LAPORAN TUGAS BESAR II

IF2032 Pemrograman 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-TB2-14-01* | *56/56* |
| *10 Mei 2013* |

Daftar Isi

[1 Ringkasan 3](#_Toc355006603)

[2 Gameplay](#_Toc355006604) 4

[3 Penjelasan Tambahan Spesifikasi Tugas 5](#_Toc355006605)

[3.1 Element](#_Toc355006606) 10

[3.2 Status Efect](#_Toc355006607) 10

[3.3 Combinatorium](#_Toc355006607) 10

[4 Diagram Kelas](#_Toc355006608) 11

[5 Penjelasan Kelas](#_Toc355006609) 12

[6 Program Utama](#_Toc355006610) 53

[7 Algoritma-Algoritma Menarik](#_Toc355006611) 54

[7.1 Pergerakan Monster 12](#_Toc355006612)

[8 Pembagian Kerja dalam Kelompok](#_Toc355006617) 55

[9 Lampiran](#_Toc355006618) 55

[9.1 Notulen Rapat](#_Toc355006619) 55

[9.2 Log Activity Anggota Kelompok](#_Toc355006620) 55

# 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 data spesies monster yang ada dalam permainan, cara bertarung, dan menangkap monster liar menggunakan *monster ball*. 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, mengabungkan monster dengan monster lain supaya lebih kuat, dan berkeliling di area luar. Perbandingan *gameplay* antara tugas besar 1 dengan tugas besar 2 adalah pada tugas besar dua permainan ini dapat dimainkan menggunakan OS Android. Selain itu ada perkembangan penggunaan *Item* seperti TM (*Technical Machine*), repel, dan penggunaan alat-alat untuk karakter (sepeda, pancingan, dsb). Lalu perkembangan dalam skill adalah skill dapat berevolusi pada level tertentu. Lalu perkembangan dalam Monster tidak signifikan, hanya perilaku di dunia luar yang berbeda yakni monster dapat bergerak secara acak, mendekat ke karakter, dan atau menjauhi karakter. Perkembangan besar pada area luar adalah ditambahnya area perairan, karakter dapat bergerak di dalam bangunan, dan munculnya monster liar tergantung waktu siang atau malam. Dari dua permainan tersebut memiliki satu kemiripan yaitu bertarung dengan monster liar di area luar. Lalu di area lumat terdapat batu (di permainan ini disebut boulder) yang dapat digeser. Lalu ada pohon kecil yang menghalangi jalan yang dapat ditebang agar jalan terbuka. Lalu di permainan ini pemain dapat berkeliling di area perairan dengan syarat memiliki monster berjenis air dan memiliki skill Swim. Di area luar ini juga ada siang malam, saat malam player hanya bisa melihat jalan disekitarnya saja, jika ingin menambah jarak penglihatan harus menggunakan item torch. Monster-monster disini juga tidak hanya diam, tapi ada yang bergerak random, mengejar player dan ada yang menjauhi player. Player dapat melihat pokedex yang berisi semua monster yang ada di pokeranch. Pergerakan pemain menggunakan animasi, dan dialog interaktif dengan npc di shop misalnya.

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).

Tugas Besar II ini bertujuan agar bisa menggunakan kelas dalam java dengan banyak kelebihan java dari bahasa C++ . Seperti kemampuan java swing, thread, dan activity yang membuat java lebih bagus digunakan dalam OOP. Saran untuk tugas ini seharusnya untuk database monster,skill,ataupun item ada di spek sehingga lebih mempermudah membuat database.

# Gameplay

Saat permainan PokeRanch dijalankan, pengguna dihadapkan pada main menu yang terdapat tombol-tombol dibawah ini:



|  |
| --- |
| Main Baru  Tombol ini digunakan untuk menjadikan pengguna sebagai pemain baru dari permainan PokeRanch. Jika tombol ini ditekan maka muncul Dialog Box untuk memasukkan nama pemain. Klik Sign Up untuk daftar dan cancel untuk membatalkan menjadi pemain baru. |

|  |
| --- |
| Lanjutkan Permainan  Tombol ini digunakan untuk mengambil pengguna yang telah terdaftar sebelumnya sebagai pemain aktif. Saat tombol ini ditekan akan muncul Dialog Box untuk memilih nama pemain yang telah terdaftar sebelumnya. Setelah nama pemain dipilih maka simpanan terakhir permainan akan di load.  Bantuan  Tombol ini digunakan untuk menampilakan bantuan bermain Pokeranch, sehingga untuk pemain yang bingung cara bermain dapat menekan tombol ini |

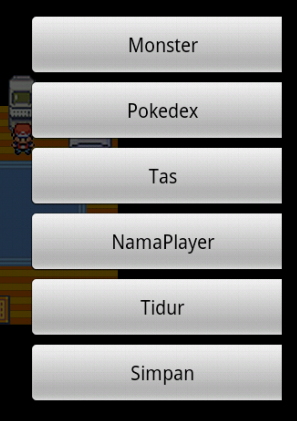
Ada pengenalan khusus untuk pemain baru atau yang memberikan perintah new pada program ini, yaitu pemain baru akan diberikan semacam *user agreement* yang menceritakan bahwa pemain baru adalah seorang pendatang ke dunia Poke. Pendatang baru tersebut dapat berteman dengan Poke (monster) dan monster tersebut 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 menekan tombol OK atau Back maka pemain akan tiba-tiba berada di rumah.

Jika pemain lama, pengantar permainan tidak akan ditampilkan.

Setelah itu pemain (baru maupun lama) akan berada di dalam rumah.



Menu-menu yang dapat dipilih di rumah:



Monster

Tombol ini digunakan untuk menampilkan monster yang dimiliki player

Pokedex

Tombol ini digunakan untuk menampilkan semua monster yang ada di Pokeranch

Tas

Tombol ini digunakan untuk menampilkan semua item yang dimiliki player

<Nama Player>

Tombol ini digunakan untuk menampilkan status dan info player yang sedang bermain

Tidur

Tombol ini digunakan untuk membuat player tidur sehingga semua monster kembali HP dan PP nya kedalam keadaan maksimal

Simpan

Tombol ini digunakan untuk menyimpang permaian player

Setelah keluar dari rumah player, player dapat bergerak keliling kota.



Menu di kota sama dengan di rumah tapi tidak ada Simpan dan Tidur

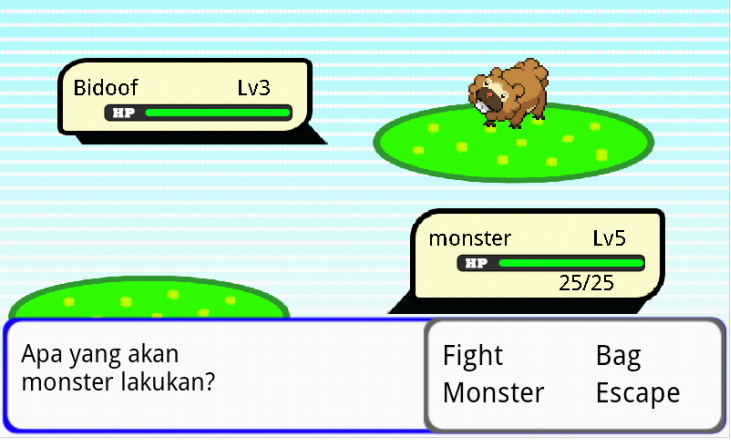
Player dapat membeli item di toko(Shop), bertanding monster dengan bet(Stadium), mengombinasikan monster(Combinatorium), serta menjelajahi dunia luar OuterArea dan Pantai.

Untuk bisa ke Stadium, Combinatorium, dan dunia luar player harus memiliki monster. Player juga dapat menggunakan sepeda agar lebih cepat.

Di OuterArea ada 10 level lantai, keluar dari kota memasuki level 1, jika berjalan ke ujung akan berindah ke level 2, dst. Disini ada monster-monster yang dapat dilawan dan ditangkap oleh player. Pergerakan monster-monster disini ada yang random, ada yang mengejar player, dan ada yang menjauhi player. Player dapat mendorong batu dan memotong pohon dengan skill push dan cut.



Jika player berpapasan dengan monster akan dilakukan battle dan player diberi 4 pilihan:

Fight

Tombol ini digunakan untuk melawan monster tersebut

Monster

Tombol ini digunakan untuk mengganti monster yang akan bettle

Bag

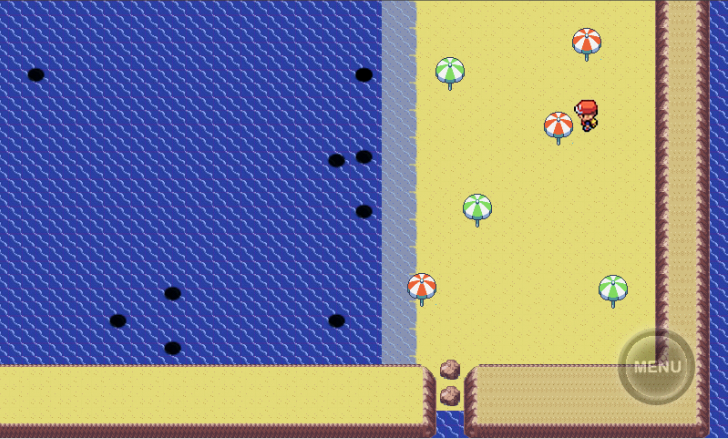
Tombol ini digunakan untuk menampilkan isi tas dan memakai item

Escape

Tombol ini digunakan untuk kabur dari battle

Untuk battle di stadium hampir sama dengan battle di OuterArea, namun bedanya di Stadium player tidak bisa menggunakan item, ataupun escape.

Jika player di pantai player dapat menggunakan monster untuk berenang dan melawan monster jika bertemu dengan monster yang bergerak seperti di OuterArea:



# Penjelasan Tambahan Spesifikasi Tugas

## Element

Pada game ini terdapat 10 elemen dasar.

|  |
| --- |
| **Nama Element** |
| Psychic |
| Fight |
| Normal |
| Water |
| Grass |
| Ground |
| Flying |
| Ice |
| Electric |
| Dragon |

## Status Effect

Pada game ini terdapat 6 status effect

|  |
| --- |
| **Nama StatusEffect** |
| Paralyz |
| Poison |
| Bad Poison |
| Burn |
| Frozen |
| Sleeep |

