LAPORAN TUGAS BESAR I

IF2032Pemrograman Berorientasi Objek

Simulasi Adu Monster di Kota Poke

Pejuang Kemerdekaan

Dipersiapkan oleh:

Kelompok 14

13511025 Lubis Sucipto

13511037 Muhammad Rizky W

13511051 Andreas Dwi Nugroho

13511071 Setyo Legowo

13511083 Fawwaz Muhammad

13511093 Asep Saepudin

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Sekolah Teknik Elektro dan Informatika ITB** | Nomor Dokumen | Halaman |
| *IF2032-TB1-14-01* | *<jml hlm>* |
| *25 Maret 2013* |

Daftar Isi

[1 Ringkasan 3](#_Toc287857359)

[2 Gameplay 3](#_Toc287857372)

[3 Penjelasan Tambahan Spesifikasi Tugas 6](#_Toc287857360)

[3.1 <Spesifikasi Fitur Tambahan 1> 3](#_Toc287857361)

[3.2 <Spesifikasi Fitur Tambahan 2> 3](#_Toc287857362)

[4 Diagram Kelas 6](#_Toc287857363)

[5 Penjelasan Kelas 9](#_Toc287857364)

[6 Program Utama 23](#_Toc287857365)

[7 Data Test 24](#_Toc287857369)

[7.1 <Data Test 1> 24](#_Toc287857370)

[7.2 <Data Test 2> 24](#_Toc287857371)

[8 Pembagian Kerja dalam Kelompok 25](#_Toc287857373)

[9 Lampiran 25](#_Toc287857374)

[9.1 Notulen Rapat 25](#_Toc287857375)

[9.2 Log Activity Anggota Kelompok 26](#_Toc287857376)

# Ringkasan

Permainan PokeRanch yang dibuat dalam tugas besar ini mengambil jalan cerita dan bentuk permainan dari permainan yang dikenalkan dengan *console* terkenal sebelumnya yaitu Pokemon® dan Monster Rancher. Cerita utama dari permainan ini adalah seorang pemain yang memelihara satu atau lebih monster yang dapat ditarungkan dengan monster lain atau pemain lain yang lebih kuat. Cerita utama dari permainan Pokemon® yang diambil untuk permainan ini adalah melengkapi seluruh spesies monster yang ada dalam permainan, cara bertarung, dan menangkap monster liar. Cerita utama dari permainan Monster Rancher yang diambil untuk permainan ini adalah meningkatkan monster yang ditangkap agar lebih kuat, monster pemain dipertarungkan dengan monster yang lebih kuat di Stadium, dan berkeliling di area luar. Dari dua permainan tersebut memiliki satu kemiripan yaitu bertarung dengan monster liar di area luar.

Dalam laporan ini dicoba untuk menjelaskan deskripsi dari permainan ini secara menyeluruh diantaranya alur cerita permainan, spesifikasi minimum yang dicapai, spesifikasi tambahan, catatan asistensi, dan pembagian tugas (catatan aktivitas pengerjaan tugas).

# Gameplay

Saat permainan PokeRanch dijalankan, pengguna dihadapkan pada main menu yang dapat digunakan sebagai berikut:

|  |
| --- |
| new <username>  Perintah ini digunakan untuk menjadikan pengguna sebagai pemain baru dari permainan PokeRanch. Namun, belum menjadi pemain yang terdaftar dalam permainan ini. |

|  |
| --- |
| load <username>  Perintah ini digunakan untuk mengambil pengguna yang telah terdaftar sebelumnya sebagai pemain aktif. |

Ada pengenalan khusus untuk pemain baru atau yang memberikan perintah newpada program ini, yaitu pemain baru akan diberikan semacam monolog yang menceritakan bahwa pemain baru adalah seorang pendatang ke dunia Poke. Pendatang baru tersebut dapat berteman dengan Poke (monster) dan monster tersebut dapat dipertarungkan dengan monster lain. Lalu diberikan tujuan akhir dari permainan ini yaitu mengumpulkan seluruh monster dan bertarung dengan trainer (pemain di dalam permainan) yang sangat kuat. Setelah itu, jika pemain melanjutkan hingga sampai di dalam rumah maka data-data awal pemain akan diset (dibuat arsip eksternalnya).

Jika pemain lama, pengantar permainan tidak akan ditampilkan. Pemain baru akan diminta memasukkan nama pemain (memasukkan nama pemain hanya bisa dilakukan di tahap ini saja) dan waktu diatur siang untuk pertama kali.

Setelah itu pemain (baru maupun lama) akan berada di dalam rumah. Di dalam rumah sendiri terdapat 4 fasilitas:

* Terdapat fasilitas untuk tidur dengan menggunakan perintah

|  |
| --- |
| #sleep |

Perintah ini digunakan untuk mengembalikan HP ke HP Maksimum dan berpindah waktu dari siang ke malam ataupun sebaliknya.

* Terdapat fasilitas untuk menyimpan permainan dengan menggunakan perintah

|  |
| --- |
| #save |

Perintah ini digunakan untuk menyimpan State permainan. Hal-hal yang disimpan antara lain:

* + Waktu permainan (siang/malam)
  + Jumlah hari bermain (bukan waktu OS)
  + Jumlah uang sekarang (mata uang yang digunakan adalah Zeny)
  + Jumlah menang/kalah dalam pertarungan di stadium.
  + Status monster party (monster yang digunakan untuk bertarung)
  + Daftar Item yang dimiliki
  + Daftar monster yang pernah ditangkap.
  + Dan lain sebagainya
* Terdapat fasilitas untuk menggunakan pemain lain dengan menggunakan perintah:

|  |
| --- |
| #load <nama-file>  Perintah ini digunakan untuk mengembalikan State permainan ke State terakhir yang disimpan atau menggunakan nama pemain lain. |

Setelah itu pemain dapat keluar dari rumah. Untuk keluar rumah dapat menggunakan perintah sebagai berikut:

|  |
| --- |
| #out  Perintah ini digunakan untuk keluar dari suatu bangunan |

Setelah berada di jalan, pemain dapat bergerak keliling kota. Setiap bangunan memiliki teleportnya masing-masing sehingga dari bangunan satu ke bangunan lain dapat berpindah secara langsung menggunakan teleport. Untuk menggunakan teleport dapat menggunakan perintah sebagai berikut:

|  |
| --- |
| #teleport <nama-tempat>  Perintah ini digunakan untuk berpindah-pindah dari satu bangunan ke bangunan lain. Posisi teleport ada di depan bangunan. Nama bangunan yang diterima sebagai berikut:   * home   Rumah pemain   * combinatorium   Tempat untuk mengkombinasikan monster   * stadium   Titik untuk bertarung dengan pemain lain   * shop   Titik untuk beli item yang bisa dibeli dan jual item yang bisa dijual   * gate   Titik untuk menuju ke OuterArea. (Tapi Gate bukan bangunan) |

Pemain juga dapat berpindah tempat menggunakan perintah sebagai berikut:

|  |
| --- |
| #move <arah> [jumlah langkah]  Perintah ini digunakan untuk berpindah ke arah tertentu (atas, kanan, bawah, kiri) dan jumlah langkah yang akan diambil. Jika jumlah langkah tidak dimasukkan maka langkah yang diambil sebanyak 1 langkah. |

Peintah perintah yang dapat digunakan di segala tempat setelah menjadi pemain baru adalah:

* list-monster

Menampilkan seluruh monster yang pernah ditangkap. Dengan menggunakan perintah ini dapat melihat status monster (Hanya HP dan CurLevel).

* list-party

Menampilkan monster-monster yang digunakan untuk battle. Dengan menggunakan perintah ini dapat menampilkan status monster (HP, Status Effect, Skills, Level).

* list-item

Menampilkan list dari item-item yang dibeli. Dapat juga melihat deskripsi dari tipe item.

* status

Menampilkan jumlah uang dan jumlah menang dan jumlah kalah dalam battle stadium.

