PALROGUE DOCUMENTATION

Crated By

6630180021 Pakornkiat Opaprakasit
6630237121 Pon Kittinaraporn

2110215 Programming Methodology
Semester 1 Year 2024
Chulalongkorn University

Pal Rogue

Introduction

Embark on an epic journey through a challenging, turn-based rogue-like game where every decision matters. In *Pal Rogue*, you'll face wave after wave of increasingly powerful opponents, battling your way through 50 waves to complete a thrilling run.

Choose your Pal carefully and develop unique strategies as you progress. Each Pal has its own abilities and potential to evolve, adding layers of depth to your gameplay. But be warned—some Pal cannot evolve, and it's up to you to decide how to use their strengths wisely.

With procedurally generated waves of battles and unpredictable encounters, no two runs are the same. Will you strategize your way to victory, or will the challenges of the rogue-like world prove too great?

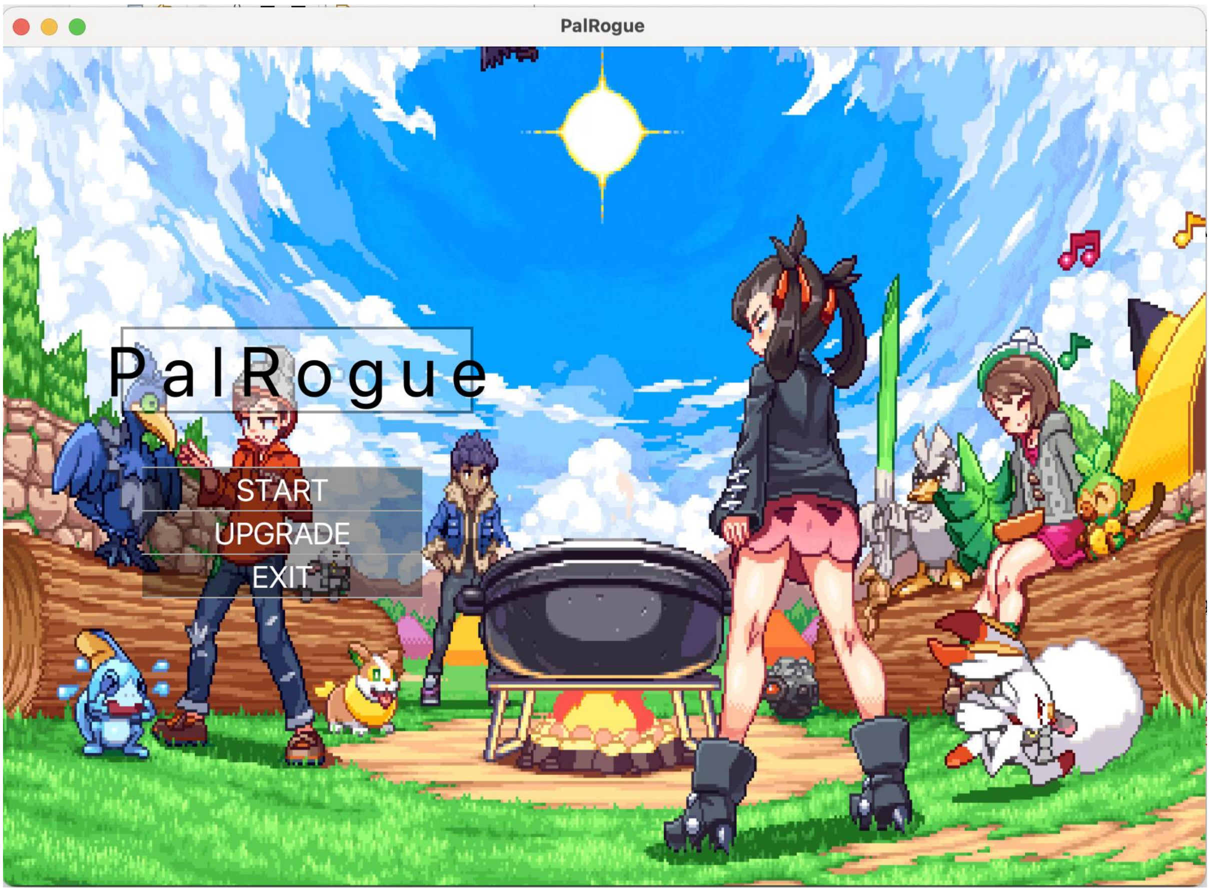
Get ready to test your skills, adapt to the unexpected, and become the ultimate Pal trainer in this one-of-a-kind adventure.