## Kombinatorium

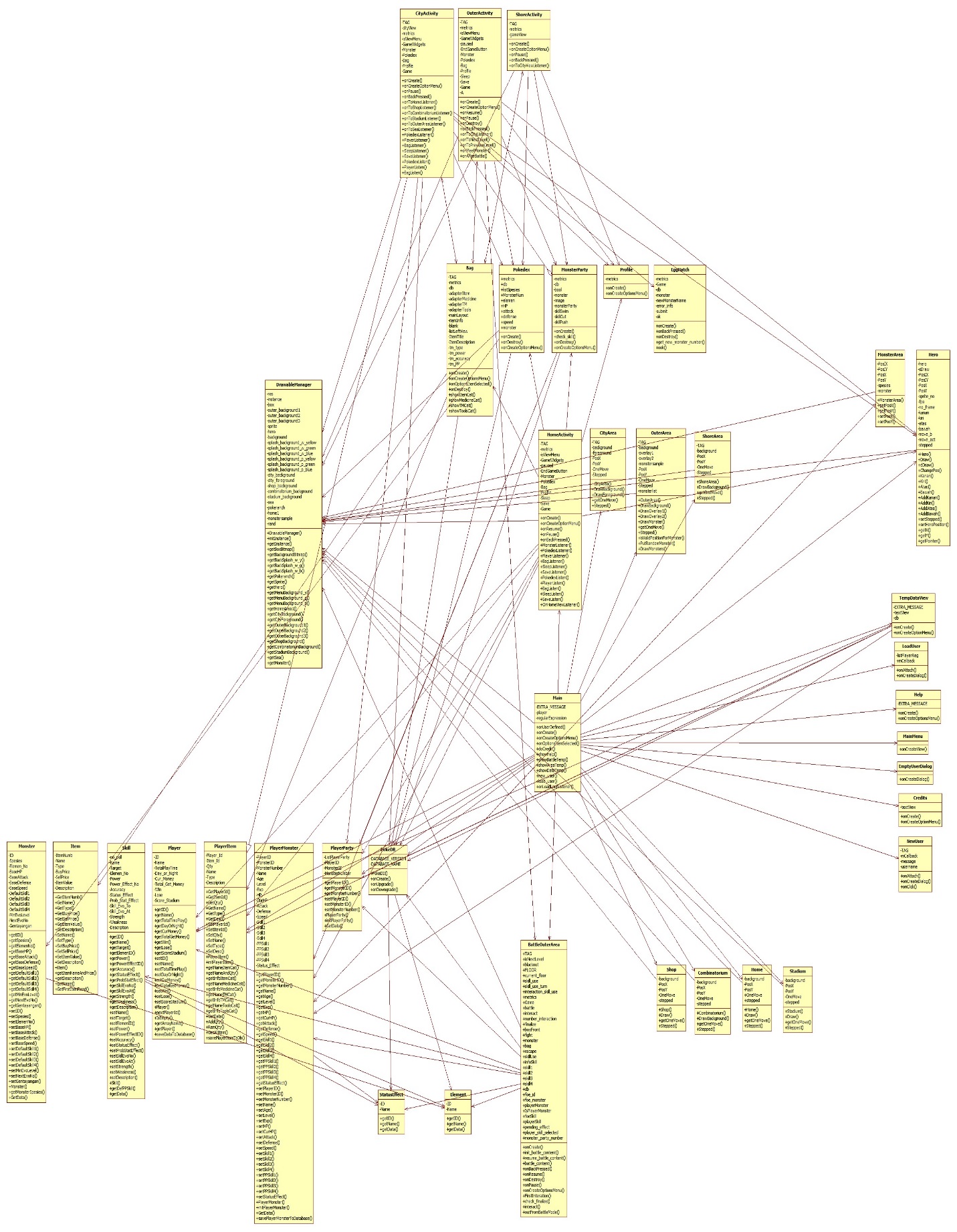
Dua monster dapat digabungkan jika mereka berasal dari spesies yang sama. Atribut dari monster baru yang digabungkan didapat dari 2 monster yang dikombinasikan. Sebagai contoh, level monster baru didapat dari rumus berikut:

Monster1 = m1

Monster2 = m2

|  |  |
| --- | --- |
| **Kondisi** | **Level Monster Baru** |
| If (50<(Level m1+level m2)<100) | (Level m1 + level m2)\*3/2 |
| If ((Level m1 + level m2) > 100) | (Level m1 + level m2)\*5/3 |
| If ((Level m1 + level m2) < 50) | Level m1 + level m2 |

# Diagram Kelas



# Penjelasan Kelas

* Class Beginning

Activity untuk awal permainan

* + - * + Atribut
* NEW\_USER\_MESSAGE: konstanta new user
* db: database
* buttonAgree: button
* textRun: TextView
* messae\_index: indeks pesan
* message\_prof: ArrayList berisi string
  + - * + Method
* onCreate(Bundle): void
* onCreateOptionMenu(Menu): boolean
* onDestroyt (): menghancurkan db
* MakeNewPlayer(String): membuat player baru
* onBackPressed(): kembali
* toHome(View): ke rumah
* toHome(): kembali ke rumah
* Class Main

Kelas untuk berisi menu-menu di awal permainan

* + - * + Atribut
* EXTRA\_MESSAGE: konstanta pokeranch
* player: Player
* metrics: menampilkan matriks
* OneMove: integer
* screeenWidthPixels: besar pixel lebar
* screenHeightPixels: besar pixel tinggi
* TAG: String
  + - * + Method
* onUserDefined(String): void
* onCreate(Bunddle): void
* onCreateOptionsMenu(Menu): boolean
* onOptionsItemSelected(MenuItem): boolean
* doCredit(): menampilkan kredit
* showHelp(View): menampilkan help
* showBattleTemp(View): menampilkan Battle temporari
* showAreaTemp(View): menampilkan Area temporari
* showDataTemp(View): menampilkan Data temporari
* new\_user(View): menambilkan user baru
* load\_user(View): meload user lama
* onLoadUserListener(String): void
* Class DrawableManager

Kelas mengatur gambar

* + - * + Atribut
* res: Resaources
* instance: DrawableManager
* box: NinePatchDrawable
* blackBox: NinePatchDrawble
* blackBox2: NinePatchDrawable
* blueBox: NinePatchDrawable
* outer\_background1: bitmap
* outer\_background2: bitmap
* outer\_background3: bitmap
* hero: bitmap
* hero\_bike: bitmap
* hero\_swim: bitmap
* city\_background: bitmap
* city\_foreground: bitmap
* shop\_background: bitmap
* combinatorium\_background: bitmap
* stadium\_background: bitmap
* sea: bitmap
* pokeranch: bitmap
* home1: bitmap
* monsertsample: bitmap
* battle\_background: bitmap
* battle\_grass: bitmap
* foe\_properties: bitmap
* self\_properties: bitmap
* health\_bar: bitmap
* frz: freeze bitmap
* brn: burn bitmap
* slp: sleep bitmap
* plz: paralyzed bitmap
* psn: poison bitmap
* grass: grass bitmap
* sea\_animate: bitmap
* monter\_load: list integer monter id yg di keluarkan
* monter1: bitmap
* mosnter2: bitmap
* monster3: bitmap
* mosnter4: bitmap
* menu\_button: bitmap
* batu: bitmap
* last\_index\_monster: index monster terakhir
* rand: randomize
  + - * + Method
* DrawaleManager(Context): konstruktor
* initInstance(Context): inisisasi instance
* getInstance(): mengembalikan instance
* initAllBitmap(): inisisasi Bitmap
* getBoxBitmap(): mengembalikan box bitmap
* getBlackBoxBitmap(): mengembalikan black box bitmap
* getBlackBox2Bitmap(): mengembalikan black box 2 bitmap
* getBlueBoxBitmap(): mengembalikan blue box bitmap
* getPokeranch(): mengembalikan pokeranch bitmap
* getHero(): mengembalikan hero bitmap
* getHeroSwim(): mengembalikan hero\_swim bitmap
* getHeroBike(): mengembalikan hero\_bike bitmap
* getHomeArea1(): mengembalikan home1 bitmap
* getCityBackground(): mengembalikan city\_background bitmap
* getCityForegorund(): mengembalikan city\_foreground bitmap
* getOuterBackground1(): mengembalikan outer\_background1 bitmap
* getOuterBackground2(): mengembalikan outer\_background2 bitmap
* getOuterBackground3(): mengembalikan outer\_background3 bitmap
* getShopBackground(): mengembalikan shop\_background bitmap
* getCombinatoriumBackground(): mengembalikan combinatorium\_background bitmap
* getStadiumbackground(): mengembalikan stadium\_background bitmap
* getSea(): mengembalikan sea bitmap
* BattleBack(): mengembalikan battle\_background bitmap
* BattleGrass(): mengembalikan battle\_grass bitmap
* getMonster(): mengembalikan monstersample bitmap
* foe\_properties(): mengembalikan foe\_properties bitmap
* self\_poperties(): mengembalikan self\_properties bitmap
* health\_bar(): mengembalikan health\_bar bitmap
* Frozen(): mengembalikan frz bitmap
* Burn(): mengembalikan brn bitmap
* Sleep(): mengmebalikan slp bimap
* Paralyzed(): mengembalikan plz bitmap
* Poison(): megembalikan psn bitmap
* Grass(): mengembalikan grass bitmap
* SeaAnimation(): mengembalikan sea\_animate bitmap
* MonsterPicture(int): mengembalikan monster\_picture bitmap sesuai id monster
* Menu\_button(): mengembaikan menu\_button bitmap
* get\_batu(): mengembalikan batu bitmap
* CityActivity

Class CityActivity digunakan untuk menampilkan City

* Atribut
* cityView: tampilan untuk kota
* TAG: nama kota
* Metrics:
* GameWidgets: Layout dari kota
* Monster: Button untuk Monster
* Pokedex: Button untuk Pokedex
* bagButton: Button untuk Profile
* boolean isViewMenu;
* ImageView menu\_button
* Method
* onCreate(Bundle): membuat tampilan City
* onCreateOptionsMenu(Menu): membuat tampilan Options
* onResume(): menampilkan keadaan saat permainan di-resume
* onePause(): menampilkan keadaan saat permainan di-pause
* onBackPressed()
* onToHomeListener(int): memeriksa apakah pemain ke Home
* onToShopListener(int): memeriksa apakah pemain ke Shop
* onToCombinatoriumListener(int): memeriksa apakah pemain ke Combinatorium
* onToStadiumListener(int): memeriksa apakah pemain ke Stadium
* onToOuterAreaListener(int): memeriksa apakah pemain ke OuterArea
* onToSeaListener(int): memeriksa apakah pemain ke Sea
* MonsterListener: Menjalankan menu MonsterParty
* PokedexListener: Menjalankan menu Pokedex
* PlayerListener: Menjalankan menu Profile
* BagListener: Menjalankan menu Bag
* MonsterPartys(): Menampilkan menu MonsterParty
* PokedexListen(): Menampilkan menu Pokedex
* PlayerListen(): Menampilkan menu Profile
* BagListen(): Menampilkan menu Bag
* CityArea

Class CityArea menampilkan grafis dari City

* Atribut
* Int scaling
* Int TAG
* Bitmap background
* Bitmap foreground
* Paint grass
* PosX: koordinat X dari Player
* PosY: koordinat Y dari Player
* Int xOffset
* Int yOffset
* stepped: menampilkan area yang bisa dilalui oleh Player
* Method
* initCityArea: inisialisasi City
* CityArea(int, int): constructor City dengan parameter
* CityArea(): constructor City
* DrawBackground(Canvas): menampilkan grafis background City
* DrawForeground(Canvas): menampilkan grafis foreground City
* Stepped(): Menampilkan area yang bisa dilalui Player
* forceChangeOffset(int, int): mengubah nilai xOffset dan yOffset
* getxOffset(): getter xOffset
* getyOffset(): getter yOffset
* ChangeOffset(int, int, int, int): mengubah nilai xOffset dan yOffset
* CityView

Class CityView digunakan untuk rendering City

* Atribut
* Float density
* screenWidth: menyimpan lebar layar
* screenHeight: menyimpan tinggi layar
* CityViewListener mCityViewListener
* String TAG
* Thread GameLoopThread
* Matrix matrix
* Pause\_event: menandai apakah permainan sedang dalam keadaan pause
* Hero: Pemain pada City
* City: menu City
* Int CallAct
* Method
* CityView(Context): constructor dengan parameter
* screenSize(int, int): setter untuk screenWidth dan screenHeight
* SetActivityFrom(String): menentukan posisi Hero berdasarkan activity sebelumnya
* surfaceChanged(SurfaceHolder, int, int, int)
* surfaceCreated(SurfaceHolder)
* surfaceDestroyed(SurfaceHolder)
* initThread(): inisialisasi CityView
* releaseThread()
* update(): meng-update map City
* render(Canvas): render gambar City
* onTouchEvent(MotionEvent): membuat event berdasarkan input Touch yang diterima
* setPauseEvent(Boolean): mengeset event Pause
* isToChangeView(): mengubah tampilan (City/ Home/ OuterArea/ Shop/ Pantai/ Kombinatorium)
* resetCallAct(): me-reset nilai CallAct menjadi 1
* CombinatoriumActivity

Class CombinatoriumActivity digunakan untuk menampilkan aktivitas pada Kombinatorium

* Atribut
* String TAG
* DisplayMetrics metrics
* RelativeLayout Game
* CombinatoriumView mGameView
* Button Monster
* Button Pokedex
* Button Bag
* Button Profile
* TextView interact
* LinearLayout boxButton
* Boolean isInteraction
* Int i\_interact
* LinearLayout GameWidgets
* Boolean isViewMenu
* ImageView menu\_button
* Method
* onCreate(Bundle): membuat tampilan Kombinatorium
* onBackPressed()
* onCreateOptionsMenu(Menu): membuat tampilan menu options
* onPause(): menampilkan keadaan saat keadaan pause terjadi
* onToCityListener: memeriksa apakah pemain ke City
* MonsterListener: menjalankan menu MonsterParty
* PokedexListener: menjalankan menu Pokedex
* BagListener: menjalankan menu Bag
* CombineListener: menjalankan menu Kombinatorium
* MonsterPartys(): menampilkan menu MonsterParty
* PokedexListen(): menampilkan menu Pokedex
* PlayerListen(): menampilkan menu Profile
* BagListen(): menampilkan menu Bag
* CombineMonsterListen(): menampilkan menu Kombinatorium
* startInteract(int): menampilkan teks interaksi kepada pemain
* InteractListener: menjalankan teks interaksi sesuai dengan input yang diterima
* CombinatoriumView

