Backend classes

Game.Creatures::Lutemon
«set» - idCounter: int
«get» # image: int
«get» # atk: int
«get» # def: int
«get/set» # experience: int
«get» # health: int
«get» # maxHealth: int
«get» # id: int
«get» # id: wins
«get» # id: losses
«get» # name: String
«get» # color: String
«get» # selected: boolean
+ isAlive(): boolean
+ attack(): int
+ defense(attacker: Lutemon)
+ resetAtk()
+ resetDef()
+ gainExp(amount: int)
+ heal()
+ toString(): String
+ addLoss()
+ addWin()
+ isSelected(): boolean
+ select(boolean)

Exception used while trying to get a Lutemonwith an ID, which isn'tin the dataset. Thrown in: Storage.getLutemon()

Game.Areas::BadIdException

Game.Areas::Storage
- serialVersionUID: long
name: String
«get» # lutemons: HashMap<Integer, Lutemon>
+ listLutemons()
+ getLutemon(id: int): Lutemon
+ getName(): String
+ sendToHome(id: int)
+ sendToBattlefield(id: int)
+ sendToTrain(id: int)
+ removeLutemon(lutemon: Lutemon)
+ checkIdExists(id: int): boolean
+ saveLutemon(context: Context, filename: String)
+ loadLutemon(context: Context, filename: String)
+ getHighestID(): int
+ deselectAll()
- sendTo(place: Storage, lutemon: Lutemon)
- addLutemon(l: Lutemon)

0..10 1

Game.Areas::Home
- storage: Home
+ getInstance(): Home
+ createLutemon(Lutemon)

Game.Areas::BattleField
- storage: BattleField
+ getInstance(): BattleField

Game.Areas::TrainingArea
- storage: TrainingArea
+ getInstance(): TrainingArea

Game.Creatures::White

Game.Creatures::White

Game.Creatures::White

Game.Creatures::White

Game.Creatures::White

Frontend classes

Game.Fragments::CreateLutemonFragment
- MAX_LUTEMONS: int
- lutemon_name: TextView
- rg: RadioGroup
- lutemon_count: TextView
- createLutemon()
- amountOfLutemons(): int
+ onCreate(savedInstanceState: Bundle)
+ onCreateView(LayoutInflater, ViewGroup, Bundle): View

Game.Fragments::GymFragment
- STORAGE: TrainingArea
- rv: RecyclerView
- rg: RadioGroup
- swap: boolean
- easy: Lutemon
- hard: Lutemon
+ onCreate(savedInstanceState: Bundle)
+ onCreateView(LayoutInflater, ViewGroup, Bundle): View
+ getCheckedLutemons(): int[0..10]
+ showChooseDifficulty()
+ getDifficulty(dialog: Dialog): int
+ showTrainingArea(difficulty: int)
+ fightTraining(Lutemon, Lutemon, Dialog)
+ endFightTraining(Lutemon, Lutemon)
+ roundOfFightTraining(Dialog, Lutemon, Lutemon, ImageView, ImageView, TextView): boolean

Game.Fragments::ArenaFragment
- STORAGE: BattleField
- rv: RecyclerView
- rg: RadioGroup
- swap: boolean
+ onCreate(savedInstanceState: Bundle)
+ onCreateView(LayoutInflater, ViewGroup, Bundle): View
+ getCheckedLutemons(): int[0..10]
+ showLutemonFight()
+ fight(Lutemon, Lutemon, Dialog)
+ endFight(Lutemon, Lutemon, boolean)
+ roundOfFight(Dialog, Lutemon, Lutemon, ImageView, ImageView, TextView): boolean

Game.Fragments::HomeFragment
- STORAGE: Home
- rv: RecyclerView
- rg: RadioGroup
+ onCreate(savedInstanceState: Bundle)
+ onCreateView(LayoutInflater, ViewGroup, Bundle): View
+ getCheckedLutemons(): int[0..10]

MainActivity
- decorView: View
- doubleBackToExitPressedOnce: boolean
- layoutButtons: LinearLayout
- getHighestID(): int
- loadData()
- hideSystemBars(): int
+ onBackPressed()
+ onWindowFocusChanged(hasFocus: boolean)
+ saveData()
+ sendTo()
+ switchToGym()
+ switchToArena()
+ switchToHome()
+ deselectAll()
+ removeLutemon(id: int)

Adapter used in fragments' RecyclerView-object.

ShowLutemonAdapter
- context: Context
- lutemons: HashMap<int, Lutemon>
- id_list: int[0..10]
+ ShowLutemonAdapter(Context, HashMap<Integer, Lutemon>)
+ onCreateViewHolder(ViewGroup, int): LutemonViewHolder
+ onBindViewHolder(LutemonViewHolder, int)
+ showLutemonInfo(pos: int, v: View)
+ getItemCount(): int

LutemonViewHolder
- lutemonImage: ImageView
- lutemonInfo: ImageView
- lutemonName: TextView
- lutemonColor: TextView
- lutemonWins: TextView
- lutemonLosses: TextView
- cb: CheckBox
+ LutemonViewHolder(View)