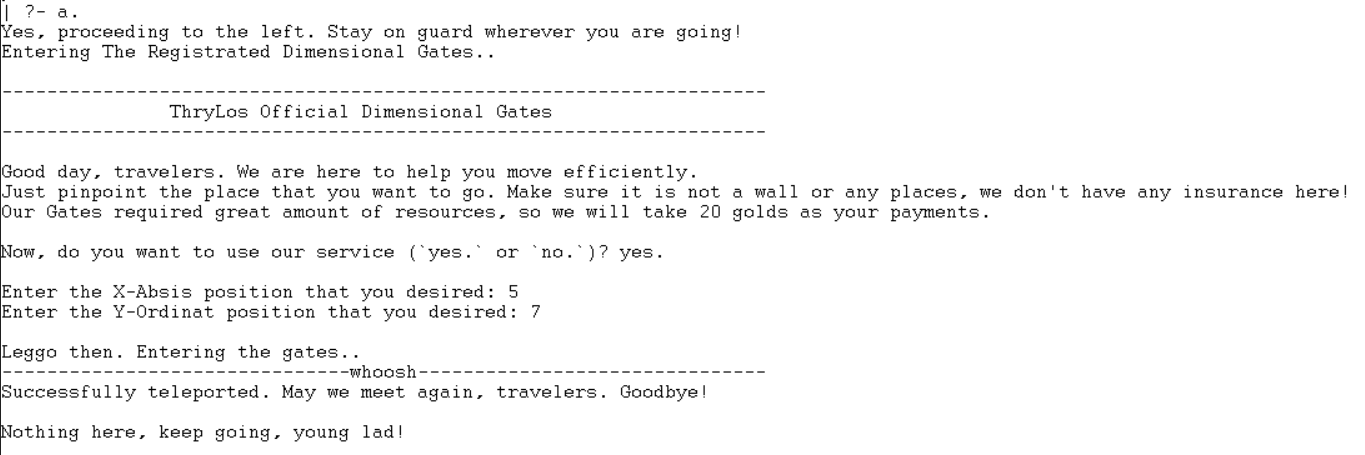
Berhasil diimplementasikan fungsionalitas tambahan untuk *potion*. Selain adanya *health potion*, di dalam game ini juga ada *potion* untuk menambah *stats* lainnya, seperti *attack potion, magic potion, defense potion,* dan *speed potion*. Di *store* sendiri, harga untuk tiap *potion* berbeda-beda. Adapun setiap *potion* memiliki penambahan *stats* sebesar 2 poin, terkecuali untuk *health potion* yang menambah 10 poin sebagai bentuk *balancing game* tersebut. 