Class CombinatoriumView digunakan untuk menampilkan grafis Kombinatorium

* Atribut
* Float density
* screenWidth: menyimpan lebar layar
* screenHeight: menyimpan tinggi layar
* CombinatoriumViewListener mCombinatoriumViewListener
* String TAG
* Thread GameLoopThread
* Matrix matrix
* Pause\_event: menandai apakah permainan sedang dalam keadaan pause
* Hero: Pemain
* Combinatorium: menu Kombinatorium
* Int CallAct
* Method
* CombinatoriumView(Context): constructor dengan parameter
* screenSize(int, int): setter untuk screenWidth dan screenHeight
* SetActivityFrom(String): menentukan posisi Hero berdasarkan activity sebelumnya
* surfaceChanged(SurfaceHolder, int, int, int)
* surfaceCreated(SurfaceHolder)
* surfaceDestroyed(SurfaceHolder)
* initThread(): inisialisasi Combinatorium View
* releaseThread()
* update(): meng-update map Combinatorium
* render(Canvas): render gambar Combinatorium
* onTouchEvent(MotionEvent): membuat event berdasarkan input Touch yang diterima
* setPauseEvent(Boolean): mengeset event Pause
* isOutComb(): memeriksa apakah pemain keluar dari kombinatorium
* isFrontPerson(): memeriksa apakah pemain berada di depan tukang kombinatorium
* resetCallAct(): me-reset nilai CallAct menjadi 1
* GameLoopThread

Class GameLoopThread digunakan untuk rendercanvas pada game

* Atribut
* Boolean running
* SurfaceHolder surfaceHolder
* homeView: tampilan Home
* cityView: tampilan City
* outerView: tampilan OuterArea
* shopView: tampilan Shop
* combinatoriumView: tampilan Kombinatorium
* stadiumView: tampilan Stadium
* shoreView: tampilan Shore
* battleView: tampilan Batle
* MAX\_FPS: fps maksimal yang diinginkan
* MAX\_FRAME\_SKIPS: fps maksimal yang bisa diskip
* Int FRAME\_PERIOD
* Int threadView
* Boolean render
* Method
* GameLoopThread(SurfaceHolder, HomeView): constructor dengan parameter HomeView
* GameLoopThread(SurfaceHolder, ShopView): constructor dengan parameter ShopView
* GameLoopThread(SurfaceHolder, CombinatoriumView): constructor dengan parameter CombinatoriumView
* GameLoopThread(SurfaceHolder, StadiumView): constructor dengan parameter StadiumView
* GameLoopThread(SurfaceHolder, CityView): constructor dengan parameter CityView
* GameLoopThread(SurfaceHolder, OuterView): constructor dengan parameter OuterView
* GameLoopThread(SurfaceHolder, ShoreView): constructor dengan parameter ShoreView
* GameLoopThread(SurfaceHolder, BattleView): constructor dengan parameter BattleView
* setRunning(Boolean): setter running
* run(): menjalankan GameLoopThread
* setRender(Boolean): setter render
* HomeActivity

Class HomeActivity digunakan untuk menampilkan aktivitas pada Home

* Atribut
* String TAG
* DisplayMetrics metrics
* Button EndGameButton
* Button Monster
* Button Pokedex
* Button Bag
* Button Profile
* Button Sleep
* Button Save
* RelativeLayout Game
* HomeView A
* LinearLayout GameWidgets
* Boolean isViewMenu
* ImageView menu\_button
* Method
* onCreate(Bundle): membuat tampilan Home
* onBackPressed()
* onCreateOptionsMenu(Menu): membuat tampilan menu options
* onPause(): menampilkan keadaan saat keadaan pause terjadi
* MonsterListener: menjalankan menu MonsterParty
* PlayerListener: menjalankan menu Profile
* PokedexListener: menjalankan menu Pokedex
* BagListener: menjalankan menu Bag
* MonsterPartys(): menampilkan menu MonsterParty
* PokedexListen(): menampilkan menu Pokedex
* PlayerListen(): menampilkan menu Profile
* BagListen(): menampilkan menu Bag
* SleepListen(): menampilkan menu Sleep
* SaveListen(): menampilkan menu Save
* OnHomeViewListener(int): menjalankan menu Home
* HomeView

Class HomeView digunakan untuk menampilkan grafis Home

* Atribut
* Float density
* HomeViewListener mHomeViewListener
* String TAG
* Thread GameLoopThread
* Matrix matrix
* Pause\_event: menandai apakah permainan sedang dalam keadaan pause
* Hero: Pemain
* Home: menu Home
* Int CallAct
* Method
* HomeView(Context, int. intt): constructor dengan parameter
* SetActivityFrom(String): menentukan posisi Hero berdasarkan activity sebelumnya
* surfaceChanged(SurfaceHolder, int, int, int)
* surfaceCreated(SurfaceHolder)
* surfaceDestroyed(SurfaceHolder)
* initThread(): inisialisasi HomeView
* releaseThread()
* update(): meng-update map Home
* render(Canvas): render gambar Home
* onTouchEvent(MotionEvent): membuat event berdasarkan input Touch yang diterima
* setPauseEvent(Boolean): mengeset event Pause
* isOutHome(): memeriksa apakah pemain keluar dari Home
* resetCallAct(): me-reset nilai CallAct menjadi 1
* OuterActivity

Class OuterActivity digunakan untuk menampilkan OuterArea

* Atribut
* String TAG
* DisplayMetrics metrics
* Button EndGameButton
* Button Monster
* Button Pokedex
* Button Bag
* Button Profile
* Button bagButton
* RelativeLayout Game
* OuterView A
* LinearLayout GameWidgets
* Boolean isViewMenu
* ImageView menu\_button
* Boolean isBounce
* Method
* onCreate(Bundle): membuat tampilan Home
* onBackPressed()
* onCreateOptionsMenu(Menu): membuat tampilan menu options
* onPause(): menampilkan keadaan saat keadaan pause terjadi
* onToCityListener: memeriksa apakah pemain kembali ke City
* onToNextLevel(): menambah level OuterArea
* onToPreviousLevel(): mengurangi level OuterArea
* onMeetMonster(): memeriksa apakah pemain bertemu Monster
* onAfterBattle(): menjalankan menu setelah Battle
* MonsterListener: menjalankan menu MonsterParty
* PokedexListener: menjalankan menu Pokedex
* BagListener: menjalankan menu Bag
* MonsterPartys(): menampilkan menu MonsterParty
* PokedexListen(): menampilkan menu Pokedex
* PlayerListen(): menampilkan menu Profile
* BagListen(): menampilkan menu Bag
* OuterArea

Class OuterArea menampilkan grafis dari Outer

* Atribut
* Int TAG
* Int scalling
* Bitmap background
* Bitmap overlay1
* Bitmap overlay2
* PosX: koordinat X dari Player
* PosY: koordinat Y dari Player
* OneMove: menentukan besarnya perpindahan pemain di Outer
* stepped: menampilkan area yang bisa dilalui oleh Player
* Paint grass
* Int xOffset
* Int yOffset
* Method
* Init(): inisialisasi Outer
* OuterArea(int, int): constructor Outer dengan parameter
* DrawBackground(Canvas): menampilkan grafis background Outer
* DrawOverlay1(Canvas): menampilkan grafis Overlay1 Outer
* DrawOverlay2(Canvas): menampilkan grafis Overlay1 Outer
* Stepped(): Menampilkan area yang bisa dilalui Player
* getOneMove(): getter OneMove
* forceChangeOffset(int, int): mengubah nilai xOffset dan yOffset
* getxOffset(): getter xOffset
* getyOffset(): getter yOffset
* ChangeOffset(int, int, int, int): mengubah nilai xOffset dan yOffset
* OuterView

Class OuterView digunakan untuk menampilkan grafis OuterArea

* Atribut
* Float density
* Int screenWidth
* Int screenHeight
* OuterViewListener mOuterViewListener
* Bitmap Cut\_tree
* String TAG
* GameLoopThread thread
* Matrix matrix
* Pause\_event: menandai apakah permainan sedang dalam keadaan pause
* Hero: Pemain
* OuterArea: menu OuterArea
* Int CallAct
* Boolean KetemuMonster
* floor\_level: level OuterArea
* Paint paint
* Bitmap batuBitmap: gambar batu
* listPosisiBatu: array berisi posisi batu pada map OuterArea
* listPosisiPohon: array berisi posisi pohon pada map OuterArea
* isUsePush: memeriksa apakah pemain menggunakan Push
* Method
* OuterView(Context): constructor dengan parameter
* BuildRockPosition(): menambahkan posisi batu pada listPosisiBatu
* BuildListPohon(): menambahkan posisi pohon pada listPosisiPohon
* screenSize(int, int): setter screenWidth dan screenHeight
* SetActivityFrom(String): menentukan posisi Hero berdasarkan activity sebelumnya
* surfaceChanged(SurfaceHolder, int, int, int)
* surfaceCreated(SurfaceHolder)
* surfaceDestroyed(SurfaceHolder)
* initThread(): inisialisasi HomeView
* releaseThread()
* update(): meng-update map Home
* GambarKetemuMonsterLho(Canvas): menampilkan grafis saat pemain bertemu monster
* DrawAllBatu(Canvas): menampilkan grafis batu pada OuterArea
* UpdateBatuList(int, int, int, Boolean): mengubah koordinat batu apabila pemain menggunakan fungsi Push
* geserBatuList(int, int, int, int, int, int, int): menggeser posisi batu
* isValidGeserBatu(int, int, int, int): memeriksa apakah batu bisa digeser ke arah tertentu
* render(Canvas): render gambar Home
* DrawCutTree(Canvas): menampilkan gambar pohon yang telah di-Cut
* onTouchEvent(MotionEvent): membuat event berdasarkan input Touch yang diterima
* setPauseEvent(Boolean): mengeset event Pause
* isToChangeView(): mengubah gambar saat pemain berada pada koodinat tertentu
* NemuMonsterGaYa(): memeriksa apakah pemain bertemu monster
* isValidPositionForMonster(int, int): memeriksa apakah posisi monster pada OuterArea valid
* RandomizeMonster(): menentukan posisi monster dengan fungsi random
* findInMonsterList(int, int): memeriksa apakah ada monster pada koordinat input
* delMonsterList(int, int): menghapus monster dari list MonsterArea
* drawMonsterArea(canvas): menggambar MonsterArea
* updatemonsterlist(int, int): mengubah list monster
* drawNightMode(canvas): menampilkan grafis saat malam
* resetCallAct(): me-reset nilai CallAct menjadi 1
* ShopActivity

Class ShopActivity digunakan untuk menampilkan aktivitas pada Shop

* Atribut
* String TAG
* DisplayMetrics metrics
* Button EndGameButton
* Button Monster
* Button Pokedex
* Button Bag
* Button Profile
* RelativeLayout Game
* ShopView A
* Boolean isInteraction
* Int i\_interact
* LinearLayout GameWidgets
* Boolean isViewMenu
* ImageView menu\_button
* Method
* onCreate(Bundle): membuat tampilan Shop
* onBackPressed()
* onCreateOptionsMenu(Menu): membuat tampilan menu options
* onPause(): menampilkan keadaan saat keadaan pause terjadi
* MonsterListener: menjalankan menu MonsterParty
* PokedexListener: menjalankan menu Pokedex
* PlayerListener: menjalankan menu Profile
* BagListener: menjalankan menu Bag
* BuyListener: menjalankan menu Buy
* SellListener: menjalankan menu Sell
* CancelListener: menjalankan menu sesudah transaksi
* MonsterPartys(): menampilkan menu MonsterParty
* PokedexListen(): menampilkan menu Pokedex
* PlayerListen(): menampilkan menu Profile
* BagListen(): menampilkan menu Bag
* BuyListen(): menampilkan menu Buy
* SellListen(): menampilkan menu Sell
* startInteract(int): menjalankan interaksi antara penjaga toko dan pemain
* InteractListener: menjalankan menu Interact
* onOutToCity(int): menjalankan menu City (keluar dari Shop)
* ShopView