* set <no-monster>

Digunakan untuk memilih monster yang akan digunakan dari list monster yang tidak digunakan ke list monster party urutan pertama.

Pemain

* switch <no-party><no-party>

Digunakan untuk mengganti urutan dari Monster Party

* dismiss <no-monster>

Digunakan untuk membuang monster dari list-monster. Jika monster yang digunakan terdapat dalam party maka list monster dalam party berkurang.

* help

menampilkan daftar command yang bias digunakan.

Saat pemain berada pada mode battle, command yang bisa digunakan antara lain:

* escape

Perintah untuk keluar dari pertarungan saat di OuterArea. Tidak dapat digunakan saat pertandingan stadium.

* list-party

Sama dengan deskripsi sebelumnya.

* item <nama-item><no-party>

Untuk mengunakan item yang digunakan saat battle.

* change <no-party>

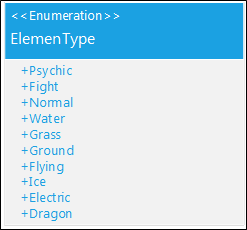
Untuk mengganti pokemon yang battle dari list-party

* skill <no-skill>

Untuk menggunakan skill dari monster yang sedang battle.

# Penjelasan Tambahan Spesifikasi Tugas

Elemen Monster



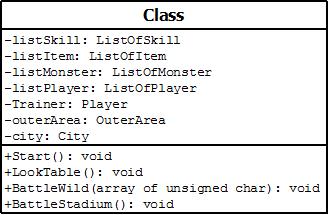
## <Spesifikasi Fitur Tambahan 1>

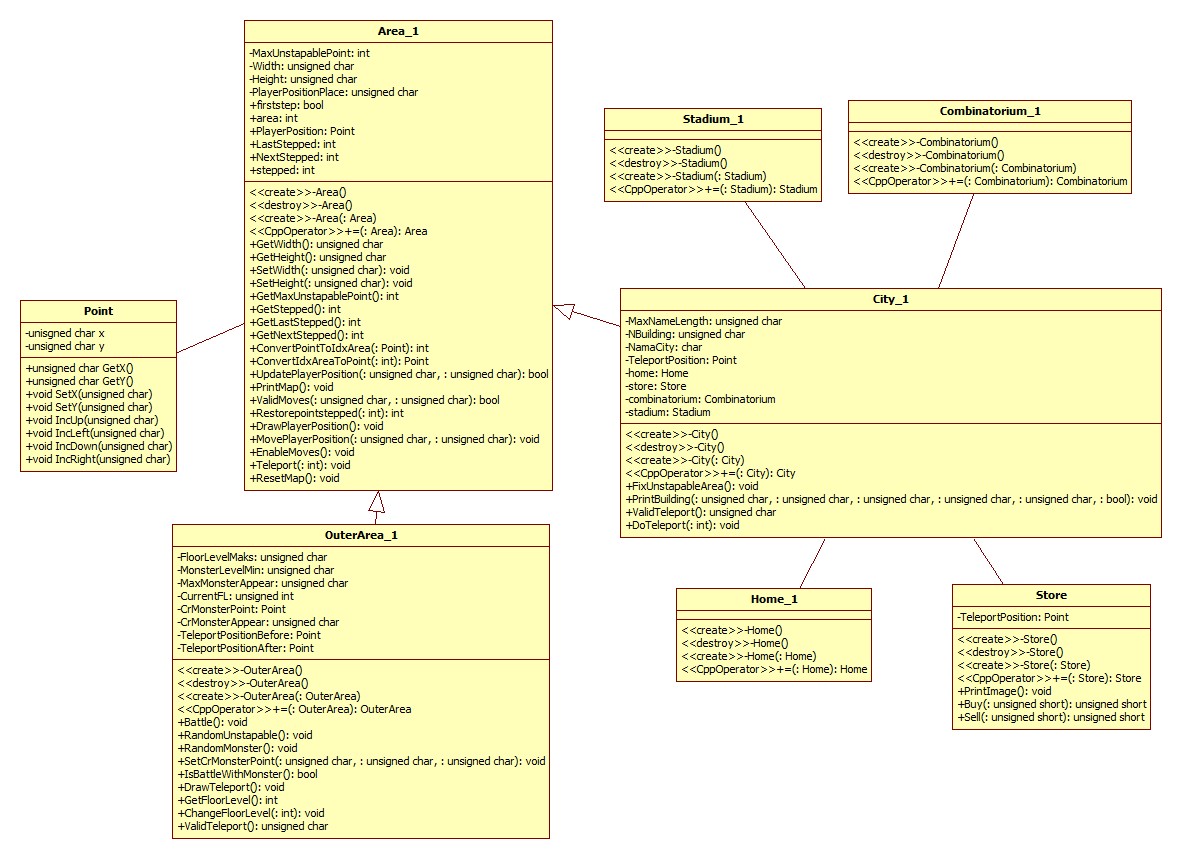
Isi dengan penjelasan spesifikasi fitur tambahan 1.

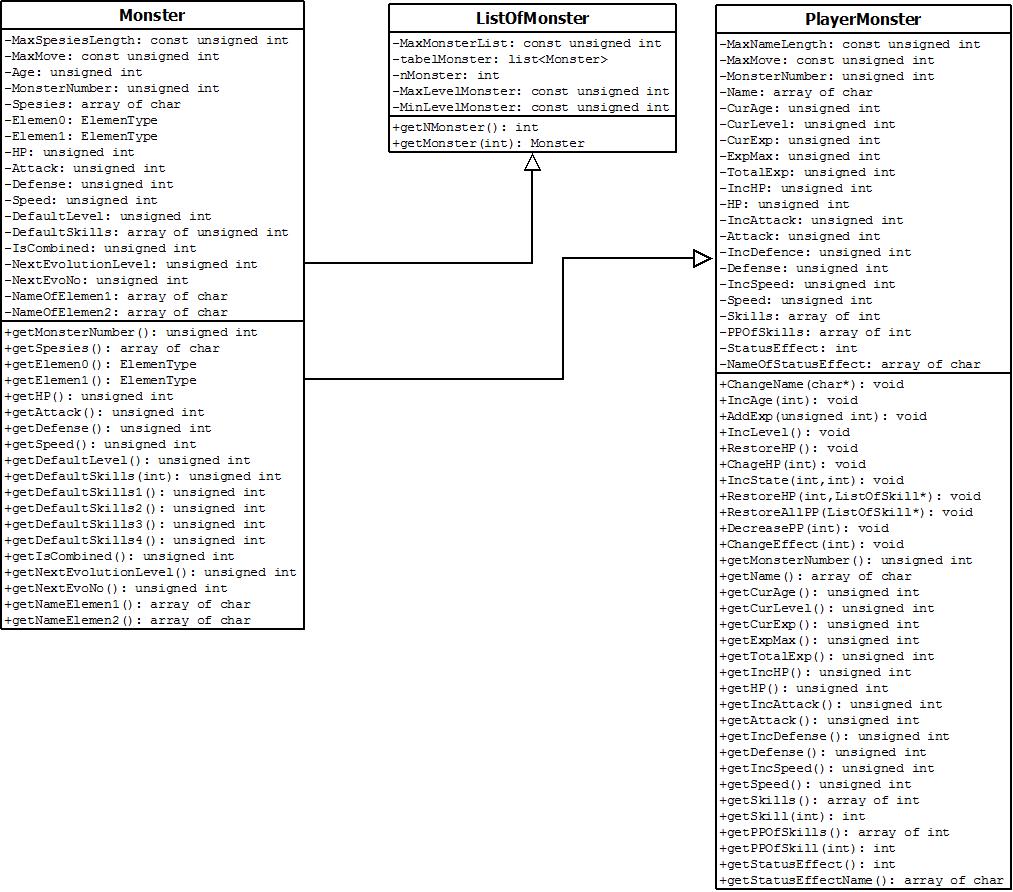
## <Spesifikasi FiturTambahan 2>

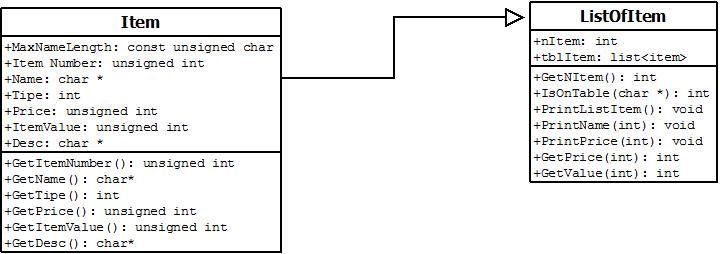
Isi dengan penjelasan spesifikasi fitur tambahan 2. (tambahkan poin spesifikasi fitur tambahan sesuai kebutuhan)