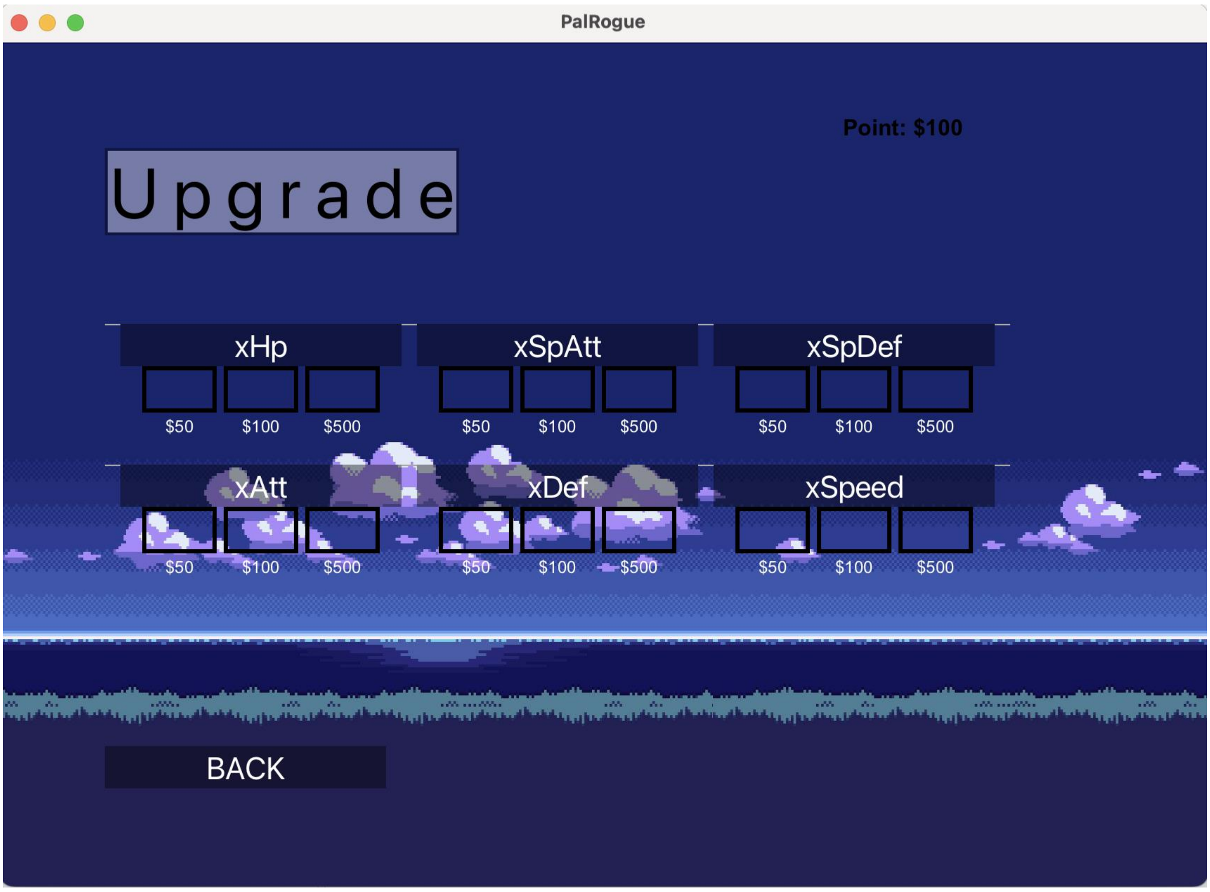
Are you prepared to take on the challenge and make it to the final wave? Let the journey begin!