Class ShopView digunakan untuk menampilkan grafis Shop

* Atribut
* Float density
* Int screenWidth
* Int screenHeight
* ShopViewListener mShopViewListener
* String TAG
* Thread GameLoopThread
* Matrix matrix
* Pause\_event: menandai apakah permainan sedang dalam keadaan pause
* Hero: Pemain
* Shop: menu Shop
* Int CallAct
* Method
* ShopView(Context): constructor dengan parameter
* changeScreen(int, int): setter ScreenWidth dan ScreenHeight
* SetActivityFrom(String): menentukan posisi Hero berdasarkan activity sebelumnya
* surfaceChanged(SurfaceHolder, int, int, int)
* surfaceCreated(SurfaceHolder)
* surfaceDestroyed(SurfaceHolder)
* initThread(): inisialisasi HomeView
* releaseThread()
* update(): meng-update map Home
* render(Canvas): render gambar Home
* onTouchEvent(MotionEvent): membuat event berdasarkan input Touch yang diterima
* setPauseEvent(Boolean): mengeset event Pause
* isToChangeView(): memeriksa apakah pemain keluar dari Shop
* isFrontShop(): memeriksa apakah pemain berada di depan Shop
* resetCallAct(): me-reset nilai CallAct menjadi 1
* ShoreActivity

Class ShoreActivity digunakan untuk menampilkan aktivitas pada Shore

* Atribut
* String TAG
* DisplayMetrics metrics
* Button Monster
* Button Pokedex
* Button Bag
* Button Profile
* RelativeLayout Game
* Boolean isInteraction
* LinearLayout GameWidgets
* Boolean isViewMenu
* ImageView menu\_button
* Method
* onCreate(Bundle): membuat tampilan Shore
* onBackPressed()
* onCreateOptionsMenu(Menu): membuat tampilan menu options
* onResume(): menampilkan keadaan saat keadaan resume terjadi
* onPause(): menampilkan keadaan saat keadaan pause terjadi
* MonsterListener: menjalankan menu MonsterParty
* PokedexListener: menjalankan menu Pokedex
* PlayerListener: menjalankan menu Profile
* BagListener: menjalankan menu Bag
* MonsterPartys(): menampilkan menu MonsterParty
* PokedexListen(): menampilkan menu Pokedex
* PlayerListen(): menampilkan menu Profile
* BagListen(): menampilkan menu Bag
* OnToCityListener(int): menjalankan menu City
* ShoreArea

Class ShoreArea menampilkan grafis dari Shore

* Atribut
* Int TAG
* Int scaling
* Int cur\_frame
* Bitmap background
* Paint cur\_background
* Bitmap background2
* Bitmap shader0
* Bitmap shader1
* Bitmap shader2
* PosX: koordinat X dari Player
* PosY: koordinat Y dari Player
* stepped: menampilkan area yang bisa dilalui oleh Player
* surfed: menampilkan area yang bisa dilalui (dengan swim) oleh Player
* int xOffset
* inf yOffset
* Method
* Init(): inisialisasi Shore
* ShoreArea(): constructor Shore
* DrawBackground(Canvas): menampilkan grafis background Outer
* Stepped(): Menampilkan area yang bisa dilalui Player
* Surfed(): menampilkan area yang bisa dilalui Player dengan swim
* getOneMove(): getter OneMove
* forceChangeOffset(int, int): mengubah nilai xOffset dan yOffset
* getxOffset(): getter xOffset
* getyOffset(): getter yOffset
* ChangeOffset(int, int, int, int): mengubah nilai xOffset dan yOffset
* ShoreView

Class ShopView digunakan untuk menampilkan grafis Shop

* Atribut
* Int screenWidth
* Int screenHeight
* ShoreViewListener mShoreViewListener
* String TAG
* Thread GameLoopThread
* Matrix matrix
* Pause\_event: menandai apakah permainan sedang dalam keadaan pause
* Hero: Pemain
* Shore: menu Shore
* Int CallAct
* Method
* ShoreView(Context): constructor dengan parameter
* screenSize(int, int): setter ScreenWidth dan ScreenHeight
* SetActivityFrom(String): menentukan posisi Hero berdasarkan activity sebelumnya
* surfaceChanged(SurfaceHolder, int, int, int)
* surfaceCreated(SurfaceHolder)
* surfaceDestroyed(SurfaceHolder)
* initThread(): inisialisasi HomeView
* releaseThread()
* update(): meng-update map Home
* render(Canvas): render gambar Home
* onTouchEvent(MotionEvent): membuat event berdasarkan input Touch yang diterima
* setPauseEvent(Boolean): mengeset event Pause
* isToChangeView(): memeriksa apakah pemain keluar dari Shore
* isCanTransSwim(): memeriksa apakah pemain dapat menggunakan Swim
* trans\_hero(): membuat pemain ke area dimana pemain bisa menggunakan Swim
* drawNightMode(canvas): menampilkan Shore pada saat malam
* ChangeToLand(): memindahkan pemain dari area Sea ke area Land (darat)
* StadiumActivity

Class StadiumActivity digunakan untuk menampilkan aktivitas pada Stadium

* Atribut
* String TAG
* String TOTAL\_BET
* DisplayMetrics metrics
* StadiumView GameView
* TextView interact
* Int i\_interact
* Boolean isInteraction
* Int i\_interact
* LinearLayout boxButton
* LinearLayout boxBet
* EditText totalBet
* LinearLayout GameWidgets
* Boolean isViewMenu
* ImageView menu\_button
* Method
* onCreate(Bundle): membuat tampilan Shop
* onBackPressed()
* onCreateOptionsMenu(Menu): membuat tampilan menu options
* onPause(): menampilkan keadaan saat keadaan pause terjadi
* MonsterListener: menjalankan menu MonsterParty
* PokedexListener: menjalankan menu Pokedex
* PlayerListener: menjalankan menu Profile
* BagListener: menjalankan menu Bag
* MonsterPartys(): menampilkan menu MonsterParty
* PokedexListen(): menampilkan menu Pokedex
* PlayerListen(): menampilkan menu Profile
* BagListen(): menampilkan menu Bag
* OnToInteract(int): menjalankan interaksi antara penjaga Stadium dan pemain
* InteractListener: menjalankan menu Interact
* ParticipateListener: menjalankan interaksi apabila pemain ingin mengikuti pertandingan di stadium
* CancelListener: keluar dari interaksi
* ParticipateButtonListener: menjalankan menu apabila tombol Participate ditekan
* CancelButtonListener: keluar dari interaksi sesudah tombol Participate ditekan
* OnToCityListener(int): menjalankan menu City
* StadiumView

Class StadiumView digunakan untuk menampilkan grafis Stadium

* Atribut
* Float density
* Int screenWidth
* Int screenHeight
* StadiumViewListener mStadiumViewListener
* String TAG
* Thread GameLoopThread
* Matrix matrix
* Pause\_event: menandai apakah permainan sedang dalam keadaan pause
* Hero: Pemain
* Shop: menu Shop
* Int CallAct
* Method
* StadiumView(Context): constructor dengan parameter
* changeScreen(int, int): setter ScreenWidth dan ScreenHeight
* SetActivityFrom(String): menentukan posisi Hero berdasarkan activity sebelumnya
* surfaceChanged(SurfaceHolder, int, int, int)
* surfaceCreated(SurfaceHolder)
* surfaceDestroyed(SurfaceHolder)
* initThread(): inisialisasi HomeView
* releaseThread()
* update(): meng-update map Home
* render(Canvas): render gambar Home
* onTouchEvent(MotionEvent): membuat event berdasarkan input Touch yang diterima
* setPauseEvent(Boolean): mengeset event Pause
* isToChangeView(): memeriksa apakah pemain keluar dari Stadium
* isFrontDoor(): memeriksa apakah pemain berada di depan Stadium

resetCallAct(): me-reset nilai CallAct menjadi 1

* Class Batu

Kelas atribut dan method” mengatur batu pada outer area dan sea

* + - * + Atribut
* monster: monster bitmap
* PosIX: posisi batu dalam integer X
* PosIY: posisi batu dalam integer Y
* PosX: posisi batu dalam real X
* PosY: posisi batu dalam real Y
* stepped: boolean posisi x,y ada batu tidak
* xOffset: pergeseran batu jika layar bergeser X
* yOffset: pergeseran batu jika layar bergeser Y
* fps: frame per second
* kanan: boolean untuk ke kanan
* kiri: boolean untuk ke kiri
* atas: boolean untuk ke atas
* bawah: boolean untk ke bawah
* move\_act: aksi gerakan kemana
* no\_frame: nomor frame
* isLog: boolean log
  + - * + Method
* init(): inisisalisasi batu
* Batu(int, int): konstruktor batu dengan posisi X, Y
* getPosX(): getter PosIX
* getPosY(): getter PosIY
* update\_pos(int,int,int): update posisi batu dengan inputan X,Y,arah
* Draw(Canvas, int): menggambar batu ke canvas
* setOffset(int,int): pergeseran batu jika layar bergeser
* dDraw(): void
* ChangePos(): mengganti posisi
* Kanan(): mengganti kanan menjadi true
* Kiri(): mengganti kiri menjadi true
* Atas(): mengganti atas menjadi true
* Bawah(): mengganti bawah menjadi true
* AddKanan(): menambah ke kanan
* AddKiri(): menambah ke kiri
* AddAtas(): menambah ke atas
* AddBawah(): menambah ke bawah
* setStepped(): setter steped
* Class Hero

Kelas berisi atribut dan method dari hero dalam permainan

* + - * + Atribut
* hero: hero bitmap
* PosIX: posisi default hero X
* PosIY: posisi default hero Y
* PosX: posisi hero dalam real X
* PosY: posisi hero dalam real Y
* sprite\_no: nomor yang merepresentasikan sdraw
* wps: banyaknya jalan dalam satu detik
* fps: frame per second
* xOffset: offset hero untuk layar yang bergeser X
* yOffset: offse hero untuk layar yang bergeser Y
* spriteHeight: nomor sdraw tinggi
* no\_frame: nomor frame
* kanan: boolean untuk ke kanan
* kiri: boolean untuk ke kiri
* atas: boolean untuk ke atas
* bawah: boolean untk ke bawah
* move\_b: move before
* move\_act: aksi gerakan kemana
* stepped: mengatur posisi yang dapat dilewati dan tidak oleh hero
* surfed: mengatur posisi yang untuk bisa berenang oleh hero
* status: status hero
* hero0: bitmap hero0
* hero1: bitmap hero1
* hero2: bitmap hero2
* hero3: bitmap hero3
* hero4: bitmap hero4
* hero5: bitmap hero5
* hero6: bitmap hero6
* hero7: bitmap hero7
* hero8: bitmap hero8
* hero9: bitmap hero9
* hero10: bitmap hero10
* hero11: bitmap hero11
* scaling: skala
* sum\_walk: jumlah walking hero
* maxWalkSession: pergantian waktu malam siang jika sum\_walk melebihi
  + - * + Method
* initHero(): inisisasi untuk hero
* Hero(): konstruktor Hero
* Draw3(Canvas): menggambar hero ke kanvas
* setOffset(int,int): setter x y offset
* getRealX(int): getter PosX dengan masukan move
* getRealY(int): getter PosY dengan masukan move
* sumWalkCheck(): mengecheck jumlah walking hero
* setOffset(int,int): pergeseran batu jika layar bergeser
* dDraw(): void
* ChangePos(): mengganti posisi
* Kanan(): mengganti kanan menjadi true
* Kiri(): mengganti kiri menjadi true
* Atas(): mengganti atas menjadi true
* Bawah(): mengganti bawah menjadi true
* AddKanan(): menambah ke kanan
* AddKiri(): menambah ke kiri
* AddAtas(): menambah ke atas
* AddBawah(): menambah ke bawah
* Change\_status(int): mengganti status hero
* setStatus(): getter Status
* setStepped(boolean): setter stepped
* setSurfed(boolean): setter surfed
* setHeroPosition(int,int): setter posisi hero
* getX(): getter PosIX
* getY(): getter PosIX
* getPointer(): getter untuk move\_act
* getSpriteHeight(): getter hero.getHeight()
* getSpriteWidht(): getter hero.getWidth()
* getSumWalk(): getter jumlah walking hero
* setSumWalk(int): setter sum\_walk
* recycleAllBitmap(): recycle bitmap
* changeImageStatus(): mengganti status image
* Class MonsterArea