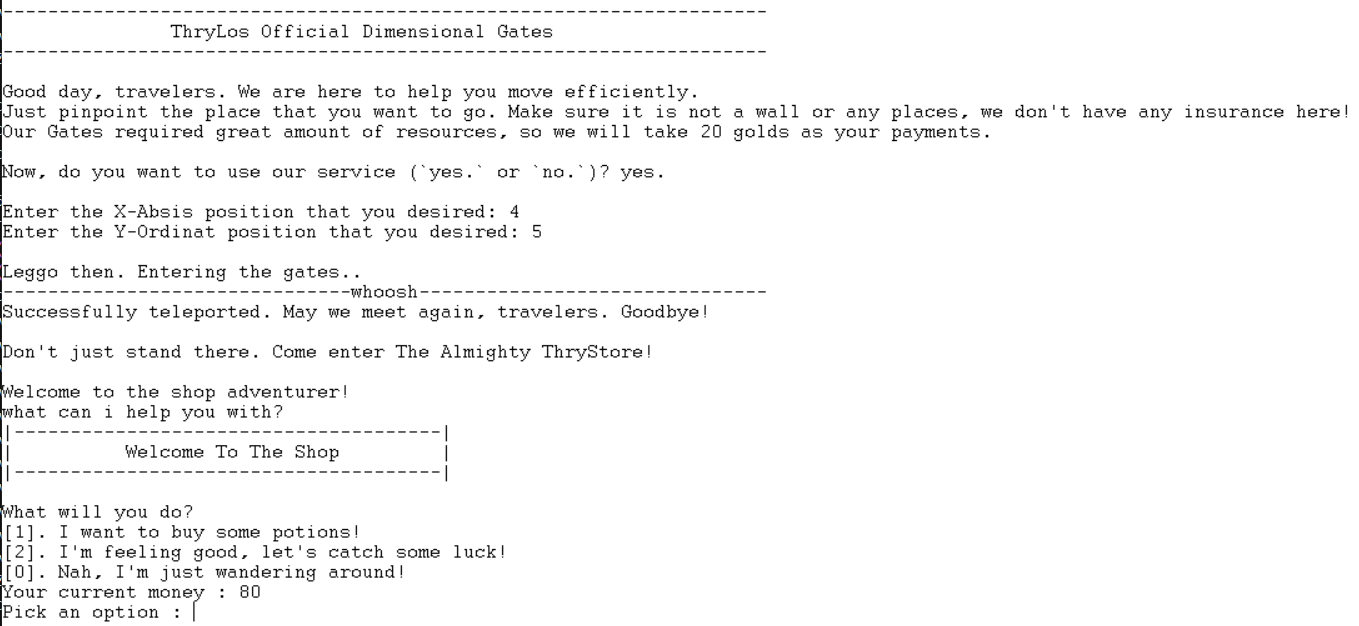
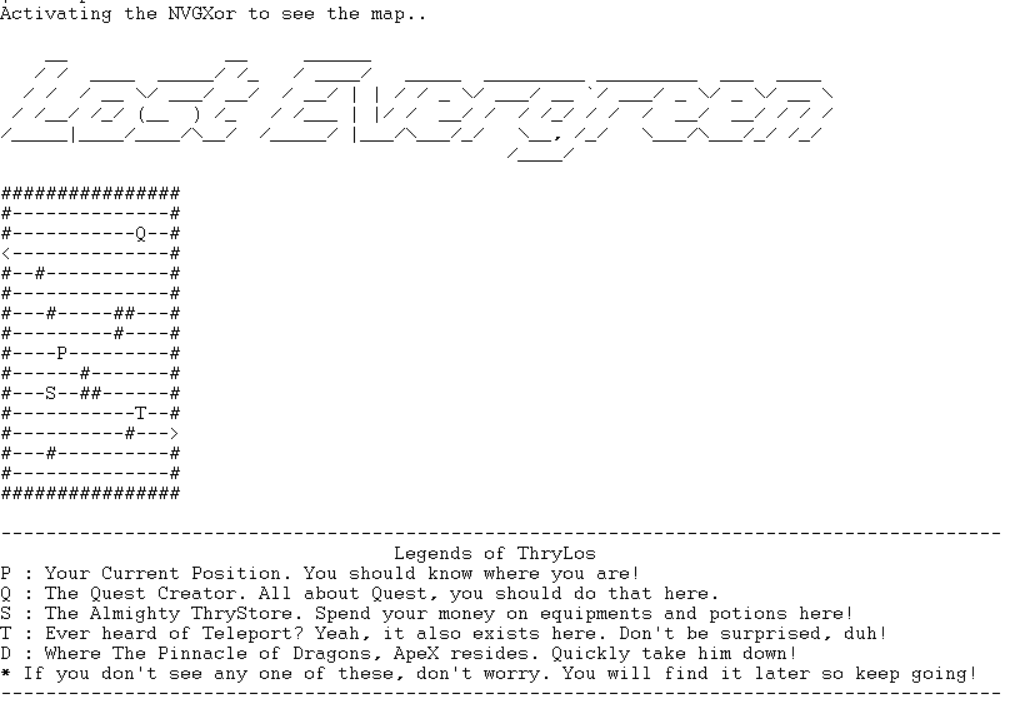
1. **Zona Kemunculan Monster dan Teleportasi**