Rule

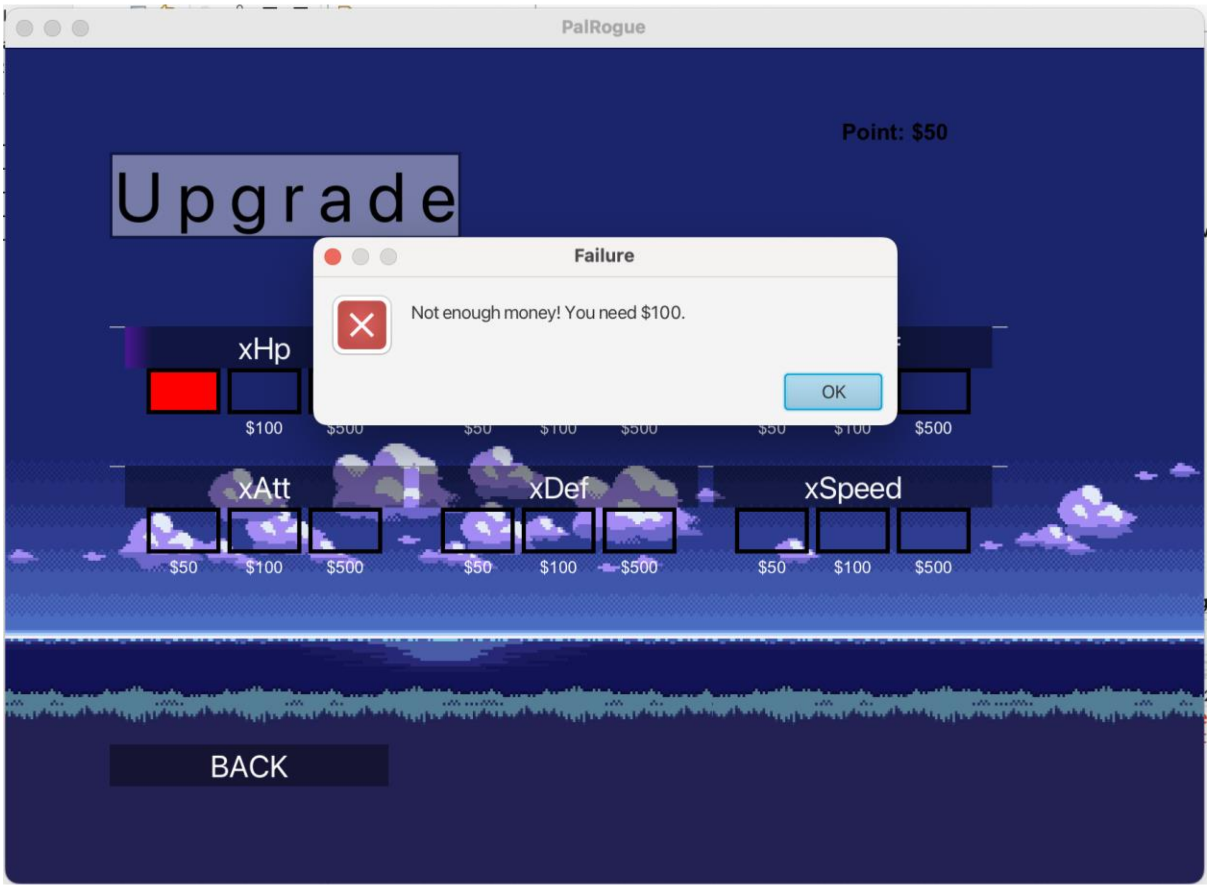
In *Pal Rogue*, your goal is to survive 50 waves of turn-based battles, defeating increasingly powerful enemy Pokémon while managing your team strategically. At the start of a run, you choose a team of Pokémon, each with unique abilities, stats, and the potential to evolve, though some cannot. During battles, you can attack, defend, use special abilities, or items to gain an advantage. Between waves, you can heal, level up, and earn rewards, but as the waves progress, the challenges become harder. Pokémon gain experience from battles, and those that can evolve will unlock stronger moves and stats, while fainted Pokémon cannot be used for the rest of the run unless revived with rare items. The game ends if all your Pokémon faint, but you win if you clear all 50 waves. Each wave and event is randomized, offering unique rewards and challenges, and completing waves unlocks new Pokémon or items for future runs.