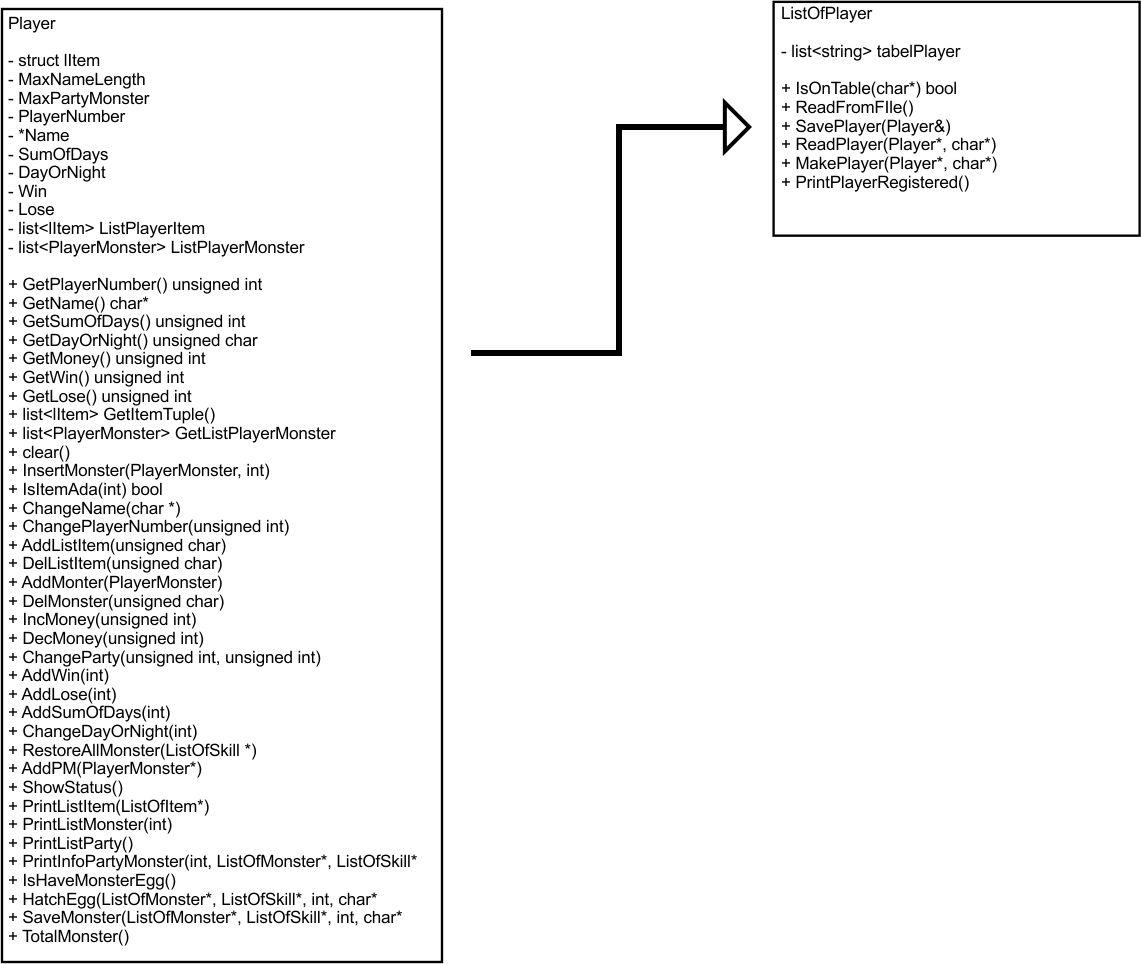
# Diagram Kelas

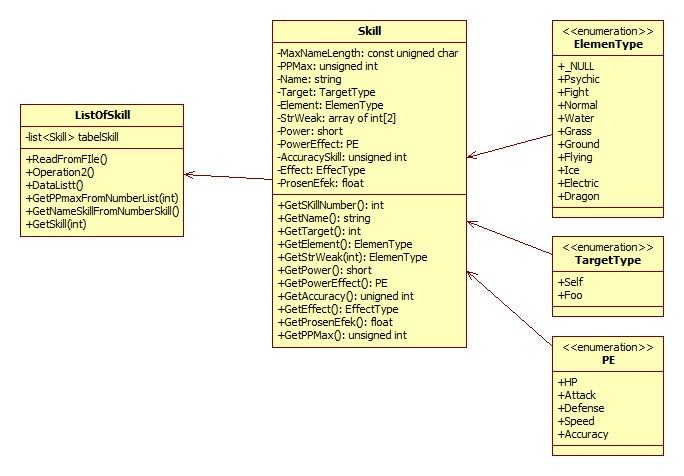












# Penjelasan Kelas

* **PokeRanch**

Kelas Utama yang mengontrol semua penciptaan objek-objek untuk permainan PokeRanch

* listItem: ListOfItem
* listMonster: ListOfMonster
* listPlayer: LostOfPlayer
* listSkill: ListOfSkill
* area: Area
* Inisialisasi(): void

Membaca file ekternal, dan pengambilan database ke objek kelas

* BeginStory(): void

Memulai cerita PokeRanch

* ReadInput(:\*char,:char,:char): void

Membaca Input command di PokeRanch

* InputExecution(:char):void

Menjalankan input command di PokeRanch

* **ListOfSkill**

Class ListOfSkill adalah kelas yang berisi semua skill atau penampung kelas skill yang ada di PokeRanch dan method-method yang digunakan dalam pemrosesan skill

* + - * + Atribut
* tabelSkill:List<Skill>

Berisi List semua skill yang ada di PokeRanch

* + - * + Method
* ReadFromFile():void

Membaca file eksternal skill.txt dan memasukkan ke dalam atribut tabelSkill

* PrintList():void

Menuliskan semua isi tabelSkill

* DataList(int): void

Menampilkan semua atribut skill dengan nomor skill yang sesuai dengan inputan

* GetPPMaxFromNumberSkill(int): unsigned int

mengembalikan PPMax suatu skill yang NumberSkillnya sama dengan inputan

* GetNameSkillFromNumberSkill(int): string

mengembalikan NameSkill suatu skill yang NumberSkillnya sama dengan inputan

* GetSkill(int): Skill

Mengembalikan kelas Skill dengan NumberSkill sesuai inputan

* **Skill**

Class Skill adalah kelas yang berisi informasi sebuah skill dengan atribut dan method-method:

* + - * + Atribut
* MaxNameLength: unsigned char = 10

Constanta panjang maksimal nama suatu skill yaitu 10 character

* PPMax: unsigned char

Maksimal penggunaan suatu skill sebelum direstore kembali. Misal PPMax=50 maka skill tersebut harus di restore kembali stetelah digunakan 50 kali dalam batle

* SkillNumber:unsigned int

Nomor ID dari sebuah skill, untuk setiap skill memiliki SikllNumber berbeda-beda

* Name:array of char[MaxNameLength]

Nama suatu skill

* Target:TargetType

Type target suatu skill terdiri enumerasi dari TargetType(Self,Foo). Self untuk skill yang diapakai sendiri, dan Foo untuk ke musuh.