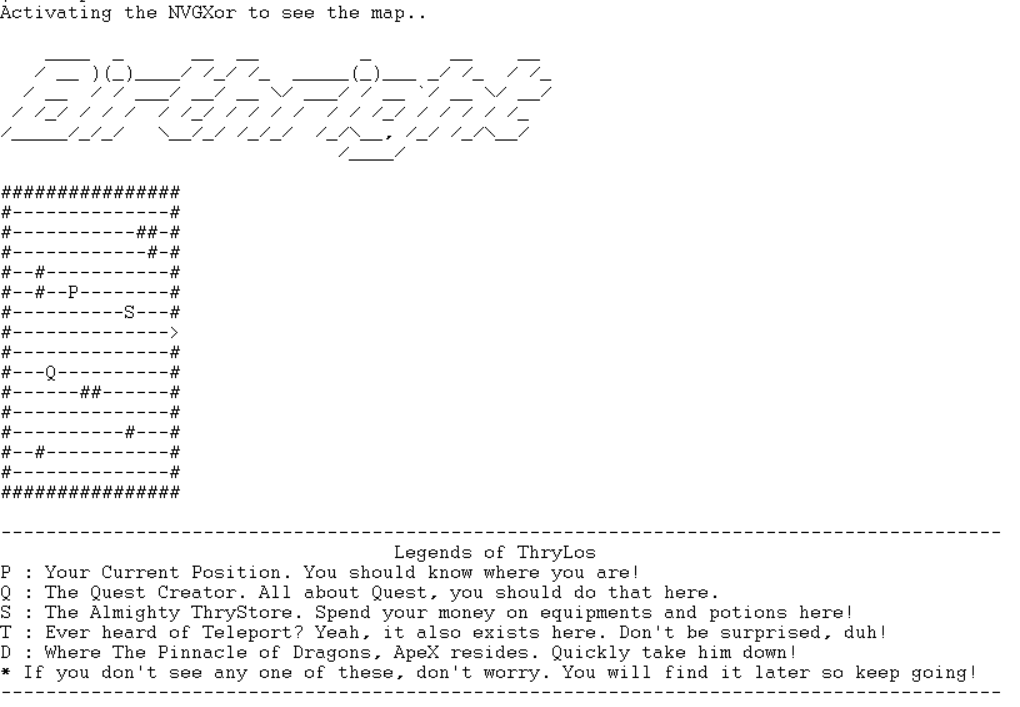
Mekanisme peta dalam game ThryLos ini adalah peta berbasis *region*. Setiap area akan memiliki perbedaan geografi, terutama susunan lokasi-lokasi penting yang terdapat dalam peta. Berikut merupakan deskripsi lebih lanjut dari peta ini. Agar pemain bisa berpindah tempat ke area lainnya, maka pada *border* peta disediakan sebuah *dimensional gate* yang bisa dikunjungi pemain untuk berpindah *region*.

Selain itu, bonus lainnya yaitu sistem teleportasi di dalam game juga berhasil diimplementasikan. Pemain bisa mengunjungi bagian peta yang ditandakan dengan **‘T’** untuk menggunakan fungsi teleportasi. Teleportasi bisa dipakai dalam satu *region* saja dan jika pemain memilih untuk melakukan teleportasi di tempat-tempat khusus, maka fungsionalitas dari setiap tempat tersebut akan bekerja.