Example

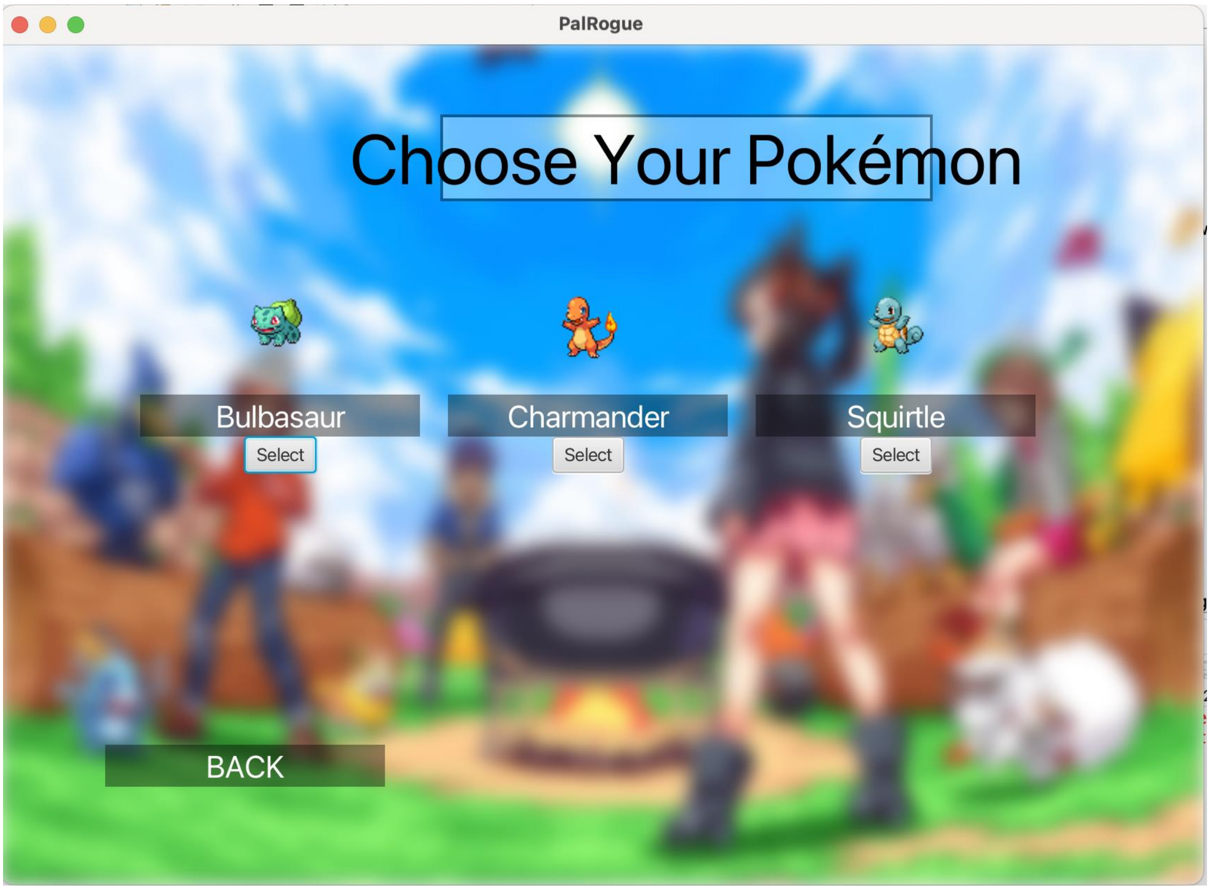

PalRogue
PalRogue
START
UPGRADE
EXIT

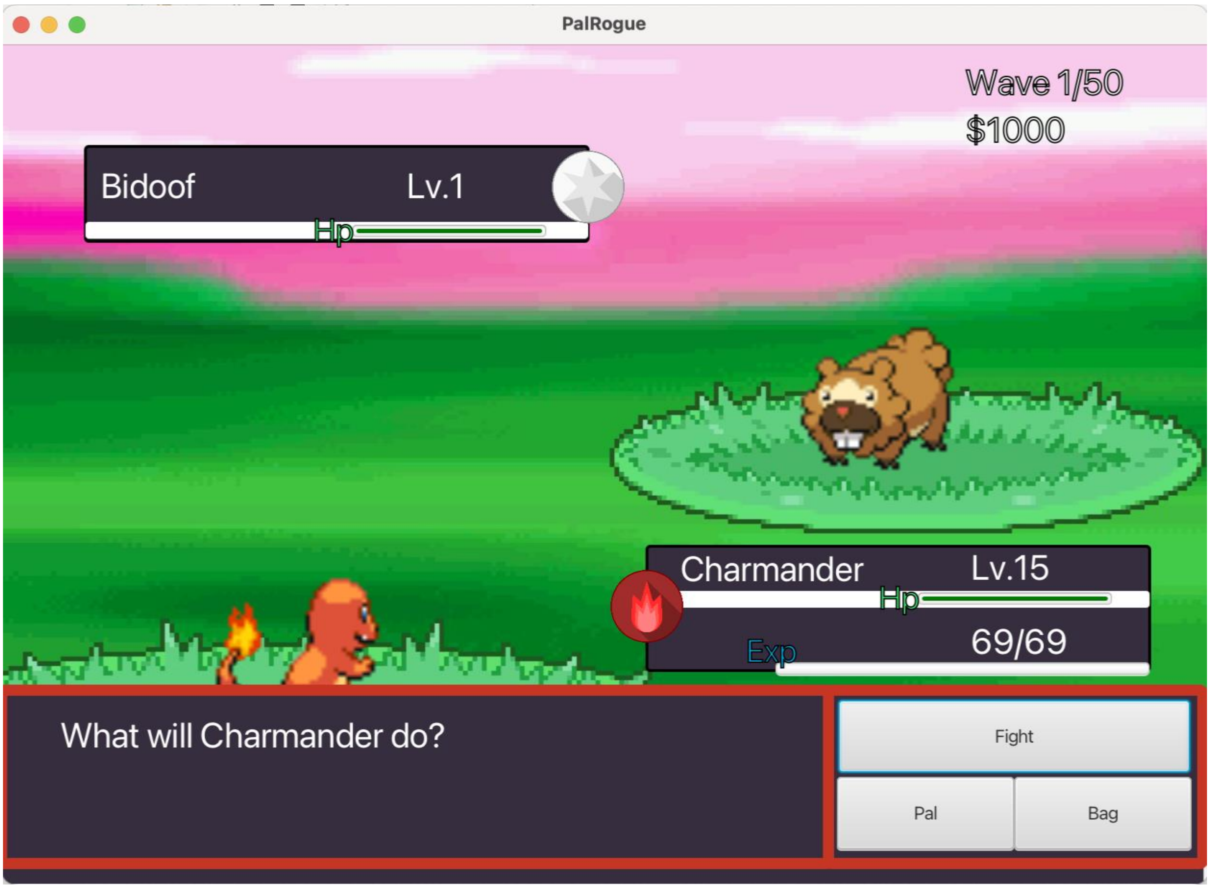

PalRogue
Upgrade
xHp
$50 $100 $500
xSpAtt
$50 $100 $500
xSpDef
$50 $100 $500
xAtt
$50 $100 $500
xDef
$50 $100 $500
xSpeed
$50 $100 $500
BACK