* Element: ElemenType

Type elemen suatu skill terdiri enumarasi dari ElemenType(Psychic, Fight, Normal, Water, Grass, Ground, Flying, Ice, Electric, Dragon)

* StrWeak: array of int[2]

Tabel yang terdiri dari kekuatan dan kelemahan elemen skill. Misal skill yang berlemen Psychic 2 kali damage terhadap monster berlemen Fight dan 0.5 kali damage terhadap Normal

|  |  |  |
| --- | --- | --- |
| Elemen Type | Target 2X damage | Target 0.5X damage |
| Psychic | Fight | Normal |
| Fight | Normal | Psychic |
| Normal | Psychic | Fight |
| Fly | Ground | Grass |
| Ground | Electric | Fly |
| Electric | Water | Ground |
| Water | Ice | Electric |
| Ice | Grass | Water |
| Grass | Fly | Ice |
| Dragon | Dragon | NULL |

* Power: short

Besar dari efek skill. Misal Skill Health dengan Power=50 maka akan menambah HP sebesar 50

* PowerEffect: PE

Effect yang akan diterima dari suatu skill.Enumerasi dari PE(HP, Attack, Defense, Speed, Accuracy). Misal Atack maka skill akan berefek pada Atack monster.

* Accurancy: unsigned int

Besar akurasi suatu skill terhadap monster.

* Effect: EffectType

Effect suatu skill. Terdiri enumerasi dari EffectType(NoEffect, Paralyz, Poison, BadPoison, Burn, Freze, Sleep)

* ProsenEffect: float

Besar presentasi kemungkinan suatu Effect Skill

* + - * + Method
* GetSkillNumber():int

Getter SkillNumber

* GetName():string

Getter Name

* GetTarget(): int

Getter Target

* GetElement():ElemenType

Getter Element

* GetStrWeak(int):ElemenType

Getter StrWeak dengan input 0 untuk Str dan 1 untuk Weak

* GetPower():short

Getter Power

* PowerEffect():PE

Getter PowerEffect

* GetAccuracy():unsigned int

Getter Accuracy

* GetEffect()EffectType

Getter Effect

* GetProsenEfek():float

Getter ProsenEfek

* GetPPMax()unsigned int

Getter PPMax

* **Player**

Class Player berisi informasi umum mengenai pemain pada game

* Atribut:
* lItem: Berisi noItem yang menunjukkan jumlah item, dan nItem yang menunjukkan nama item.
* MaxNameLength: Bernilai 7, menunjukkan jumlah karakter maksimal yang bisa dipakai untuk memberi nama pada player
* MaxPartyMonster: Bernilai 6, fungsinya untuk menunjukkan jumlah monster maksimal yang bisa dimasukkan dalam party
* PlayerNumber: Berisi nomor pemain pada game
* Name: Berisi nama pemain pada game
* SumOfDays: Berisi jumlah hari yang telah dilalui seorang player pada game. Merupakan parameter waktu pada game
* DayOrNight: Berisi informasi berupa keadaan lingkungan pada game, 0 menunjukkan siang sedangkan 1 menunjukkan malam
* Money: Menunjukkan jumlah uang yang dimiliki pemain
* Win: Menunjukkan jumlah kemenangan pemain di stadion
* Lose: Menunjukkan jumlah kekalahan pemain di stadion
* ListPlayerItem: List yang berisikan lItem, berfungsi untuk menyimpan item yang dimiliki oleh pemain
* ListPlayerMonster: List yang berisikan Monster, berfungsi untuk menyimpan monster yang dimiliki oleh pemain
* Method:
* GetPlayerNumber(): Mengembalikan nomor pemain
* GetName(): Mengembalikan nama pemain
* GetSumOfDays(): Mengembalikan jumlah hari
* GetDayOrNight(): Mengembalikan nilai DayOrNight
* GetMoney(): Mengembalikan nilai Money
* GetWin(): Mengembalikan nilai Win
* GetLose(): Mengembalikan nilai Lose
* GetItemTuple(): Mengembalikan nilai ListPlayerItem
* GetListPlayerMonster(): Mengembalikan nilai ListPlayerMonster
* Clear(): Menghapus semua data Player
* IsItemAda(int): Memeriksa apakah ada item bernomor sesuai input pada ListPlayerItem
* ChangeName(char\*): Mengubah nama pemain
* ChangePlayerNumber(unsigned int): Mengubah nomor pemain
* AddListItem(unsigned char, unsigned char): Menambahkan Item pada ListPlayerItem
* DelListItem(unsigned char, unsigned char): Menghapus Item pada ListPlayerItem
* AddMonster(PlayerMonster): Menambahkan Monster pada ListPlayerMonster
* DelMonster(unsigned char): Menghapus Monster pada ListPlayerMonster
* IncMoney(unsigned int): Menambah nilai Money sesuai dengan input yang dimasukkan
* DecMoney(unsigned int): Mengurangi nilai Money sesuai dengan input yang dimasukkan
* ChangeParty(unsigned int): Mengubah Party Pemain
* AddWin(int): Menambah nilai Win pemain di stadion
* AddLose(int): Menambah nilai Lose pemain di stadion
* AddSumOfDays(int): Menambah nilai SumOfDays sesuai dengan input
* ChangeDayOrNight(int): Mengubah nilai DayOrNight sesuai dengan input
* RestoreAllMonster(ListOfSkill \*): Mengembalikan status semua monster di ListPlayerMonster
* AddPM(PlayerMonster\*): Menambah PlayerMonster pada ListPlayerMonster
* ShowStatus: Menghasilkan output berupa status pemain yang terdiri dari Nama (Name), Uang (Money), Total Menang (Win), Total Kalah (Lose), Jumlah Hari Main (SumOfDays), Status Hari Ini (DayOrNight)
* PrintListItem(ListOfItem \*): Menghasilkan output berupa daftar Item yang dimiliki oleh pemain
* PrintListMonster(int): Menghasilkan output berupa daftar monster yang dimiliki oleh pemain
* PrintListParty(): Menghasilkan output berupa daftar monster yang terdapat pada *party* pemain
* PrintInfoPartyMonster(int, ListOfMonster\*, ListOfSkill\*): Menghasilkan output berupa info monster yang ada pada *party* pemain yang terdiri dari Nomor Monster(MonsterNumber), Nama (Name), Level (CurLevel), Spesies (Spesies), Stat Efek (StatusEffect), Elemen 1, Elemen2, Cur Exp (CurExp), HP(CurHP), Serangan (Attack), Pertahanan (Defense), Kecepatan (Speed), Umur (Age), Total Exp (TotalExp)
* IsHaveMonsterEgg(): Memeriksa apakah pemain mempunyai item MonsterEgg
* HatchEgg(ListOfMonster \*, ListOfSkill \*, int, char \*): “Memecahkan” MonsterEgg, mengubah MonsterEgg yang terdapat pada ListPlayerItem menjadi Monster baru untuk selanjutnya ditambahkan pada ListPlayerMonster
* TotalMonster(): Menghasilkan output berupa total monster yang dimiliki oleh pemain
* **ListOfPlayer**

Class ListOfPlayer merupakan kelas yang berfungsi untuk menampung semua data player pada game PokeRanch

* Atribut:
* tabelPlayer: list<Player>

Berfungsi untuk menampung semua data player

* Metode
* IsOnTable(char \*): Mencari apakah player dengan nama yang di input ada pada listofplayer
* ReadFromFile(): Membaca file player.txt
* SavePlayer(Player \*, char \*): Menyimpan data pemain
* ReadPlayer(Player \*, char \*): Membaca file player dari file player.txt
* MakePlayer(Player \*, char \*): Membuat player baru
* PrintPlayerRegistered():
* **AREA**

Class area memiliki perubahan yang cukup besar dari rancangan class mula-mula terdapat fungsi-fungsi yang bertambah, dipindahkan (pemilik fungsinya berubah), dan penghapusan fungsi.