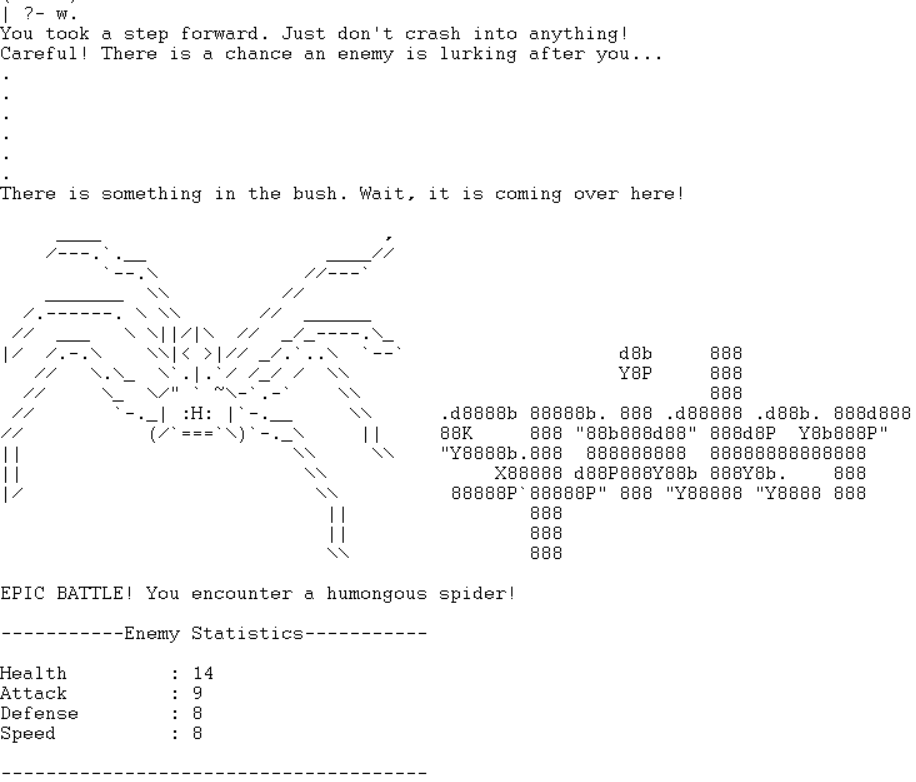
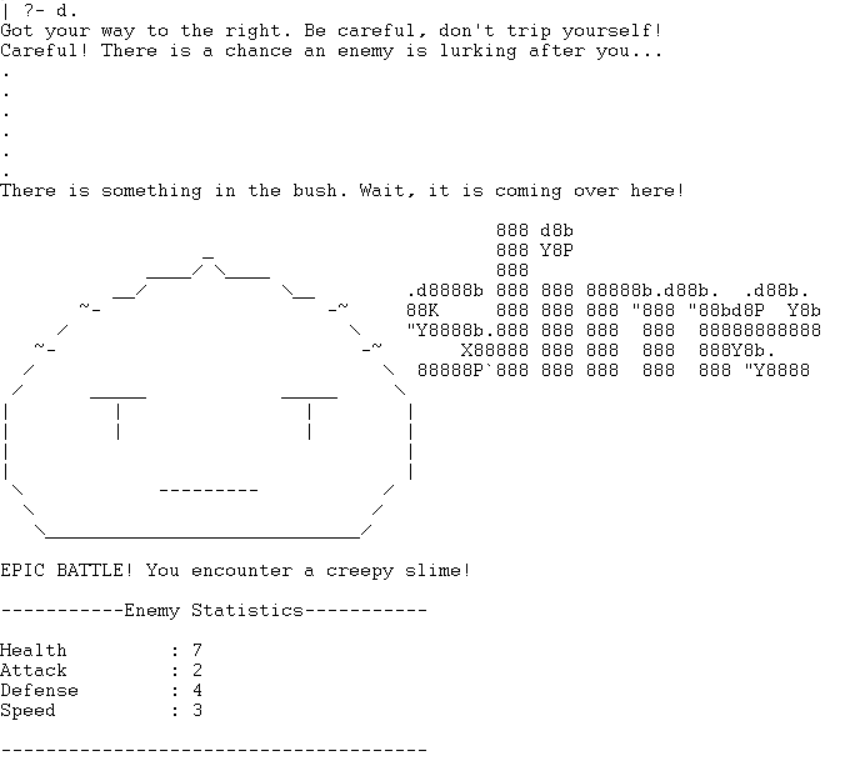
 Misalnya, pemain ingin melakukan teleportasi ke titik yang telah ditempati dinding, maka teleportasi akan gagal. Namun, apabila pemain bisa melakukan teleportasi ke dalam lokasi dengan fungsionalitas tertentu seperti *store* dan juga *quest* yang akan otomatis melakukan fungsionalitasnya masing-masing. Perlu diingat bahwa penggunaan teleportasi membutuhkan sumber daya yang tidak sedikit, sehingga pemain yang menggunakan sistem teleportasi ini harus membayar 20 Gold sebagai biaya teleportasi tersebut.