Point: $100

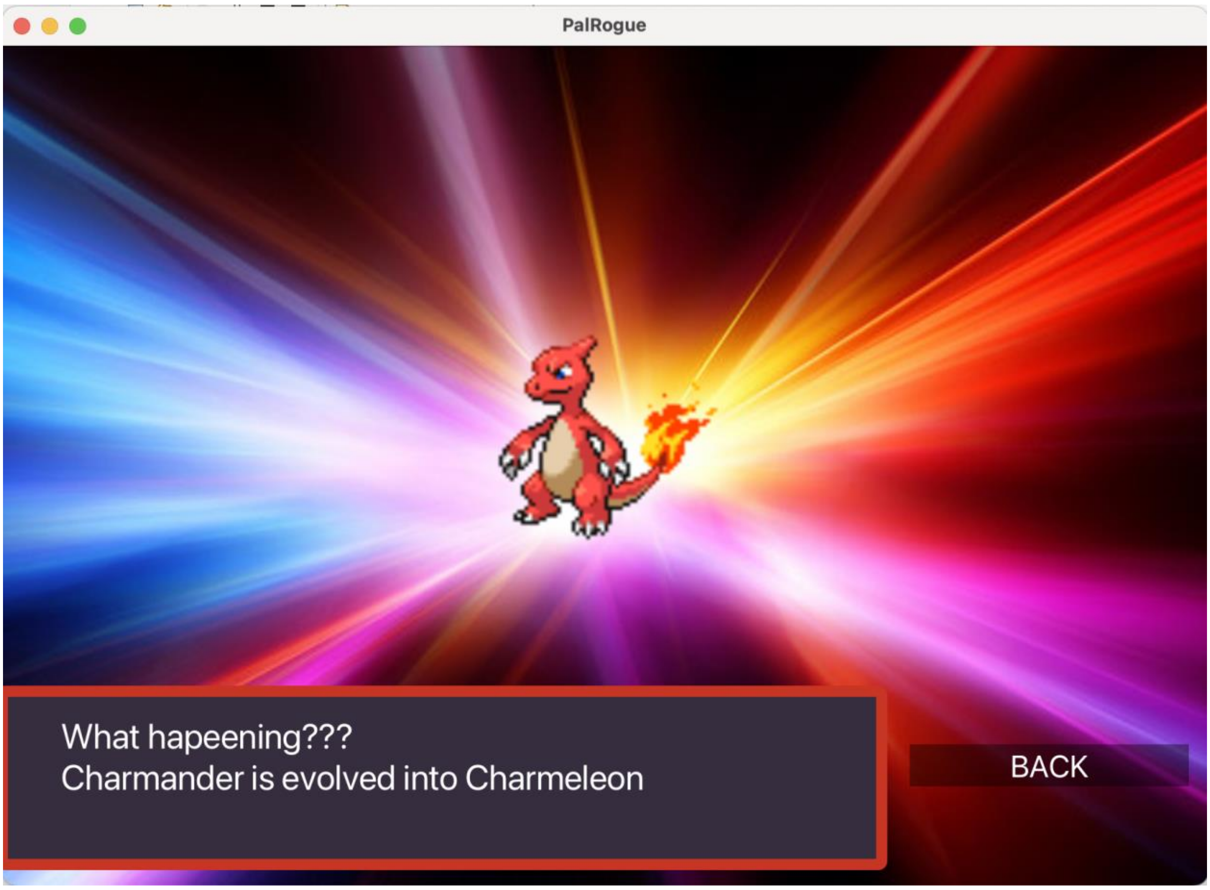


Not enough money! You need $100.