Kelas berisi atribut dan method dari monter-monter yang keluar di area

* + - * + Atribut
* monster: monster bitmap
* PosIX: posisi default hero X
* PosIY: posisi default hero Y
* PosX: posisi hero dalam real X
* PosY: posisi hero dalam real Y
* fps: frame per second
* xOffset: offset hero untuk layar yang bergeser X
* yOffset: offse hero untuk layar yang bergeser Y
* no\_frame: nomor frame
* kanan: boolean untuk ke kanan
* kiri: boolean untuk ke kiri
* atas: boolean untuk ke atas
* bawah: boolean untk ke bawah
* move\_act: aksi gerakan kemana
* stepped: mengatur posisi yang dapat dilewati dan tidak oleh hero
* scaling: skala
* type: type monster yang mendekat menjauh atau random
* rangearea: area yang dijelajahi monster
* isLog: log
  + - * + Method
* initHero(): inisisasi untuk monsterArea
* MonterArea(int,int,int): konstruktor monterArea dengan masukkan posisi x,y dan type
* getPosX(int): getter PosIX dengan masukan move
* getPosY(int): getter PosIY dengan masukan move
* update\_pos(int,int): mengupdate posisi hero dalam x,y
* decideMendekatMovement(int,int): pengaturan monster mendekat
* decideMenjauhMovement(int,int): pengautran monster menjauh
* abs(int): absolut nilai
* setOffset(int,int): pergeseran batu jika layar bergeser
* dDraw(): void
* ChangePos(): mengganti posisi
* Kanan(): mengganti kanan menjadi true
* Kiri(): mengganti kiri menjadi true
* Atas(): mengganti atas menjadi true
* Bawah(): mengganti bawah menjadi true
* AddKanan(): menambah ke kanan
* AddKiri(): menambah ke kiri
* AddAtas(): menambah ke atas
* AddBawah(): menambah ke bawah
* setStepped(boolean): setter stepped
* Class Combinatorium

Kelas berisi atribut dan method dari combinatrium monster

* + - * + Atribut
* scalling: skala
* TAG: String
* background: background combinatorium dalam bitmap
* PosX: posisi X
* PosY: posisi Y
* xOffset: offset x
* yOffset: offset y
* stepped: posisi mana saja yang bisa dilalui dan tidak
  + - * + Method
* init(): inisiasi home
* Home(int,int): konstruktor
* Draw(Canvas): menggambar background
* Stepped(): getter stepped
* getXOffset(): getter xOffset
* getYOffset(): getter yOffset
* Class Shop

Kelas berisi atribut dan method dari shop

* + - * + Atribut
* scalling: skala
* TAG: String
* background: background shop dalam bitmap
* PosX: posisi X
* PosY: posisi Y
* xOffset: offset x
* yOffset: offset y
* stepped: posisi mana saja yang bisa dilalui dan tidak
  + - * + Method
* init(): inisiasi shop
* shop(int,int): konstruktor
* Draw(Canvas): menggambar background
* Stepped(): getter stepped
* getXOffset(): getter xOffset
* getYOffset(): getter yOffset
* Class Home

Kelas berisi atribut dan method dari home

* + - * + Atribut
* scalling: skala
* background: background home dalam bitmap
* PosX: posisi X
* PosY: posisi Y
* xOffset: offset x
* yOffset: offset y
* stepped: posisi mana saja yang bisa dilalui dan tidak
  + - * + Method
* init(): inisiasi home
* Home(int,int): konstruktor
* DrawBackground(Canvas): menggambar background
* Stepped(): getter stepped
* getXOffset(): getter xOffset
* getYOffset(): getter yOffset
* Class Stadium

Kelas berisi atribut dan method dari Stadium

* + - * + Atribut
* scalling: skala
* background: background stadium dalam bitmap
* PosX: posisi X
* PosY: posisi Y
* xOffset: offset x
* yOffset: offset y
* stepped: posisi mana saja yang bisa dilalui dan tidak
  + - * + Method
* init(): inisiasi Stadium
* Stadium(int,int): konstruktor
* Draw (Canvas): menggambar background
* Stepped(): getter stepped
* getXOffset(): getter xOffset
* getYOffset(): getter yOffset
* Class BattleOuterArea

Kelas berisi atribut dan method dari Battle di aouter area

* + - * + Atribut
* TAG: String
* isNextLevel: apakah player naik level
* blackot: true jika malam
* FLOOR: floor\_battle String
* Current\_floor: posisi lantai sekarang
* skill\_use: skill yang digunakan
* skill\_use\_turn: skill yang dipakai dalam giliran menyerang
* interaction\_skill\_use: interaksi skill
* metrics: DisplayMetrics
* Game: relativeLayout battle
* Battle: BattleView
* Interact: TextView dalam battle
* number\_interaction: nomor interaksi battle
* finnalize: hasil akhir battle
* boxFront: LinearLayout
* fight: TextView untuk battle fight
* monter: TextView untuk monster
* bag: TextView untuk bag
* escape: TextView untuk escape
* skillUse: LinearLayout skill
* infoSkill: info skill TextView
* skill1: skill1 TextView
* skill2: skill2 TextView
* skill3: skill3 TextView
* skill4: skill4 TextView
* db: database pokeranch
* foe\_id: id monster
* foeMonster: MonsterBattle yang akan dilawan
* palyerMonster: MonsterBattle untuk player
* IsPlayerFrist: boolean apakkah player memulai battle duluan
* foeSkill: skill monster
* playerSkill: skill monster player
* pending\_effect: boolean untuk pending efek
* player\_skill\_selected: skill yang dipilih player
* monster\_party\_number: monster party number
* exp\_get: exp yang didapat dari battle
* money\_get: uang yang didapat dari battle
* try\_escape: keadaan escape
* EscapeSuccesfully: boolean kesuksesan escape
* monster\_skill\_use: skill monster yang dipakai
  + - * + Method
* onCreate(Bundle): void
* init\_battle\_content(): inisiasi battle
* resume\_battle\_content(): resume battle
* battle\_content(): battle content
* onBackPressed(): pengaturan jika Back di tekan
* onResume(): resume
* onDestroy(): destroy db
* onPause(): pause
* onCreateOptionsMenu(Menu): boolean menu
* InteractorListener: kelas listener interaksi
* FirstInteraction(int): memanggil interaksi pertama
* Check\_finalize(): mengechek hasil battle
* interact(int): interaksi
* outFromBattleMode(): kelaur dari battle
* FightListener: kelas listener untuk battle
* MonsterListener: kelas listener untuk monster
* BagListener: kelas listener untuk bag
* EscapeListener: kelas listener untuk escape
* Skill1Listener: kelas listener untuk skill 1
* Skill2Listener: kelas listener untuk skill 2
* Skill3Listener: kelas listener untuk skill 3
* Skill4Listener: kelas listener untuk skill 4
* InfoSkillListener: kelas listener untuk info skill
* using\_skill(): menggunakan skill
* use\_skill(): penggunaan skill
* FailedEscape(): gagal escape
* player\_use\_skill(): player menggunakan skill
* foe\_use\_skill(): monster menggunakan skill
* MonsterBattle: kelas monsterbattle
* Class BattleStadium

Kelas berisi atribut dan method dari Battle stadium

* + - * + Atribut
* TAG: String
* isNextLevel: apakah player naik level
* blackot: true jika malam
* skill\_use: skill yang digunakan
* skill\_use\_turn: skill yang dipakai dalam giliran menyerang
* interaction\_skill\_use: interaksi skill
* metrics: DisplayMetrics
* Game: relativeLayout battle
* Battle: BattleView
* Interact: TextView dalam battle
* number\_interaction: nomor interaksi battle
* finnalize: hasil akhir battle
* boxFront: LinearLayout
* fight: TextView untuk battle fight
* monter: TextView untuk monster
* bag: TextView untuk bag
* escape: TextView untuk escape
* skillUse: LinearLayout skill
* infoSkill: info skill TextView
* skill1: skill1 TextView
* skill2: skill2 TextView
* skill3: skill3 TextView
* skill4: skill4 TextView
* db: database pokeranch
* foe\_id: id monster
* foeMonster: MonsterBattle yang akan dilawan
* palyerMonster: MonsterBattle untuk player
* IsPlayerFrist: boolean apakkah player memulai battle duluan
* foeSkill: skill monster
* playerSkill: skill monster player
* pending\_effect: boolean untuk pending efek
* player\_skill\_selected: skill yang dipilih player
* total\_bet: total taruhan
* monster\_party\_number: monster party number
* exp\_get: exp yang didapat dari battle
* money\_get: uang yang didapat dari battle
* try\_escape: keadaan escape
* EscapeSuccesfully: boolean kesuksesan escape
* monster\_skill\_use: skill monster yang dipakai
  + - * + Method
* onCreate(Bundle): void
* init\_battle\_content(): inisiasi battle
* resume\_battle\_content(): resume battle
* battle\_content(): battle content
* onBackPressed(): pengaturan jika Back di tekan
* onResume(): resume
* onDestroy(): destroy db
* onPause(): pause
* onCreateOptionsMenu(Menu): boolean menu
* InteractorListener: kelas listener interaksi
* FirstInteraction(int): memanggil interaksi pertama
* Check\_finalize(): mengechek hasil battle
* interact(int): interaksi
* outFromBattleMode(): kelaur dari battle
* FightListener: kelas listener untuk battle
* MonsterListener: kelas listener untuk monster
* BagListener: kelas listener untuk bag
* EscapeListener: kelas listener untuk escape
* Skill1Listener: kelas listener untuk skill 1
* Skill2Listener: kelas listener untuk skill 2
* Skill3Listener: kelas listener untuk skill 3
* Skill4Listener: kelas listener untuk skill 4
* InfoSkillListener: kelas listener untuk info skill
* using\_skill(): menggunakan skill
* use\_skill(): penggunaan skill
* FailedEscape(): gagal escape
* player\_use\_skill(): player menggunakan skill
* foe\_use\_skill(): monster menggunakan skill
* MonsterBattle: kelas monsterbattle
* Class BattleView

Kelas berisi atribut dan method dari Battle view

* + - * + Atribut
* TAG: String
* screenWidth: ukuran lebar layar
* screenHeight: ukuran tinggi layar
* scale: skala
* scaling: skala
* thread: thread untuk looping game
* matrix: matrix
* mBattleView: battle view
* background: Paint
* battleGrass: battleGrass bitmap
* foe\_box: kotak monster bitmap
* player\_box: kotak player bitmap
* health\_bar: health bar bitmap
* FirstTransition: keadaan transisi
* wait\_event: keadaan wait event
* callAct: panggil action
* TextName: Paint text name
* foe\_number: nomor monster
* foe\_bitmap: monster bitmap
* foe\_name: nama monster
* foe\_level: level monster
* foe\_status: status monster bitmap
* player\_name: nama player
* player\_level: level player
* player\_status: player status bitmap
* player\_maxHP: besar max HP player
* player\_curHP: besar HP player sekarang
* trans: besar nilai transisi
* foeGrass: posisi grass monster
* foebitmap: posisi monster bitmap
* foeBox: posisi monster box
* foeName: posisi nama monster
* foeLevel: posisi level monster
* foeHPBar: posisi monster HP Bar
* foeStatus: posisi status monter
* playerGrass: posisi grass player
* playerbitmap: posisi player bitmap
* playerBox: posisi player box
* playerName: posisi nama player
* playerLevel: posisi level player
* playerHPBar: posisi player HP Bar
* playerStatus: posisi status player
* isChangeFoeHp: apakah HP monster berubah
* isChangePlayerHP: apakah HP player berubah
* foeMaxHP: monster max HP
* foeCurHP: monster HP sekarang
* foeHPDraw: besar perubahan HP Bar monster
* playerMaxHP: player max HP
* playerCurHP: player HP sekarang
* playerHPDraw: besar perubahan HP Bar player
  + - * + Method
* BattleView(Context): konstruktor
* screenSize(float, float): mengatur ukuran layar
* foe\_picture(int): setter nomor monster
* surfaceChanged(SurfaceHolder, int, int, int): merubah surface
* surfaceCreated(SurfaceHolder): membuat surface
* surfaceDestroyed(Surfaceolder): mengapus surface
* initThread(): inisiasi thread
* releaseThread(): merilis thread
* update(): mengupdate data
* render(Canvas): merender gambar
* foe\_hp(int): menampilkan HP monster
* update\_foe\_hp(int,int,int): mengupdate HP monster
* player\_hp(int): menampilkan HP player
* update\_player\_hp(int,int,int): mengupdate HP player
* onTounchEvent(MotionEvent): adakah touch event
* isWaitEvent(): mengembalikan wait\_event
* FoeName(String): setter nama monster
* FoeLevel(String): setter level monster
* FoeStat(int): setter status monster
* PlayerName(String): setter nama player
* PlayerLevel(String): setter level palyer
* PlayerStatus(int): setter status player
* PlayerMaxHP(String): setter Max HP player
* PlayerCurHP(String): setter HP player sekarang
* status(int): mengembalikan status bitmap
* Position: kelas Position
* Class UseItem