1. Class Outer Area

Class outer area menggambarkan dunia luar selain kota. Kami mengasumsikan bahwa outer area dapat dianggap seperti bangunan bertingkat. setiap tingkatan mengandung monster-monster yang bervariasi levelnya bergantung dari "lantai" levelnya. misal untuk lantai 1, monster yang ada disana hanya ada monster dengan level 1-5, lalu di lantai 2, monster yang ada hanya monster dengan level 5-10, dan seterusnya. Kami menganggap untuk masuk ke "lantai" yang lebih tinggi kita harus masuk dari "lantai" yang lebih rendah, begitu juga untuk keluar dari outer area. Misal kita ingin masuk ke lantai 3, berarti kita harus masuk ke lantai 1, lalu lantai 2 baru lantai 3. sementara untuk kembali ke kota, kita harus teleport dari lantai 3, lalu ke lantai 2, ke lantai 1 baru ke kota.

Untuk mendukung asumsi-asumsi tersebut kami membuat beberapa atribut/properties/data member untuk class outer area:

* Floor Level Max : "konstanta" yang dapat di configurable menunjukan level maksimum "lantai" yang ada di game
* Monter Level Min : Menunjukan level monster terendah yang mungkin ditemukan di "lantai" saat ini default: 10n+x; n menunjukan lantai saat ini, x level minimal configurable.
* Max Monster Appear: Jumlah maksimum monster yang muncul di area, default 10.
* CurrentFL: lokasi lantai saat ini yang dipijaki oleh pemain
* CrMonsterPoint: Koordinat dirandom untuk lokasi monster-monster yang bisa ditemukan di area saat ini
* Teleport Position Before: Koordinat tempat untuk melakukan teleport untuk ke "lantai" sebelumnya.
* Teleport Position Next : Koordinat tempat untuk melakukan teleport untuk ke "lantai" setelahnya.

Class ini memiliki beberapa function member/methodyaitu:

* Battle() memanggil function untuk masuk ke mode battle screen
* Random unstepable: mengacak koordinat-koordinat yang unstapable
* Random monster : mengacak monster-monster yang muncul.
* Print map: mencetak area ke layar.

Pada akhirnya class area menggunakan sebuah array satu dimensi untuk merepresentasikan kondisi layar, indeks nilai yang berada dalam array merepresentasikan apakah di koordinat tersebut ada unstapable/monster atau hal lainya(teleport spot)

Terdapat funsi yang bertambah yaitu :

* Setter Getter seiap atribut
* Update player position : mengupdate posisi player
* Print map : mencetak gambar map ke layar
* Valid moves : validasi apakah movement valid atau tidak
* Restore point stepped : mengembalikan kondisi peta yang baru saja “diinjak” oleh pemain ke kondisi normal
* DrawPlayer position : menggambar posisi player di peta
* Move player position : Memindahkan posisi player secara alami (tidak seperti jin yang kalau ada tembok bisa nimbus dinding)
* enable moves : mengembalikan kondisi agar movement bisa diakses karena mungkin di tengah pemanggilan di lock (karena battle dengan monster)
* Teleport : Melakukan teleport

1. Class City

Kami mengasumsikan kota memiliki ukurang 39x9 dot sesuai dengan spek di dokumen.Sebagaimana City pada umumnya, City memiliki beberapa properties:

* Max NameLEngth : konstanta configurable panjang nama maksimum
* NBangunan : konstanta configurable jumlah bangunan yang ada di kota
* City name: Nama kota
* Teleport position : Lokasi dalam koordinat untuk melakukan teleport entah itu ke outer area atau ke tempat lain.
* Buildings : array of Building, yang merepresentaiskan bangunan yang ada di kota ini.

Sebagaimana class outer area, class ini memiiliki fungsi print untuk mencetak kondisi area saat ini (koordinat pemain, rumah dll) dan fixunstapable area untuk memberi array koordinat point fix yang unstappable di City.

Class city akan digunakan oleh beberapa kelas lainnya:

* 1. Home :

Properties:

* Teleport position: lokasi koordinat teleport

Function member:

* + - Print images : Menampilkan gambar rumah ke layar
    - Sleep : memanggil fungsi sleep agar pemain dapat "tidur" dan mengganti waktu siang jadi malam/ malam jadi siang
    - Save : Melakukan save game
    - Load : Melakukan load game
    - New : MElakukan new game
  1. Combinatorium:

Properties:

* Teleport position: lokasi koordinat teleport

Function member:

* Combine: untuk melakukan combine antar monster
  1. Store:

Properties:

* Teleport position: lokasi koordinat teleport

Function member:

* + - Printimage: menampilkan gambar toko ke layar
    - Buy:membeli item tertentu
    - Sell: menjual item tertentu
  1. Stadium:

Properties:

* Teleport position: lokasi koordinat teleport

Function member:

* + - Battle: untuk melakukan battle antar musuh,pada battle di stadium, user diharuskan bertanding dengan 3 monster berturut-turut bukan seperti di outer area yang hanya 1 kali
    - Random Monster : untuk merandom monster yang muncul di stadium dan unstapable point

Pada akhirnya class City memiliki fungsi fungsi yang dipindahkan ke class induk seperti print map dll karena dianggap lebih universal dan tidak perlu diimplementasikan berbeda dengan class outer area. Terdapat fungsi fungsi sbb:

* Print building : melakukan print building ke layar
* Valid Teleport : mengembalikan nilai apakah valid atau tidak melakukan teleport di koordiinat pemain berada
* Do teleport : melakuan teleport ke spot-spot tertentu seperti toko,rumah, stadium,(sudah dilengkapi validasi).

Sebenarnya seharusnya saya membuat properties dan fungsi setter/getter width, height,offset x, offset y ke setiap kelasbuilding(home,store,stadium,combinatorium) bukan sebagai sebuah konstanta di dalam kelas city.h(agar lebih OOP), tapi karena waktu tak sempat maka lebih baik memenuhi spek minimum terlebih dahulu.

Adapun class Area sendiri digunakan oleh class city dan outer area sehingga kedua kelas ini meng-inherit dari class area. class area sendiri terdiri dari beberapa properties yaitu :

* Max unstapable point : jumlah maksimum point yang unstapable dalam map saat ini
* Width :lebar map default : 39
* Height : tinggi map default : 9
* Unstapable P: Koordinat-koordinat (Array) yang tidak bisa distep
* Unstapable active ;Point-point yang stepable / active / boleh dipijak
* Player position point: koordinat player dalam peta saat ini
* Player position place : Lokasi player berada misal: di rumah/ di lcombinatorium / di stadium
* **ListOfMonster**

Kelas ini merupakan yang menyimpan data kumpulan monster yang ada di PokeRanch ini. Kumpulan monster disusun dalam suatu list yang elemennya terdiri dari monster. Selain terdiri dari list monsters, di kelas ini juga tersimpan data maksimal level dan minimum level suatu monster. Berikut adalah atribut dari kelasListOfMonster:

* TabelMonster: List <Monster>

List kumpulan moster.

* MaxLevelMonster: unsigned int= 100

Level tertinggi yang dapat dicapai suatu monster, yaitu level 100.

* MinLevelMonster: unsigned int = 1

Level terendah yang ada di suatu monster, yaitu level 1.

* MaxMonsterList: unsigned int = 110

Jumlah maksimal monster yang ada di list.

* nMonster: int

Atribut yang menyimpan jumlah monster yang ada di list.

* getNMonster(): int

Metode yang akan mengembalikan nilai nMonster.

* getMonster(:int): Monster

Metode yang akan mengembalikan monster yang ada di list ke-int.

* **Monster**

Kelas ini adalah kelas yang akan digunakan dalam menciptakan objek monster. Terdiri dari cukup banyak atribut. Atribut-atribut tersebut adalah

* MaxSpesiesLength: unsigned int = 60

Panjang karakter maksimal untuk nama spesies monster.