Area pertama merupakan area tempat pertama kali *player spawn* sehingga diberi nama BirthRight. Pada area ini, hanya akan disediakan *store*, *quest*, dan juga dinding dalam pada peta sehingga area ini menjadi area paling sederhana. Pemain dapat berpindah ke *region* selanjutnya melalui *dimensional gate* yang bisa diakses dengan mengunjungi *border* peta. Tidak terdapat fungsi teleportasi di dalam map ini karena dianggap sebagai *region* pemula yang tidak terlalu diserang oleh *monster*. 

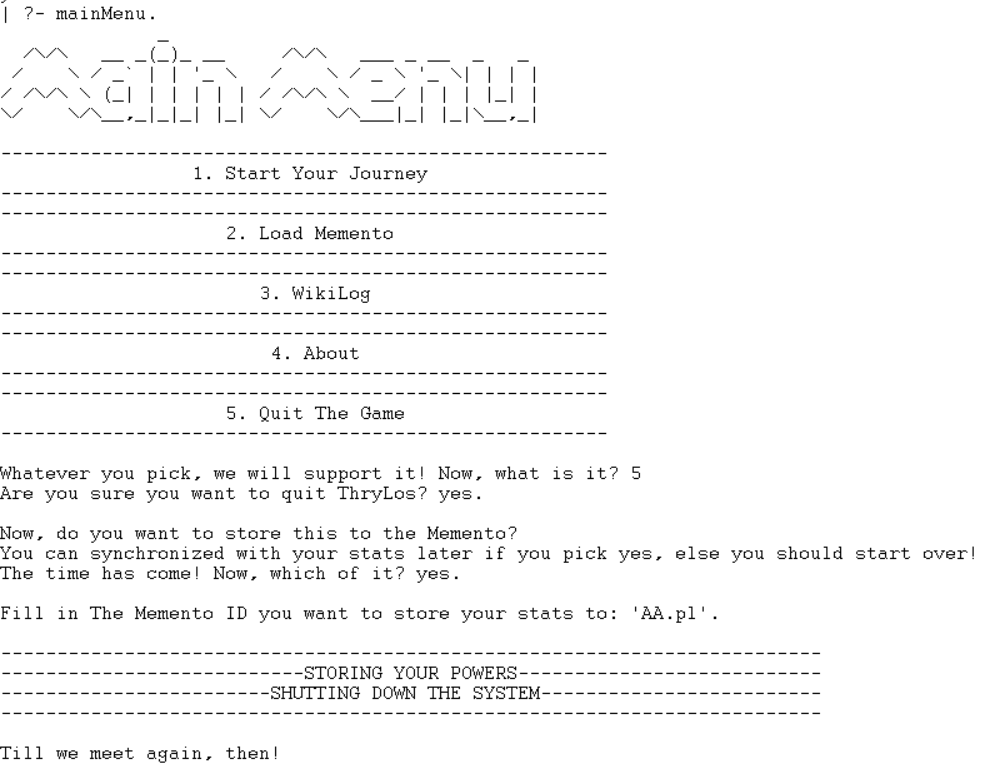
Ketika memasuki area kedua, maka pemain akan berada di depan *dimensional gate*. Setelah itu, pemain akan memiliki peta dengan geografi dan susunan yang berbeda dari peta sebelumnya. Di area kedua ini, ada fungsionalitas teleportasi yang bisa digunakan pemain serta dua buah *dimensional gate* untuk maju ke zona berikutnya dan mundur ke zona sebelumnya. Pada zona ini, *monster* yang muncul adalah *spider* dan juga *wolf,* dengan *wolf* memiliki peluang *spawn* yang lebih besar. Adapun pada peta sebelumnya monster yang muncul adalah *slime* dan *goblin*, dengan *slime* memiliki peluang *spawn* yang lebih besar.