Kelas berisi atribut dan method dari memakai item

* + - * + Atribut
* metrics: DisplayMatrics
* db: pokeranch database
* lispLayerItem: list player item
* quantity: banyaknya item
* itemDescription: deskripsi item
* useitem: button memakai item
* item\_selected: item yang dipilih
* empty: kosong
* ball\_use: ball yang dipakai
* isUseItem: boolean apakah item dipakai
  + - * + Method
* onCreate(Bundle): void
* usetheitem(): memakai item
* onCreateOptionMenu(Menu): mengmebalikan menu
* onbackPressed(): jika tombol back di tekan
* onDestroy(): menghapus db
* Class Cridits

Kelas berisi atribut dan method dari kridit game dengan menginclude xml

* + - * + Atribut
* EXTRA\_MESSAGE: String
  + - * + Method
* onCreate(Bundle): void
* onCreateOptionMenu(Menu): mengmebalikan menu
* Class EmptyUserDialog

Kelas berisi atribut dan method dari user kosong dengan menginclude xml

* + - * + Method
* onCreateOptionMenu(Menu): mengmebalikan menu
* Class LoadUser

Kelas berisi atribut dan method dari meload user

* + - * + Atribut
* listPlayerReg: ArrayList dari player yang terdaftar
* mCallBack: LoadUserListener
  + - * + Method
* LoadUserListener: interface meload nama
* onAttach(Activity): mengattach activity
* onCreateDialog(Bundle): membuat dialog
* Class MainMenu

Kelas berisi atribut dan method dari menu utama dengan menginclude xml

* + - * + Method
* onCreateView(LayoutInflater, ViewGroup, Bundle): membuat view
* Class NewUser

Kelas berisi atribut dan method dari meload user baru

* + - * + Atribut
* TAG: String
* mCallBack: new user listener
* message: String
* username: user edit text
  + - * + Method
* OnNewUserListener: interface dialog
* onAttach(Activity): mengattach activity
* onCreateDialog(Bundle): membuat dialog
* onClick(DialodInterface,int): jika di klik
* Bag

Class Bag berisi item-item yang dimiliki pemain

* Atribut
* String TAG
* String KEY\_ROD\_USE
* String KEY\_AT\_SEA
* DisplayMetrics metrics
* PokeDB db
* ArrayAdapter<String> adapterItem
* ArrayAdapter<String> adapterMedicine
* ArrayAdapter<String> adapterTM
* ArrayAdapter<String> adapterTools
* LinearLayout mainLayout
* LinearLayout itemInfo
* LinearLayout blank
* ListView listleftView
* TextView ItemTitle
* TextView ItemDescription
* TextView tm\_type
* TextView tm\_power
* TextView tm\_accuracy
* TextView tm\_PP
* Button use
* Int selected\_bag
* Int selected\_item
* Int filterItem
* Boolean isRodCanUse
* Int useItemId
* Method
* onCreate(Bundle): membuat tampilan Bag
* StartActivityFrom(Intent): menjalankan menu Bag berdasarkan activity yang dilakukan pemain sebelumnya
* onCreateOptionsMenu(Menu): membuat tampilan menu options
* onOptionsItemSelected(MenuItem): mengembalikan info Item yang dipilih pemain
* onResume(): menjalankan keadaan Resume
* onBackPressed()
* onDestroy(): destructor
* showItemCat(): menampilkan Item yang dimiliki pemain
* showMedicineCat(): menampilkan Item Medicine yang dimiliki pemain
* showTMCat(): menampilkan Item TM yang dimiliki pemain
* showToolsCat(): menampilkan Item Tools yang dimiliki pemain
* ItemListener: menjalankan menu Item
* MedicineListener: menjalankan menu Item Medicine
* TMListener: menjalankan menu Item TM
* ToolsListener: menjalankan menu Item Tools
* Use\_item(): dijalankan apabila pemain menggunakan item
* Buy

Class Buy dijalankan apabila pemain menjalankan menu Buy di Shop

* Atribut
* DisplayMetrics metrics
* PokeDB db
* ArrayList<String> listItemSell
* TextView currentPlayerMoney
* TextView itemDescription
* Button buy
* Item item\_selected
* Method
* onCreate(Bundle): membuat tampilan Buy
* onCreateOptionsMenu(Menu): membuat tampilan menu options
* onOptionsItemSelected(MenuItem): mengembalikan info Item yang dipilih pemain
* onResume(): menjalankan keadaan Resume
* onBackPressed()
* onDestroy(): destructor
* ChangeSkillMonster

Class ChangeSkillMonster digunakan apabila pemain ingin mengubah skill yang dimiliki oleh monster dengan menggunakan item TM. ChangeSkillMonster hanya bisa digunakan apabila skill monster mencapati batas maksimal skill, yaitu 4.

* Atribut
* String INSTALL\_USE\_ITEM\_ID
* String INSTALL\_SKILL\_ID
* String INSTALL\_MONSTER\_ID
* PokeDB db
* TextView bool
* Skill skill: skill yang akan ditambahkan pada monster
* Skill skill1: skill 1 yang dimiliki monster
* Skill skill2: skill 2 yang dimiliki monster
* Skill skill3: skill 3 yang dimiliki monster
* Skill skill4: skill 4 yang dimiliki monster
* Int index\_skill\_use
* PlayerMonster monster
* Int item\_id
* Method
* onCreate(Bundle): membuat tampilan ChangeSkillMonster
* onCreateOptionsMenu(Menu): membuat tampilan menu options
* onDestroy(): destructor
* Combine

Class Combine digunakan apabila pemain ingin menggabungkan 2 monster. Class ini hanya dapat dijalankan saat pemain berada di Kombinatorium

* Atribut
* DisplayMetrics metrics
* PokeDB db
* Int monster\_selected
* Int monster1sel
* Int monster2sel
* TextView monster1selected
* TextView monster2selected
* ListView listParty
* TextView bool
* Bitmap monster
* ImageView Image
* Method
* onCreate(Bundle): membuat tampilan Combine
* Boolean IsCombinable(int, int): memeriksa apakah 2 monster yang dipilih bisa dikombinasi
* onCreateOptionsMenu(Menu): membuat tampilan menu options
* onBackPressed()
* onDestroy(): destructor
* EggHatch

Class EggHatch digunakan untuk menetaskan Monster Egg yang dimiliki pemain. Class ini hanya dapat dijalankan ketika pemain berada di Home

* Atribut
* DisplayMetrics metrics
* LinearLayout Game
* PokeDB db
* Monster monster
* EditText newMonster
* TextView error\_info
* Button submit
* Button ok
* Method
* onCreate(Bundle): membuat tampilan EggHatch
* onBackPressed()
* onDestroy(): destructor
* get\_new\_monster\_number(): getter monster\_num dari monster yang baru ditetaskan
* ook(): menjalankan menu Home (keadaan setelah Monster Egg menetas/tidak menetas)
* Sell

Class Sell dijalankan apabila pemain menjalankan menu Sell di Shop

* Atribut
* DisplayMetrics metrics
* PokeDB db
* ArrayList<String> listItemSell
* LinearLayout listViewItemSell
* TextView itemPlayer
* Init itemWillSell
* Button sellButton
* LinearLayout blank
* Context thisContext
* Method
* onCreate(Bundle): membuat tampilan Sell
* onCreateOptionsMenu(Menu): membuat tampilan menu options
* onResume(): menjalankan keadaan Resume
* onBackPressed()
* onDestroy(): destructor
* getItemWillSell(): getter itemWillSell
* setItemWillSell(): setter itemWillSell
* ItemUse

Class ItemUse dijalankan apabila pemain menggunakan Item

* Atribut
* String ITEM\_USE
* PokeDB db
* TextView bool
* Bitmap monster
* ImageView image
* Button useItem
* Int scaling
* Int monster\_selected
* Int itemIdUse
* Method
* onCreate(Bundle): membuat tampilan ItemUse
* onCreateOptionsMenu(Menu): membuat tampilan menu options
* onBackPressed()
* onDestroy(): destructor
* MonsterParty

Class MonsterParty digunakan untuk menampilkan Monster yang ada pada Party pemain

* Atribut
* String IS\_SWIM\_TRANS
* String IS\_CAN\_CUT
* String IS\_USE\_PUSH
* String TAG
* DisplayMetrics metrics
* PokeDB db
* TextView bool
* Bitmap monster
* ImageView image
* PlayerMonster MonsterParty
* Button skillSwim: tombol untuk menggunakan skill Swim
* Button skillCut: tombol untuk menggunakan skill Cut
* Button skillPush: tombol untuk menggunakan skill Push
* Boolean usingSkill: status pemain apakah pemain sedang menggunakan Skill
* Method
* onCreate(Bundle): membuat tampilan MonsterParty
* check\_skill(): memeriksa apakah ada monster pada party yang mempunyai skill Swim, Cut, atau Push
* onCreateOptionsMenu(Menu): membuat tampilan menu options
* SwimListener: menjalankan menu Swim
* CutListener: menjalankan menu Cut
* PushListener: menjalankan menu Push
* onDestroy(): destructor
* Pokedex

Class Pokedex digunakan untuk menampilkan Pokedex yang dimiliki pemain

* Atribut
* DisplayMetrics metrics
* PokeDB db
* ArrayList<String> listSpesies: list monster yang ada pada game
* MonsterNum: Nomor Monster
* Elemen: elemen dari Monster
* HP: HP maksimal dari Monster
* Attack: nilai Attack dari Monster
* Defense: nilai Defense dari Monster
* Speed: nilai Speed dari Monster
* Monster: gambar monster
* ImageView monsterPicture
* Method
* onCreate(Bundle): membuat tampilan Pokedex
* onCreateOptionsMenu(Menu): membuat tampilan menu options
* onBackPressed()
* onDestroy(): destructor
* Profile

Class Profile digunakan untuk menampilkan Profil Pemain

* Method
* onCreate(Bundle): membuat tampilan Buy
* onCreateOptionsMenu(Menu): membuat tampilan menu options onBackPressed()
* Class Item

Class Item adalah kelas yang berisi semua item di PokeRanch dan method-method yang digunakan dalam berbagai pemrosesan item

* + - * + Atribut
* ItemNumb: nomor ID item
* Name: nama item
* Type: tipe dari item
* BuyPrice: besar harga beli item
* SellPrice: besar harga jual item
* ItemValue: besar nilai value suatu item
* Description: deskripsi item
  + - * + Method
* getItemNumb(): getter ItemNumb
* getName(): getter Name
* getType(): getter Type
* getBuyPrice(): getter BuyPrice
* getSellPrice(): getter SellPrice
* getItemValue(): getter ItemValue
* getDescription(): getter Description
* setItemNumb(): setter ItemNumb
* setName(): setter Name
* setType(): setter Type
* setBuyPrice(): setter BuyPrice
* setSellPrice(): setter SellPrice
* setItemValue(): setter ItemValue
* setDescription(String): setter Description
* ItemTabel: abstrak kelas untuk membuat tabel item dlm bentuk SQLite
* ItemCD: abstrak kelas mengisi tabel item dengan SQLite
* Item(String,SQLiteDatabase): konstruktor item dengan masukan ItemName dan tabel item
* getItemNameAndPrice(SQLiteDatabase): menghasilkan list nama dan harga untuk ditampilkan di Shop
* getName(SQLiteDatabase,int): menghasilkan nama item dengan masukkan tabel item dan item\_num
* Item(int,SQLiteDatabase): konstruktor item dengan masukkan Id item dan tabel item
* getFirstItemRead(SQLiteDatabase): Menampilkan semua isi tabel item
* Class Monster