* MaxMove: unsigned int= 4

MaxMove adalah jumlah maksimal suatu level dalam memiliki suatu skill, di mana suatu monster tidak akan memiliki dari 4 skill.

* DefaultLevel: unsigned int = 1

Suatu monster apabila belum diapa-apakan, akan memiliki level default yaitu level 1.

* MonsterNumber: unsigned int

MonsterNumber merupakan nomor unik yang akan menjadi identitas suatu monster, selain nama monster.

* Spesies: aray of char[MaxSpesiesLength]

Atribut ini akan menyimpan nama spesies suatu monster. Spesies merupakan pengenal jenis dari monster ini. Jenis spesies akan berubah setelah monster mencapai level tertentu.

* Element0: Element Type

Merupakan tipe monster yang akan berpengaruh pada serangan monster lawan (misal: serangan tipe air akan superior terhadap monster tipe api).

* Element1: ElementType

Monster juga bisa memiliki lebih dari satu elemen.

* HP: unsigned int

Health Point. Nyawa suatu monster. Semakin besar level suatu monster, maka nilai HP-nya semakin besar.

* Attack: unsigned int

Attack. Nilai serangan suatu monster. Semakin besar, maka serangan semakin berbahaya. Nilai attack akan membesar sesuai dengan naiknya level.

* Defense: unsigned int

Defense. Nilai pertahanan suatu monster. Semakin besar, maka pertahanannya semakin tangguh. Nilai defense akan membesar sesuai dengan naiknya level.

* Speed: unsigned int

Speed. Nilai kecepatan suatu monster. Semakin besar, maka moster tersebut semakin gesit. Nilai speed akan membesar sesuai dengan naiknya level.

* DefaultSkill: array of int[MaxMove]

List Skill yang dimiliki oleh suatu monster. Jumlah skill maksimal yang mungkin adalah 4

* NextEvoLevel: unsigned int

NextEvoLevel adalah monster yang berevolusi menjadi suatu monster baru (yang sudah terdefinisi) .

* IsCombined: unsigned int

IsCombined adalah evolusi monster apabila 2 monster bergabung. Bergabungnya dua monster ini akan menghasilkan suatu monster baru (yang sudah terdefini).

* NextEvoNo: unsigned int

NextEvoNo menyimpan nomor monster di mana suatu monster akan bervolusi menjadi monster yang ada di nomor tersebut.

* NameOfElemen1: array of char[10]

Variabel yang akan menampung nama dari elemen 0.

* NameOfElemen2: array o char[10]

Variabel yang akan menampung nama dari elemen 1.

* getMonsterNumber(): unsigned int

Metode yang akan mengembalikan nomor monster.

* getSpesies(): array of char

Metode yang akan mengembalikan nama monster.

* getElemen0():ElemenType

Metode yang akan mengembalikan informasi elemen 0 monster.

* getElemen1():ElemenType

Metode yang akan mengembalikan informasi elemen 1 monster.

* getHP():unsigned int

Metode yang akan mengembalikan nilai HP monster.

* getAttack():unsigned int

Metode yang akan mengembalikan nilai Attack monster.

* getDefense():unsigned int

Metode yang akan mengembalikan nilai Defense monster.

* getSpeed():unsigned int

Metode yang akan mengembalikan nilai Speed monster.

* getDefaultLevel():unsigned int

Metode yang akan mengembalikan nilai level default monster.

* getDefaultSkills(int): unsigned int

Metode yang akan mengembalikan nilai default skill ke-int.

* getDefaultSkills1():unsigned int

Metode yang akan mengembalikan nilai defaul skill 1.

* getDefaultSkills2():unsigned int
* Metode yang akan mengembalikan nilai defaul skill 2.
* getDefaultSkills3():unsigned int
* Metode yang akan mengembalikan nilai defaul skill 3.
* getDefaultSkills4():unsigned int

Metode yang akan mengembalikan nilai defaul skill 4.

* getIsCombined():unsigned int
* Metode yang akan mengembalikan nilai IsCombined monster.
* getNextEvolutionLevel():unsigned int
* Metode yang akan mengembalikan nilai NextEvolutionLevel monster.
* getNextEvoNo():unsigned int

Metode yang akan mengembalikan nilai NextEvoNo monster.

* getNameElemen1(): array of char

Metode yang akan mengembalikan nama elemen 0 monster.

* getNameElemen2(): array of char

Metode yang akan mengembalikan nama elemen 1 monster.

* **PlayerMonster**

Kelas PlayerMoster ini adalah kelas yang akan menhidupkan objek monster yang sudah dimiliki atau sudah diubah-ubah.

* MaxNameLength: unsigned int =15

Panjang maksimal nama monster, yaitu 15.

* MonsterNumbers: unsigned int

Nomor identitas monster.

* Name: array of char[MaxNameLength]

Nama dari monster, pengenal unik monster tersebut. Bisa diisikan oleh pemain.

* Age: unsigned int

Umur dari monster.

* MaxAge: unsigned int = 100

Umur maksimal dari monster, yaitu 100. Apabila sudah mencapai batas ini, monster akan mati.

* CurLevel: unsigned int

Current level. Level yang dimiliki monster pada saat itu.

* CurExp: unsigned int

Current Experience. Nilai experience yang dimiliki oleh monster pada saat itu.

* MaxExp: unsigned int

Nilai maksimal dari Exp.

* TotalExp: unsigned int

Total experience yang dimiliki oleh monster.

* IncHP: unsigned int

Increase HP. Nilai yang akan ditambahkan ke health point monster.

* HP: unsigned int

Health Point. Nyawa suatu monster. Semakin besar level suatu monster, maka nilai HP-nya semakin besar.

* IncAttack: unsigned int

Nilai yang akan ditambahkan ke nilai Attack monster.

* Attack: unsigned int

Attack. Nilai serangan suatu monster. Semakin besar, maka serangan semakin berbahaya. Nilai attack akan membesar sesuai dengan naiknya level.

* IncDefense: unsigned int

Nilai yang akan ditambahkan ke nilai Defense monster.

* Defense: unsigned int

Defense. Nilai pertahanan suatu monster. Semakin besar, maka pertahanannya semakin tangguh. Nilai defense akan membesar sesuai dengan naiknya level.

* IncSpeed: unsigned int

Nilai yang akan ditambahkan ke nilai Speed monster.

* Speed: unsigned int

Speed. Nilai kecepatan suatu monster. Semakin besar, maka moster tersebut semakin gesit. Nilai speed akan membesar sesuai dengan naiknya level.

* Skill: array of int[MaxMove]

Skill yang dimiliki oleh monster. Monster maksimal memiliki 4 skill.

* PPOfSkills: array of int[MaxMove]

Nilai PP dari skill-skill yang ada di monster. Nilai PP ini bisa di-restore ke nilai maksimum.

* StatusEffect: int

Efek yang sedang dialamai oleh monster pada saat itu.

* NameOfStatusEffect[11]: char

Variabel yang akan menampung nama status effect.

* ChangeName(:char[]): void

Metode yang akan mengubah nama monster. Seperti dijelaskan sebelumnya, nama monster bisa ditentukan oleh pemain.

* IncAge(:int): void

Metode yang akan menambah usia monster.

* AddExp(:unsigned int): void

Add experience. Metode yang digunakan untuk menambah nilai experience pada monster.

* IncLevel(): void

Increase level. Meningkatkan level suatu monster apabila sudah mencapai nilai tertentu.

* RestoreHP(): void

Metode yang akan mengembalikan nilai HP monster ke nilai maksimal.

* ChangeHP(:int): void

Metode yang akan mengubah nilai HP dengan suatu nilai int.

* IncState(:int,:int): void

Increase state. Metode yang akan menambahkan nilai pada PE (HP, Attack, Defense, Speed) monster.

* RestorePP(:int,:ListOfSkill\*): void

Metode yang akan mengembalikan nilai suatu PP monster ke nilai maksimal.