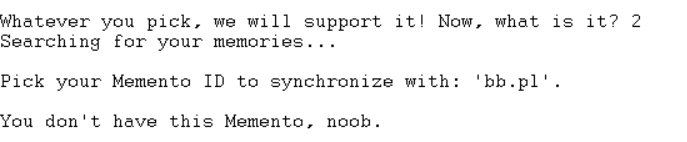


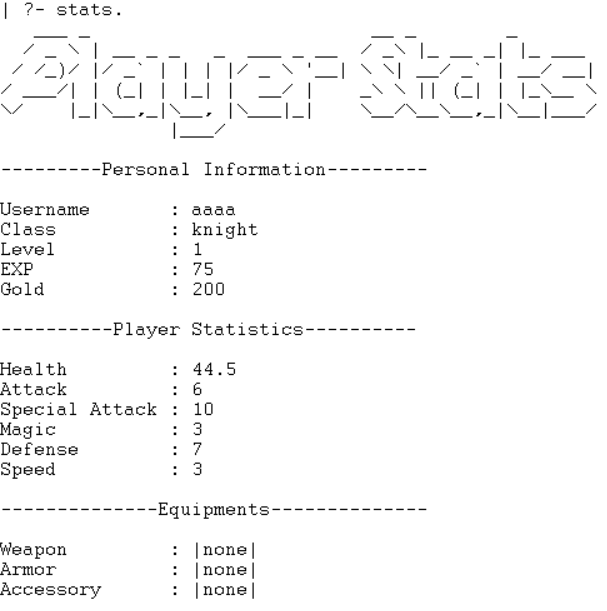
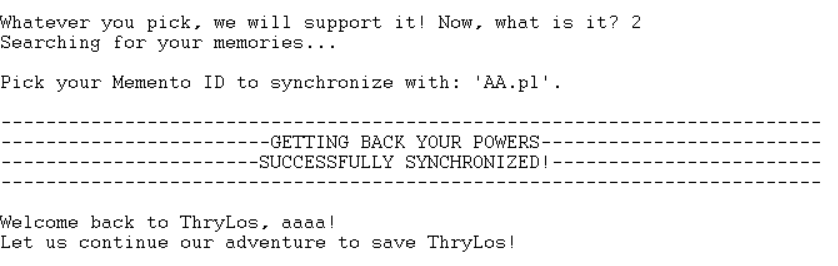
Jika pada *region* kedua hanya ada sebuah teleportasi, maka pada *region* ketiga diberikan dua buah teleportasi serta dengan dinding dalam yang lebih banyak. Di *region* ketiga ini, pemain juga bisa melihat adanya *Dungeon* yang berisi *final boss* dari game ini. *Monster* yang bisa muncul dalam *region* ketiga ini adalah *spider* dan *zool*, dengan *spider* memiliki peluang *spawn* yang lebih tinggi. 

1. **Fungsionalitas Save dan Load**

Selain ketiga bonus sebelumnya, juga berhasil diimplementasikan fungsionalitas *Save* dan *Load* untuk menyimpan dan memainkan kembali *state* permainan. Secara umum, hal-hal yang disimpan di dalam *file* pemain adalah fakta-fakta dinamis yang berhubungan langsung dengan *stats* dan *progress* pemain, seperti posisi pemain, *progress* dalam *quest*, serta *cooldown special attack* yang dimiliki oleh pemain. File disimpan dan dimuat kembali hanya jika berada dalam satu direktori yang sama dengan **main.pl**.