Class Monster adalah kelas yang berisi semua monster di PokeRanch dan method-methodnya

* + - * + Atribut
* ID: nomor id monster
* Spesies: nama spesies
* Elemen\_No: nomor elemen monster
* BaseHP: HP normal dilevel 50
* BaseAttack: Attack normal dilevel 50
* BaseDefense: Defense normal dilevel 50
* BaseSpeed: Speed normal dilevel 50
* DefaultSkill1: nomor id skill1 default monster
* DefaultSkill2: nomor id skill2 default monster
* DefaultSkill3: nomor id skill3 default monster
* DefaultSkill4: nomor id skill4 default monster
* MinEvoLevel: level minimal untuk melakukan evolusi
* NextEvoNo: nomor monster evo jika monster dapat evolusi
* Gentanyangan: menyatakan monster kelar malam atau siang hari
  + - * + Method
* getItemNumb(): getter ItemNumb
* getName(): getter Name
* getType(): getter Type
* getBuyPrice(): getter BuyPrice
* getSellPrice(): getter SellPrice
* getItemValue(): getter ItemValue
* getDescription(): getter Description
* setItemNumb(): setter ItemNumb
* setName(): setter Name
* setType(): setter Type
* setBuyPrice(): setter BuyPrice
* setSellPrice(): setter SellPrice
* setItemValue(): setter ItemValue
* setDescription(String): setter Description
* ItemTabel: abstrak kelas untuk membuat tabel item dlm bentuk SQLite
* ItemCD: abstrak kelas mengisi tabel item dengan SQLite
* Item(String,SQLiteDatabase): konstruktor item dengan masukan ItemName dan tabel item
* getItemNameAndPrice(SQLiteDatabase): menghasilkan list nama dan harga untuk ditampilkan di Shop
* getName(SQLiteDatabase,int): menghasilkan nama item dengan masukkan tabel item dan item\_num
* Item(int,SQLiteDatabase): konstruktor item dengan masukkan Id item dan tabel item
* getFirstItemRead(SQLiteDatabase): Menampilkan semua isi tabel item
* Class Player

Class Player adalah kelas yang berisi informasi umum dari pemain

* Atribut:
* ID: nomor id dari player
* Name: nama player
* TotalPlayTime: lamanya player bermain
* Day\_or\_Night: menyatakan status waktu di game
* Cur\_Money: total uang sekarang
* Total\_Get\_Monet: total uang yang baru didapat
* Win: total kemenangan player di stadium
* Lose: total kekalahan player di stadium
* Score\_Stadium: total skor stadium player
* Method:
* getID(): getter ID.
* getName(): getter Name.
* getTotalTimePlay(): getter TotalPlayTime
* getDayOrNight(): getter Day\_or\_Night
* getCurMoney(): getter Cur\_Money
* getTotalGetMoney(): getter Total\_Get\_Money
* getWin(): getter Win
* getLose(): getter Lose
* getScoreStadium(): getter Score\_Stadium
* setID(int): setter ID
* setName(String): setter Name
* setTotalTimePlay(int): setter TotalPlayTime
* setDayOrNight(int): setter Day\_or\_Night
* setCurMoney(int): setter Cur\_Money
* setTotalGetMoney(int): setter Total\_Get\_Money
* setWin(int): setter Win
* setLose(int): setter Lose
* setScoreStadium(int) setter Score\_Stadium
* Player(): membuat objek player baru yang kosong
* Player(SQLiteDatabse, String): membuat player baru dan menyimpan dalam database
* PlayerTabel: abstrak kelas untuk membuat tabel Player dalam bentuk SQLite
* PlayerCD: abstrak kelas untuk mengisi tabel Player dengan SQLite
* nextPlayerId(SQLiteDatabase): mengembalikan Id Player selanjutnya pada tabel Player
* IsEmpty(SQLiteDatabase): memeriksa apakah tabel Player kosong atau tidak
* getArrayName(SQLiteDatabase): mengembalikan isi array pada kolom yang diminta
* getPlayer(SQLiteDatabase, String): mengembalikan info Player sesuai dengan nama player yang diminta
* saveDataToDatabase(SQLiteDatabase): menyimpan data Player pada tabel Player
* Class PlayerItem

Class PlayerItem berisi daftar Item yang dimiliki Player

* Atribut:
* Player\_Id: nomor ID dari Player
* Item\_Id: nomor ID dari Item
* Qty: jumlah item dengan ID tertentu yang dimiliki Player
* Name: nama dari Item
* Type: tipe dari Item
* Description: Deskripsi mengenai item
* Method:
* GetPlayerID(): getter Player\_Id
* GetItemId(): getter Item\_Id
* GetQty(): getter Qty
* GetName(): getter Name
* GetType(): getter Type
* GetDesc(): getter Description
* SetPlayerId(int): setter Player\_Id
* SetItemId(int): setter Item\_Id
* SetQty(int): setter Qty
* SetName(String): setter Name
* setType(int): setter Type
* setDesc(String): setter Description
* PlayerItemTable: abstrak kelas untuk membuat tabel PlayerItem dalam bentuk SQLite
* PlayerItemCD: abstrak kelas untuk mengisi tabel PlayerItem dalam bentuk SQLite
* initPlayerItem(int, SQLiteDatabase):
* GetNameAndQty(int, int): mengembalikan jumlah Item dengan Item\_Id yang di input yang dimiliki Player
* getNameItemCat(int): mengembalikan list nama item
* GetInfoItemCat(int, int): mengembalikan nama Item beserta deskripsi dan jumlahnya yang dimilki Player
* getNameItemBattle(int): mengembalikan list nama Item
* getNameMedicineCat(int): mengembalikan list nama Item Potion
* getInfoMedicineCat(int, int): mengembalikan list berisi info mengenai item yang bersifat potion
* getMedicineItemCat(int): mengembalikan list item Potion
* getNameTMCat(int): mengembalikan list berisi nama Skill(TM).
* getInfoTMCat(int, int): mengembalikan list berisi info mengenai Skill
* getTMItemCat(int,int): mengembalikan list Skill.
* getNameToolsCat(int,int): mengembalikan list item sepeda dan pancingan
* getInfoToolsCat(int,int): mengembalikan list berisi info mengenai item sepeda dan pancingan.
* GetData(): mengembalikan info berupa item yang dimiliki pemain
* GetAllData(SQLiteDatabase): mengembalikan tabel berisi item yang dimiliki pemain
* AddQty(int, int, int, SQLiteDatabase): menambahkan jumlah item pada database
* RemQTY\_itemNum(int, int): mengurangi jumlah item bernomor sesuai dengan input sesuai dengan jumlah yang diinput
* RemQTY(SQLiteDatabase, int, int): mengurangi jumlah item pada database
* GetAllItem(String): mengembalikan list tabel PlayerItem
* savePlayerItem(SQLiteDatabase): menyimpan data Player Item
* isHaveEgg(SQLiteDatabase, int): memeriksa apakah Player mempunyai Monster Egg
* Class PlayerMonster

Class PlayerMonster berisi daftar Monster yang dimiliki pemain

* Atribut:
* PlayerID: Nomor Id dari Player
* MonsterID: Nomor Id dari Monster pada Player
* MonsterNumber: Nomor Monster pada List Monster
* Name: Nama Monster
* Age: Umur Monster
* Level: Level dari Monster
* Exp: Exp yang dimiliki Monster
* HP: Health Point dari Monster
* CurHP: Health Point yang dimiliki Monster saat ini
* Attack: nilai Attack dari Monster
* Defense: nilai Defense dari Monster
* Speed: nilai Speed dari Monster
* Skill1: Skill yang dimiliki Monster
* Skill2: Skill yang dimiliki Monster
* Skill3: Skill yang dimiliki Monster
* Skill4: Skill yang dimiliki Monster
* PPSkill1: Banyaknya PP yang dimiliki oleh monster pada Skill1
* PPSkill2: Banyaknya PP yang dimiliki oleh monster pada Skill2
* PPSkill3: Banyaknya PP yang dimiliki oleh monster pada Skill3
* PPSkill4: Banyaknya PP yang dimiliki oleh monster pada Skill4
* Status\_Effect: Status Monster berdasarkan efek yang diterima monster saat battle atau penggunaan item
* Method:
* getPlayerID(): getter PlayerID
* getMonsterID(): getter MonsterID
* getMonsterNumber(): getter MonsterNumber
* getName(): getter Name
* getAge(): getter Age
* getLevel(): getter Level
* getExp(): getter Exp
* getHP(): getter HP
* getCurHP(): getter CurHP
* getAttack(): getter Attack
* getDefense(): getter Defense
* getSpeed(): getter Speed
* getSkill1(): getter Skill1
* getSkill2(): getter Skill2
* getSkill3(): getter Skill3
* getSkill4(): getter Skill4
* getPPSkill1(): getter PPSkill1
* getPPSkill2(): getter PPSkill2
* getPPSkill3(): getter PPSkill3
* getPPSkill4(): getter PPSkill4
* getStatusEffect(): getter StatusEffect
* setPlayerID(): setter PlayerID
* setMonsterID(): setter MonsterID
* setMonsterNumber(): setter MonsterNumber
* setName(): setter Name
* setAge(): setter Age
* setLevel(): setter Level
* setExp(): setter Exp
* setHP(): setter HP
* setCurHP(): setter CurHP
* setAttack(): setter Attack
* setDefense(): setter Defense
* setSpeed(): setter Speed
* setSkill1(): setter Skill1
* setSkill2(): setter Skill2
* setSkill3(): setter Skill3
* setSkill4(): setter Skill4
* setPPSkill1(): setter PPSkill1
* setPPSkill2(): setter PPSkill2
* setPPSkill3(): setter PPSkill3
* setPPSkill4(): setter PPSkill4
* setStatusEffect(): setter StatusEffect
* PlayerMonsterTable: abstrak kelas untuk membuat tabel PlayerMonster dalam bentuk SQLite
* PlayerMonsterCD: abstrak kelas untuk mengisi tabel PlayerItem dalam bentuk SQLite
* initPlayerMonster(SQLiteDatabase, int):
* PlayerMonster(): constructor
* PlayerMonster(SQLiteDatabase, int, int, String, int): constructor berdasarkan parameter
* DeleteMonsterById(int): Menghapus monster sesuai dengan id yang di input
* getSomeDataPM(SQLiteDatabase): mengembalikan informasi monster yang dimiliki Player
* getPlayerMonster(int): mengembalikan Monster berdasarkan nomor id yang di input
* getData(SQLiteDatabase):
* savePlayerMonster(SQLiteDatabase): menyimpan data Player Monster
* getTotalParty(): mengembalikan jumlah monster pada party pemain
* incEXPOfMonsterIndex(int, int): meningkatkan Exp Monster
* fullRestoreAllMonster(): mengembalikan semua status monster menjadi normal
* ChangeHPOfMonsterIndex(int, int): Mengubah nilai HP Monster
* ListNameMonsterDied(): List Monster yang mati
* incAgeOfMonster(): Menambah umur monster
* updatePlayerMonster(PlayerMonster): meng-update info Player Monster
* Class PlayerParty

Class PlayerParty menyimpan monster yang ada pada party pemain

* PlayerID: Nomor ID dari Player
* MonsterID: Nomor Monster pada database PlayerMonster
* MonsterNumber: Nomor Monster pada database Monster
* getPlayerID(): getter PlayerID
* getMonsterID(): getter MonsterID
* getMonsterNumber: getter Monster Number
* setPlayerID(): setter PlayerID
* setMonsterID(): setter MonsterID
* setMonsterNumber: setter Monster Number
* PlayerPartyTable: abstrak kelas untuk membuat tabel PlayerParty dalam bentuk SQLite
* PlayerPartyCD: abstrak kelas untuk mengisi tabel PlayerParty dalam bentuk SQLite
* initPlayerParty(SQLiteDatabase, int):
* PlayerParty():
* PlayerParty(int, int, int): constructor berdasarkan parameter
* PlayerParty(int, PlayerMonster): contructor berdasarkan parameter PlayerMonster
* getMonsterIndex(int): mengembalikan indeks ID PlayerMonster
* getMonsterNumberIndex(int): mengembalikan indeks MonsterNumber dari PlayerMonster
* getNameMonster(SQLiteDatabase): mengembalikan nama Monster
* DeleteMonsterAndPartyMonster(int): menghapus Monster dari List PlayerParty
* getFullInfoMonster(int): mengembalikan info salah satu Monster yang ada pada list PlayerParty
* savePlayerPartyToDB(SQLiteDatabase, int): menyimpan data PlayerParty
* GetData(SQLiteDatabase): mengembalikan data monster pada PlayerParty
* isCanAddToParty(): memeriksa apakah List PlayerParty dapat ditambah
* getTotalParty(): mengembalikan size List PlayerParty
* Class Skill