* RestoreAllPP(ListOfSkill\*): void

Metode yang akan mengembalikan semua PP monster ke nilai maksimal.

* DecreasePP(:int): void

Metode yang akan digunakan untuk mengurangi nilai PP.

* ChangeEffect(int): void

Metode yang akan digunakan untuk mengubah efek monster.

* getMonsterNumber(): unsigned int

Metode yang akan mengembalikan nomor monster.

* getName(): array of char

Metode yang akan mengembalikan informasi nama monster.

* getCurAge(): unsigned int

Metode yang akan mengembalikan umur monster saat itu.

* getCurLevel(): unsigned int

Metode yang akan mengembalikan nilai level monster saat itu.

* getCurExp(): unsigned int

Metode yang akan mengembalikan nilai Exp monster saat itu.

* getExpMax(): unsigned int

Metode yang akan mengembalikan nilai Exp maksimal.

* getTotalExp(): unsigned int

Metode yang akan mengembalikan nilai total Exp monster.

* getIncHP(): unsigned int

Metode yang akan mengembalikan nilai IncHP monster.

* getHP(): unsigned int

Metode yang akan mengembalikan nilai HP mosnter.

* getIncAttack(): unsigned int

Metode yang akan mengembalikan nilai IncAttack monster.

* getAttack(): unsigned int

Metode yang akan mengembalikan nilai Attack monster.

* getIncDefense(): unsigned int

Metode yang akan mengembalikan nilai IncDefence monster.

* getDefense(): unsigned int

Metode yang akan mengembalikan nilai Defense monster.

* getIncSpeed(): unsigned int

Metode yang akan mengembalikan nilai IncSpeed monster.

* getSpeed(): unsigned int

Metode yang akan mengembalikan nilai Speed monster.

* getSkills(): array of int

Metode yang akan mengembalikan semua nilai skill monster.

* getSkill(int): int

Metode yang akan mengembalikan nilai skill ke-int suatu monster.

* getPPOfSkills(): array of int

Metode yang akan mengembalikan semua nilai PP monster.

* getPPOfSkill(int): int

Metode yang akan mengembalikan nilai suatu PP monster.

* getStatusEffect(): int

Metode yang akan mengembalikan nilai status effect yang dimiliki monster.

* getStatusEffectName(): array of char

Metode yang akan mengembalikan nama dari status effect yang dimiliki monster.

* **Item**

Kelas Item menjelaskan tentang item yang digunakan untuk membantu pemain dalam pertarungan monster, misalnya menaikkan Max Health Point, menggunakan Ball untuk menangkap monster , dan lain-lain

* Atribut
* const unsigned char MaxNameLength

Menunjukkan panjang maksimal dari nama sebuah item yaitu sebanyak 12 karakter

* unsigned int ItemNumber

Menunjukkan nomor atau indeks yang nantinya digunakan jika pemain akan mengakses sebuah item

* char \*Name

Merupakan nama dari sebuah item dengan panjang maksimal 12 karakter

* int Tipe

Merupakan atribut yang menunjukkan nomer dari suatu tipe item yaitu Potion, Ball, StatIncrease, MonsterEgg.

* Potion digunakan untuk menambah Health Point atau Mana Point pada monster.
* Ball digunakan untuk menangkap monster dalam area luar
* StatIncrease digunakan untuk menaikkan status dari sebuah monster. Yang hanya bisa dinaikkan dengan tipe ini yaitu Max HP, Attack, Defense, Speed, dan Level.
* MonsterEgg digunakan untuk membeli telur monster.
* unsigned int Price

Menunjukkan harga dari item yang dimiliki oleh pemain. Harga tersebut digunakan saat mau menjual item tersebut

* unsigned int ItemValue

Menunjukkan nilai dari tipe sebuah item yang akan digunakan oleh pemain.

* char \*Desc

Berisi deskripsi tentang suatu item

* Method
* unsigned int GetItemNumber()

Mengembalikan nilai indek atau nomor sebuah item

* char\* GetName()

Mengembalikan nama dari item

* int GetTipe()

Mengembalikan nomor tipe suatu item

* unsigned int GetPrice()

Mengembalikan nilai harga suatu item

* unsigned int GetItemValue()

Mengembalikan nilai item saat digunakan

* char\* GetDesc()

Mengembalikan deskripsi sebuah item

* **ListOfItem**

Kelas ini digunakan untuk menampung daftar item ke dalam suatu list dan menampilkan list atau daftar item yang dimiliki.

* + - Atribut
* list<Item> tblItem

berisi list dari semua item yang dimiliki pada PokeRanch

* int nItem

menunjukkan jumlah item yang dimiliki

* Method
* void ReadFromFile()

membaca list item dari sebuah file eksternal dan memasukkan ke dalam atribut tblItem

* int GetNItem()

mengembalikan nilai dari jumlah item yang dimiliki (atribut nItem)

* int IsOnTable(char \*)

Mengecek apakah sebuah item yang akan dibeli terdapat di toko item atau tidak. Jika iya maka method ini akan mengembalikan indeks nomor dari item yang akan dibeli. Jika tidak, maka method ini mengembalikan nilai 0

* void PrintName(int)

menampilkan nama sebuah item

* void PrintPrice(int)

menampilkan harga sebuah item

* int GetPrice(int)

mengembalikan nilai harga suatu item

* void PrintDescription(int)

menampilkan deskripsi dari suatu item

# Program Utama

Program akan memulai *game* di rumah, dimana ada 3 pilihan menu, yaitu New, Load dan Exit. Menu New digunakan untuk membuat player baru, sedangkan menu Load digunakan untuk memakai data pemain yang sudah tersimpan pada game, dan menu Exit digunakan untuk keluar dari game, tetapi data pemain tidak akan disimpan. Menu Exit dapat ditemui di semua area. Selain Exit, fungsi Help juga dapat ditemui di semua area. Help berfungsi untuk menampilkan list fungsi yang dapat digunakan pada area dimana fungsi Help digunakan.

Setelah memilih pemain, pemain dapat memilih untuk keluar dari rumah dengan memilih menu Out. Pemain juga bisa ”menyembuhkan” monster dengan memakai menu Sleep, dimana dalam menu ini HP semua monster akan dikembalikan seperti semula. Untuk menyimpan data, pemain dapat memakai menu Save.

Di luar rumah, ada 3 tempat yang bisa dituju, yaitu Shop, Stadium, Kombinatorium, Home, Gate. Untuk dapat pergi ke tempat tersebut, pemain dapat menggunakan menu Teleport.

Shop merupakan tempat untuk para pemain membeli item yang digunakan dengan fungsi Buy, di Shop para pemain juga bisa menjual barang yang mereka punya dengan fungsi Sell.

Stadium digunakan untuk bertarung melawan komputer, dimana komputer berfungsi seperti pemain yang memiliki monster dan dikendalikan program. Untuk dapat bertarung di stadium, pemain harus mempertaruhkan uang sejumlah minimal 50,000 zeny dengan menggunakan fungsi Battle <bet>. Apabila pemain menang, maka pemain akan mendapatkan uang sejumlah yang dipertaruhkan. Sebaliknya, apabila pemain kalah, maka pemain akan kehilangan uang sejumlah yang dipertaruhkan. Pertarungan di dalam stadium tidak akan mempengaruhi exp monster.

Kombinatorium digunakan untuk menggabungkan dua monster yang dapat digabungkan. Untuk dapat digabungkan, monster tersebut harus berasal dari satu spesies, dan memenuhi syarat level minimum penggabungan. Untuk keluar dari city, pemain dapat memilih untuk Teleport ke Gate, dimana pemain dapat memasuki OuterArea dengan menggunakan fungsi Entry.