 Adapun fungsionalitas *save* sendiri telah diintegrasikan bersama dengan fungsionalitas kendali dasar *quit*, sehingga pemain bisa menentukan apakah ingin menyimpan *progress* yang dimilikinya atau tidak ketika memutuskan untuk keluar dari permainan. Ketika pemain memutuskan untuk tidak menyimpan *progress* permainan, maka *progress* tersebut akan hilang bersamaan dengan dilakukannya fungsionalitas dari *quit*. Jika sebaliknya, pemain harus memasukkan nama file sebagai tempat penyimpanannya. Jika ada file yang bernama sama, program akan secara otomatis melakukan *overwrite* ke dalam file tersebut. Adapun *progress* yang disimpan berada dalam format *listing* predikat-predikat program yang dinamis dan diperlukan dalam permainan berikutnya ketika melakukan *load*, seperti predikat **started/1**.

Fungsi *load* sendiri dilakukan dengan membaca file yang telah disimpan dan kemudian memasukkan setiap baris fakta yang ada dalam file tersebut ke dalam sebuah *List*. Dari *List* tersebut, kemudian setiap *Head* akan dilakukan proses *asserta* untuk memasukkan kembali *progress* pemain ke dalam permainan. Jika ternyata pemain memasukkan file yang tidak ada, maka akan dikeluarkan pesan kesalahan untuk pemain. Fungsionalitas ini bisa dilakukan melalui Main Menu atau melalui tampilan awal permainan.

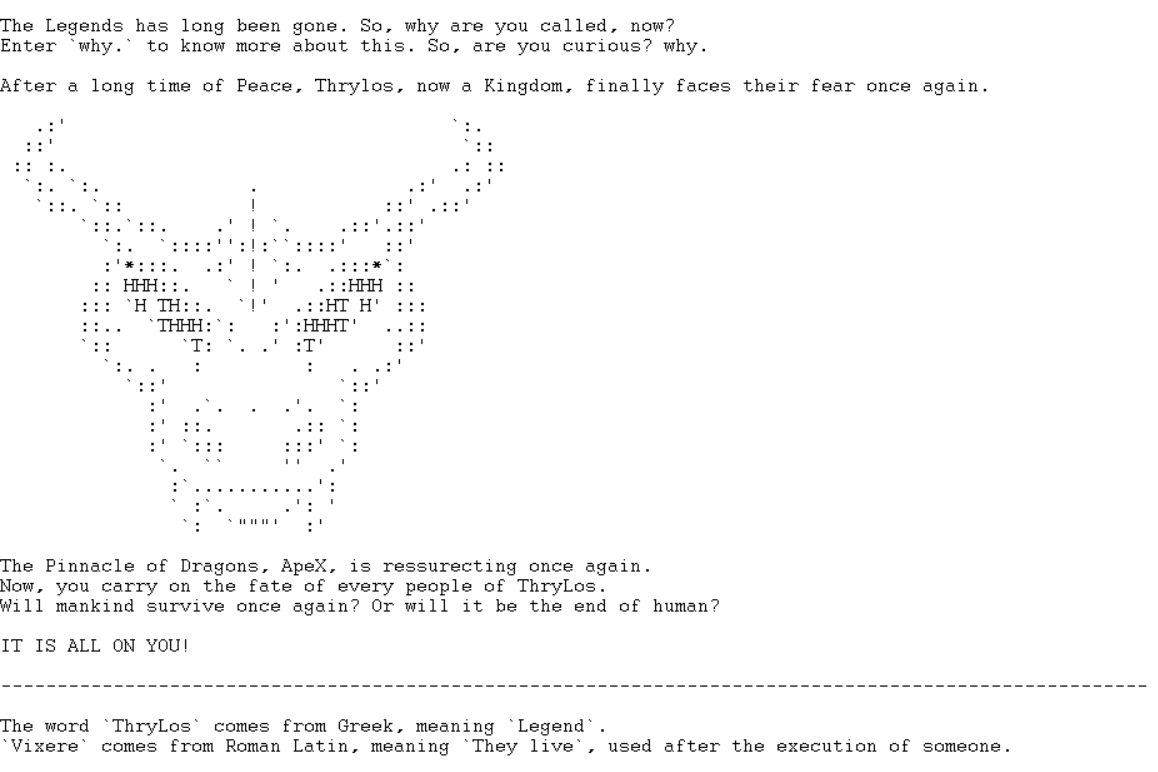
Setelah dilakukan *load*, *stats* dan posisi pemain dimasukkan kembali ke dalam *game*:

1. **Kreativitas dalam Game**

Aspek-aspek di dalam permainan diberi nama melalui berbagai *research*, terutama yang berkaitan dengan penamaan-penamaan di dalam game. Misalnya, judul dari permainan ini, yaitu ‘**ThryLos**’, diambil dari bahasa Yunani yang berarti *legends*, dan saat memulai permainan, ada kata ‘Vixere’ yang berasal dari bahasa Latin yang sering digunakan pada saat seseorang dieksekusi (untuk menggambarkan tema dari spesifikasi awal tugas besar yang diberikan).

 Berkaitan dengan hal tersebut, cerita dari game yang telah dibuat bisa diakses pada bagian **WikiLog** ataupun secara langsung masuk ke dalam Menu **About** melalui Main Menu. Selain itu, beberapa submenu dalam permainan dibuat dengan menggunakan ASCII Art yang didapatkan dari *website patorjk.com*. Contoh dari hal ini dapat dilihat melalui gambar berikut, ataupun dari menu-menu lain yang ada di dalam permainan.

Teks-teks yang dicetak di dalam permainan, seperti ketika meminta input atau menampilkan informasi-informasi kepada pemain dibuat sedemikian rupa sehingga seolah-olah sistem permainan juga berinteraksi dengan pemain. Selain itu, untuk setiap karakter dalam *game* juga dibuat ASCII Art yang bisa dilihat ketika pembuatan karakter dengan kelas tertentu berhasil diaktifkan. ASCII Art yang didapat dari *asciiart.eu* ini juga diedit kembali dan digunakan pada saat pemain bertemu dengan *monster* seperti yang bisa dilihat pada bagian-bagian sebelumnya. Berikut merupakan salah satu contoh ASCII Art yang didapat pada saat pemilihan kelas.

Beberapa tambahan lain yang terdapat dalam permainan ini misalnya adalah penamaan item-item dalam game yang diambil dari nama-nama berbagai mitologi, pengelompokkan item berdasarkan *grade* dengan penambahan *stats* yang berbeda, penamaan *region* peta yang diberikan sesuai dengan konteks peta (*Lost Evergreen* sebagai hutan yang telah hilang, *DragonLance* sebagai *region* tempat adanya *Dungeon Boss*), sistem Gacha yang menghadirkan beberapa fitur Gacha dengan persentase yang berbeda-beda, serta adanya *hidden class* dan *stats* tambahan pada karakter, dan lain sebagainya.

**Sumber Pembelajaran**

[*http://www.gprolog.org/manual/gprolog.html*](http://www.gprolog.org/manual/gprolog.html)*.*

[*http://www.learnprolognow.org/*](http://www.learnprolognow.org/)*.*

**GitHub**

[*https://github.com/RichardRivaldo/IF2121-Logika-Komputasional*](https://github.com/RichardRivaldo/IF2121-Logika-Komputasional)*.*

**Persentase dan Pembagian Kerja Kelompok**

|  |  |  |  |
| --- | --- | --- | --- |
| Nama | NIM | Bagian yang Dikerjakan | Persentase |
| Michael P. G. | 13519121 | store.pl, sebagian battle.pl, laporan | 20% |
| Maximillian Lukman | 13519153 | enemy.pl, battle.pl, sedikit inventory.pl, laporan | 20% |
| David Owen A. | 13519169 | quest.pl, balancing, laporan | 20% |
| Richard Rivaldo | 13519185 | character.pl, inventory.pl, items.pl, leveling.pl, map.pl, main.pl, Bonus 1-4, beberapa bagian store.pl, quest.pl, enemy.pl, dan battle.pl, laporan | 40% |