Class Skill adalah kelas yang berisi semua skill di PokeRanch dan method-method yang digunakan dalam pemrosesan skill

* + - * + Atribut
* no\_skill: nomor ID skill
* Name: nama skill
* Target: target dari skill
* Elemen\_No: nomor elemen dari skill
* Power: besar power dari skill
* Power\_Effect\_No: nomor power efek skill
* Prob\_Stat\_Effect: besar probabilitas efek skill
* Skill\_Evo\_To: Evo skill ke nomor skill
* Skill\_Evo\_At: Evo skill dapat terjadi saat level player
* Strength: Skill kuat terhadap elemen nomor
* Weaknes: Skill lemah terhadap elemen nomor
* Description: Deskripsi dari skill
  + - * + Method
* getID(): getter ID
* getName(): getter Name
* getTarget(): getter Target
* getElemenID(): getter Elemen\_No
* getPower(): getter Power
* getPowerEffect(): getter Power\_Effect\_No
* getAccuracy(): getter Accuracy
* getStatusEffect(): getter Status\_Effect
* getProbStatEffect(): getter Prob\_Stat\_Effect
* getSkillEvoTo(): getter Skill\_Evo\_To
* getSkillEvoAt(): getter Skill\_Evo\_At
* getStrength(): getter Strength
* getWeaknes(): getter Weakness
* getDescription(): getter Description
* setName(String): setter Name
* setTarget(int): setter Target
* setElemenID(int): setter Elemen\_No
* setPower(int): setter Power
* setPowerEffect(int): setter Power\_Effect\_No
* setAccuracy(int): setter Accuracy
* setStatusEffect(int): setter Status\_Effect
* setProbStatEffect(int): setter Prob\_Stat\_Effect
* setSkillEvoTo(int): setter Skill\_Evo\_To
* setSkillEvoAt(int): setter Skill\_Evo\_At
* setStrength(int): setter Strength
* setWeaknes(int): setter Weakness
* setDescription(String): setter Description
* SkillTabel: abstrak kelas untuk membuat tabel skill dlm bentuk SQLite
* SkillCD: abstrak kelas mengisi tabel skill dengan SQLite
* Skill(int,SQLiteDatabase): konstruktor skill dengan masukan id skill dan tabel skill
* getData(SQLiteDatabase): Menampilkan semua isi tabel skill

# Program Utama

Program akan memulai *game* di rumah, dimana ada 2 pilihan menu, yaitu New dan Load. Menu New digunakan untuk membuat player baru, sedangkan menu Load digunakan untuk memakai data pemain terdahulu yang sudah tersimpan pada game. Untuk keluar dari game, pemain dapat menggunakan tombol *back* pada *device* masing-masing.

Permainan akan dimulai di dalam rumah (Home). Menu yang bisa dijalankan di Home antara lain menu Save dan Sleep. Menu Save digunakan untuk menyimpan data pemain, sementara menu Sleep digunakan agar pemain bisa ”tidur”. Saat pemain terbangun, kondisi lingkungan akan berlawanan dengan kondisi sebelum tidur, misalnya pemain tertidur saat siang, maka dia akan terbangun saat malam. Selain itu, kondisi semua monster juga akan dikembalikan seperti semula ketika pemain terbangun. Jika pemain ingin keluar, pemain dapat menggerakkan karakternya ke depan pintu rumah, dimana pemain akan berada di area baru bernama City

Seperti namanya, City merupakan kota kecil dimana tedapat 4 bangunan, yaitu Home, Shop, Combinatorium dan Stadium. Setiap bangunan memiliki fungsi berbeda. Shop digunakan untuk menjual-beli item, Combinatorium digunakan apabila pemain ingin menggabungkan 2 monster yang dia punya, dan Stadium digunakan untuk bertarung dengan pemain lain, dalam hal ini dijalankan oleh komputer

Shop memiliki 2 menu utama, yaitu menu Buy dan menu Sell. Menu Buy digunakan apabila pemain ingin membeli Item dari Shop, sedangkan menu Sell digunakan apabila pemain ingin menjual Item kepada Shop.

Combinatorium hanya memiliki 1 menu utama, yaitu menu combinatorium, dimana pemain dapat memilih 2 monster yang dia punya untuk digabungkan dan menghasilkan monster baru. Perlu diketahui tidak semua monster bisa digabungkan dengan monster lain. Ada beberapa monster tertentu yang tidak bisa digabungkan, sehingga Combinatorium tidak bisa menggabungkan monster tersebut.

Stadium digunakan untuk bertarung melawan pemain lain yang dikontrol oleh komputer. Untuk dapat bertarung di stadium, pemain harus membayar uang pendaftaran terlebih dahulu minimal 50.000 zeny. Apabila pemain menang, maka pemain akan mendapatkan uang sejumlah uang yang dibayar. Sebaliknya, apabila pemain kalah, maka pemain akan kehilangan uang sejumlah uang yang dibayar. Pertarungan di dalam stadium tidak akn mempengaruhi exp monster.

Selain City, di dalam game ini ada beberapa area lain yang tidak masuk ke dalam building, yaitu OuterArea dan Shore. OuterArea merupakan area yang menyerupai berupa hutan. OuterArea memiliki 10 level lantai, dimana di setiap lantai level monster yang ada di sana semakin bertambah. Pemain dapat dihalangi oleh lingkungan OuterArea dengan adanya Pohon dan Batu. Untuk mengatasinya, pemain dapat menggunakan skill Cut atau Push yang dimiliki oleh Monster yang ada pada Party mereka.

Pada Shore, pemain akan berada di area pantai, dimana pemain bisa berjalan seperti biasa di bagian pasir. Namun apabila pemain ingin menjelajahi laut, mereka harus menggunakan skill Swim yang dimiliki oleh Monster yang ada pada Party mereka. Pada City dan Shore, terdapat monster liar yang berkeliaran, apabila pemain bertemu dengan monster liar tersebut maka pemain akan memasuki mode Battle.

Pada Mode Battle, pemain dapat memilih untuk bertarung, menggunakan item, kabur, atau menangkap monster tersebut. Layar akan menunjukkan status kedua Monster dan HP yang dimiliki oleh kedua monster tersebut. Status monster dapat dipengaruhi oleh skill yang monster lawan gunakan dan item yang digunakan pada monster. Pemain juga dapat menangkap monster liar yang mereka lawan, namun tingkat keberhasilan untuk menangkap monster tersebut juga dipengaruhi oleh HP monster lawan. Semakin tinggi HP-nya, semakin susah untuk ditangkap. 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

Ada beberaa Item Tools yang dapat digunakan tergantung pada situasi yang dimiliki pemain. Misalnya, saat keadaan Night terjadi, maka layar akan menjadi gelap. Untuk mengatasinya, pemain dapat menggunakan Item Torch, agar layar menjadi terang kembali. Item Torch memiliki masa pakai, sehingga tidak bisa dipakai selamanya. Ada juga item Bicycle, yang berguna untuk menambah speed pemain di darat.

# Algoritma-Algoritma Menarik

## <Algoritma pegerakan monster>

* Fungsi update\_pos(int HeroIX, int HeroIY): disini pergerakan monster secara random sesuai posisi hero, dengan algoritma random
* decideMendekatMovement(int HeroIX, int HeroIY): disini pergerakan monster mendekati posisi hero, dengan algoritma dengan selisih koordinat x,y monster dan hero
* decideMenjauhMovement(int HeroIX, int HeroIY): disini pergerakan monster menjauhi posisi hero, dengan algoritma dengan selisih koordinat x,y monster dan hero

# Pembagian Kerja dalam Kelompok

* 13511025 Lubis Sucipto
  + - Membuat kelas skill, laporan, battle stadium
* 13511037 Muhammad Rizky W
  + - Membuat kelas player, player item, player monster, laporan,
* 13511051 Andreas Dwi Nugroho
  + - Membuat kelas item, dan combinatorium
* 13511071 Setyo Legowo
  + - Membuat kelas main, menggabungkan semua kelas, membuat animasi, membuat desain dan susunan permainan secara keseluruhan
* 13511083 Fawwaz Muhammad
  + - Membuat kelas area OuterArea dan Pantai, pergerakan monster dan player di area luar
* 13511093 Asep Saepudin
  + - Membuat kelas monster, diagram kelas

# Lampiran

## Notulen Rapat

|  |  |
| --- | --- |
| 30 Maret 2013 | Pembuatan grup dropbox tubes 2 OOP |
| 10 April 2013 | Rilis tubes dan kumpul awal |
| 14 April 2013 | Pembagian kelas dan penyelesaian kelas awal |
| 29 April 2013 | Nubes bareng dan penggabungan kelas |
| 30 April & 1 Mei 2013 | Asistensi |
| 3 Mei 2013 | Nubes bareng dan penyelesaian game |
| 9 Mei 2013 | Penyelesaian game bersama-sama |

## Log Activity Anggota Kelompok

|  |  |  |
| --- | --- | --- |
| Nama / NIM | Tanggal | Kegiatan |
| Lubis Sucipto | 13 April 2013 | * Membuat kelas awal skill |
| 22 April 2013 | * Menyelesaikan semua method-method kelas skill * Menambah atribut-atribut baru yang diperlukan seperti deskripsi skill |
| 2 Mei 2013 | * Membuat laporan * Membantu pembuatan battle stadium |
| 9 Mei 2013 | * Menyelesaikan laporan |
| M Rizky Widyayulianto | 13 April 2013 | * Membuat kelas awal player, player item, dan player monster |
| 26 April 2013 | * Menyelesaikan semua method-method di kelasnya |
| 29 April 2013 | * Membuat player party |
| 9 Mei 2013 | * Membantu menyelesaikan laporan * Membuat desain beberapa tampilan |
| Andreas Dwi Nugroho | 13 April 2013 | * Membuat kelas awal item |
| 19 April 2013 | * Menyelesaikan semua method dan membuat konstruktor player item dari SQLite database |
| 1 Mei 2013 | * Membuat kelas kombinatorium |
| 9 Mei 2013 | * Pemakaian monster ball |
| Setyo Legowo / 13511071 | 13 April 2013 | * Membuat tampilan muka dengan beberapa definisi awal kelas Main. |
| 14 April 2013 | * Mendefinisikan awal kelas Monster, Item, Player, Skill, Element, StatusEffect, PlayerMonster, dan PlayerItem dalam package model. * Mendefinisikan masing-masing kelas model sehingga dapat disimpan didalam mysql. |
| 16 April 2013 | * Membuat user interface untuk membuat player baru dan memilih player lama. |
| 5 Mei 2013 | * Membuat skenario dan desain laut, stadium, shop, efek siang malam, home serta controlernya |
| 9 Mei 2013 | * Battle, menyelesaikan semau kekurangan game yang belum tercapai dan membaguskan performansi |
| Fawwaz Muhammad / 13511083 | 18 April 2013 | * Membuat kelas awal area |
| 20 April 2013 | * Menyelesaikan semua method-method dan desain tampilan awal |
| 28 April 2013 | * Image OuterArea, Dan Tampilan dalam bangunan (shop, stadium) * Implementasi ke dalam bahasa java untuk tampilan outer Area * Menyambungkan outer area dengan city |
| 3 Mei 2013 | * Menyelesaikan pegerarakan monster |
| 9 Mei 2013 | * Menyelesaikan bagian laut, pemaikaian cut dan push |
| Asep Saepudin / 13511093 | 16 April 2013 | * Membuat kelas monster dan daftar monster |
| 18 April 2013 | * Menyelesaikan semua method-method kelas monster |
| 24 April 2013 | * Membuat desain monster-monster yang ada di pokeranch |
| 9 Mei 2013 | * Menyelesaikan desain monster dan diagram kelas |