Pada OuterArea, terdapat 7 level lantai. Di setiap lantainya memiliki level monster yang berbeda-beda. Untuk lantai level 1, level monster berkisar di antara 1-10, sedangkan di lantai level 7 pemain dapat menemui monster level 60. Kemunculan monster dan bentuk OuterArea dibuat secara acak setiap kali pemain memasuki OuterArea. Jika pemain menemui monster (ditandai dengan huruf M pada OuterArea), maka pemain aka memasuki mode Battle.

Untuk Battle, pemain dapat memilih untuk bertarung, menggunakan item, kabur, atau menangkap monster tersebut, apabila Battle dilakukan di Outer Area. Layar akan menunjukkan status monster kita dan status monster lawan. Monster pertama yang dapat menyerang/menggunakan skill ditentukan dari atribut Speed dari Monster. Skill dapat mempengaruhi perilaku monster pada giliran berikutnya. Sebagai contoh, apabila monster terkena efek Paralyze, maka akan ada kemungkinan bahwa monster tidak bisa menyerang monster lawan pada giliran berikutnya. Pemain juga dapat menggunakan item untuk melawan monster. Apabila monster yang dimiliki pemain menang, maka monster akan memperoleh sejumlah exp dan pemain akan mendapat sejumlah uang. Jika semua monster yang dimiliki pemain kalah, maka pemain akan dikembalikan ke rumah.

# Data Test

Pada bagian item (kelas Item, kelas ListOfItem)

Fitur yang dites pada kelas ini antara lain membaca dari file eksternal, memasukkan ke dalam list dan menampilkan sesuai isi dari file eksternal yang dibaca.

skenario pengetesannya dilakukan dengan memanggil operasi pembacaan file eksternal sekaligus fungsi memasukkan isi dari file eksternal ke dalam list. Kemudian dilakukan pengecekan dengan menampilkan isi dari list tersebut.

Pada bagian monster (kelas Monster, kelas ListOfMonster, dan kelas PlayerMonster)

Fitur yang dites adalah memasukkan objek monster ke list, membaca data monster dari file eksternal.

Skenario yang digunakan adalah menampilkan hasil pembacaan ke layar dan mencetak monster yang telah dimasukkan ke list

Pada bagian skill (kelas Skll, kelas ListOfSkill)

Fitur yang dites adalah pembacaan data skill dari file eksternal dan mencetak hasil pembacaan (mencetak skill). Skenario yang digunakan adalah mencetak hasil pembacaan data skill ke layar.

Pada bagian area, fitur yang dites adalah mencetak map ke layar, berpindah dari suatu tempat ke tempat lain, memindahkan objek ke suatu tempat, dan melakukan teleport. Skenario yang digunakan adalah membuat pseudo-map dan melakukan pergerakan objek di pseudo-map tersebut.

Pada bagian pokeranch, fitur yang dites pada bagian ini mencakup semua kelas yang ada, mulai dari item, skill, monster, dan area. Skenario yang digunakan adalah dengan melakukan simulasi PokeRanch, mulai dari menangkap monster, membeli monster, melakukan battle dsb.

# Pembagian Kerja dalam Kelompok

13511025 Lubis Sucipto : Membuat kelas ListOfSkills dengan turunannya

13511037 Muhammad Rizky W : Membuat kelas ListOfPlayer dengan turunannya

13511051 Andreas Dwi Nugroho : Membuat kelas ListOfItem dengan turunannya

13511071 Setyo Legowo : Membuat kelas PokeRanch dengan turunannya.

13511083 Fawwaz Muhammad : Membuat kelas Area dengan turunannya

13511093 Asep Saepudin : Membuat kelas ListOfMonster dengan turuannya.

# Lampiran

## Notulen Rapat

Asistensi 1:

* Sebaiknya membuat class independen terlebih dahulu
* Field yang mau dibuat bebas
* Teleport digambarkan untuk pindah screen
* Menggunakan string.h untuk penggunaan command

Asistensi 2:

* Bikin main program dahulu agar pembagian kerja dan integrasi bisa terlaksana
* *Framework*
* Main yang bisa dijalanin

Asistensi 3:

* Segera selesaikan laporan log dan pembagian kerja
* Demo class area
* Demo class listofitem
* Demo class listofskill
* Demo class listofmonster
* Demo class listofplayer

Rapat 1 (Minggu, 10 Maret 2013):

* Pembagian tugas

Rapat 2 (Sabtu, 16 Maret 2013):

* *Review* perkembangan pengerjaan tugas

Rapat 3 (Sabtu, 23 Maret 2013):

* Pengerjaan bersama, agar komunikasi bisa lebih baik
* Memberikan koreksi pada tugas anggota lain agar program bisa terintegrasi dengan baik

Rapat 4 (Minggu, 24 Maret 2013):

* Pengerjaan bersama

Rapat 5 (Senin, 25 Maret 2013):

* Penyelesaian semua bagian tugas bersama

## Log Activity Anggota Kelompok

|  |  |  |
| --- | --- | --- |
| Nama | Tanggal | Kegiatan |
| Lubis Sucipto | 8 Maret 2013  11 - 15 Maret 2013  18 - 22 Maret 2013  22 - 25 Maret 2013 | Membuat deklarasi kelas Skill dan ListOfSkill  Membuat implementasi kelas Skill dan ListOfSkill.  Mengimplementasikan semua kelas Skill dan ListOfSkill dengan Kelas-kelas lain  Membantu penyelesaian program utama dan perapihan tampilan |
| Muhammad Rizky W. | 10 Maret 2013  13 Maret 2013  15 Maret 2013  17 Maret 2013  19 Maret 2013  21 Maret 2013  23 Maret 2013  24 Maret 2013  25 Maret 2013 | Membuat definisi dari class Player dan fungsi atribut dan method class Player  Membuat realisasi dari class Player  Membuat constructor class Player  Membuat realisasi dari class ListOfPlayer  Meneruskan realisasi class Player, memperbaiki constructor  Menambahkan beberapa prosedur pada class Player  Meneruskan realisasi class Player, memperbaiki beberapa prosedur  Membuat void RestoreAllMonster, bool IsItemAda pada class Player  Membuat laporan |
| Andreas Dwi N. | 14 – 15 Maret 2013  16 - 18 Maret 2013  20 Maret 2013  22 Maret 2013 | Membuat deklarasi dan implementasi kelas Item  Membuat deklarasi dan implementasi kelas List Of Item  Memperbaiki kelas List Of Item dengan memakai STL List  Membuat kelas List Item Of Player (deklarasi dan implementasi) |
| Setyo Legowo | 7 Maret 2013  9 Maret 2013  10 Maret 2013  11 Maret 2013  12 Maret 2013  13 Maret 2013  15 Maret 2013  16 Maret 2013  18 Maret 2013  20 Maret 2013  21 Maret 2013  22 Maret 2013  23 Maret 2013  24 Maret 2013  25 Maret 2013 | Membuat class diagram  Membuat graph, mengerjakan main program  Membuat class PokeRanch  Meneruskan pembuatan class PokeRanch dan graph  Meneruskan pembuatan main program  Membuat class mynamespace  Meneruskan pembuatan PokeRanch  Menyatukan program  Memperbaiki class lain  Memperbaiki main program  Memperbaiki class PokeRanch  Memperbaiki main program  Membuat konsep mode Battle, mencari rumus-rumus untuk mode Battle  Mengerjakan mode Battle  Meneruskan pembuatan mode Battle |
| Fawwaz Muhammad | 10-25 Maret 2013  12-25 Maret 2013  14-25 Maret 2013 | Membuat kelas Area  Membuat kelas City  Membuat kelas Kombinatorium |
| Asep Saipudin | 14-15 Maret 2013  16 Maret 2013  17 & 22 Maret 2013  23-24 Maret 2013 | Membuat kelas Monster  Membuat kelas ListOfMonster  Membuat kelas PlayerMonster  Melengkapi kelas Monster, kelas ListOfMonster, dan PlayerMonster dengan atribut-atribut dan metode-metode yang belum ada agar bisa diintegrasikan dengan kelas-kelas lain. |