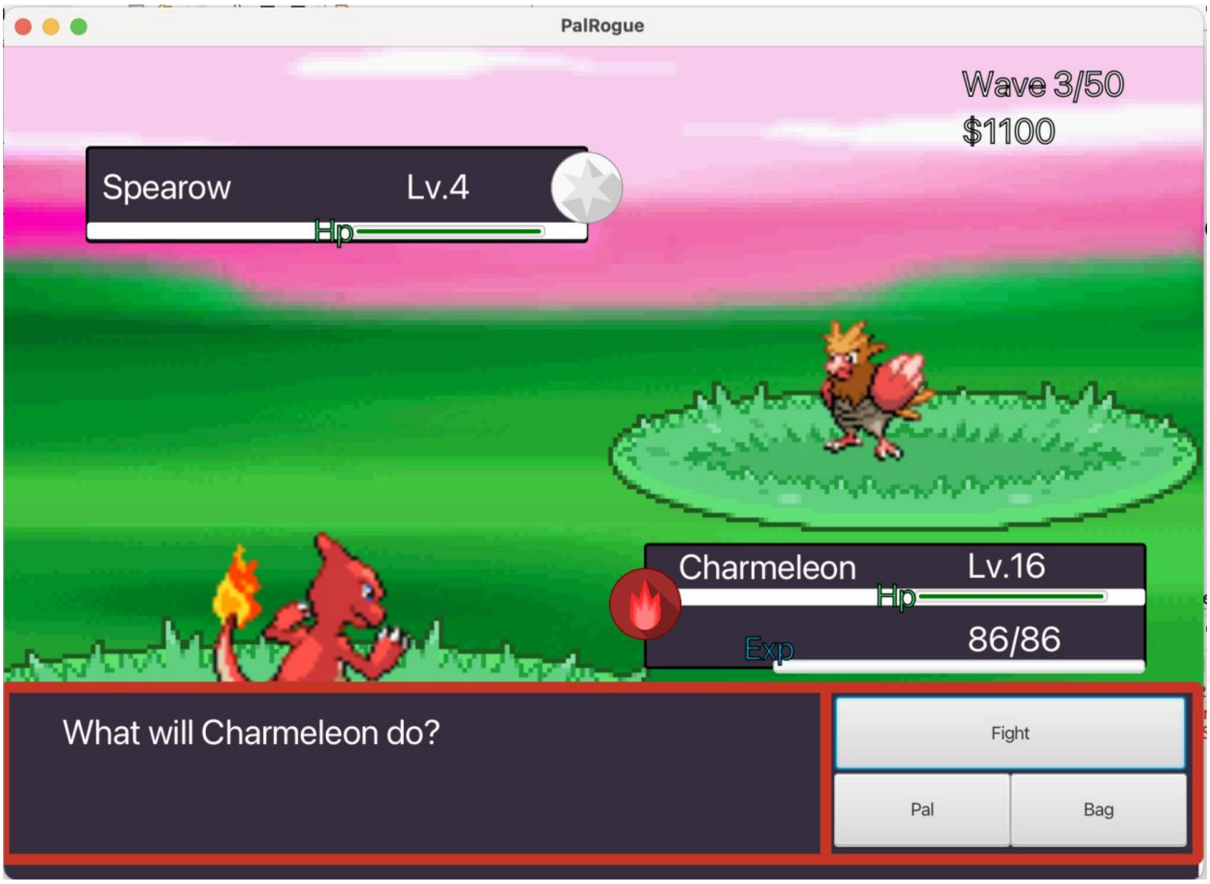




What will Charmander do?

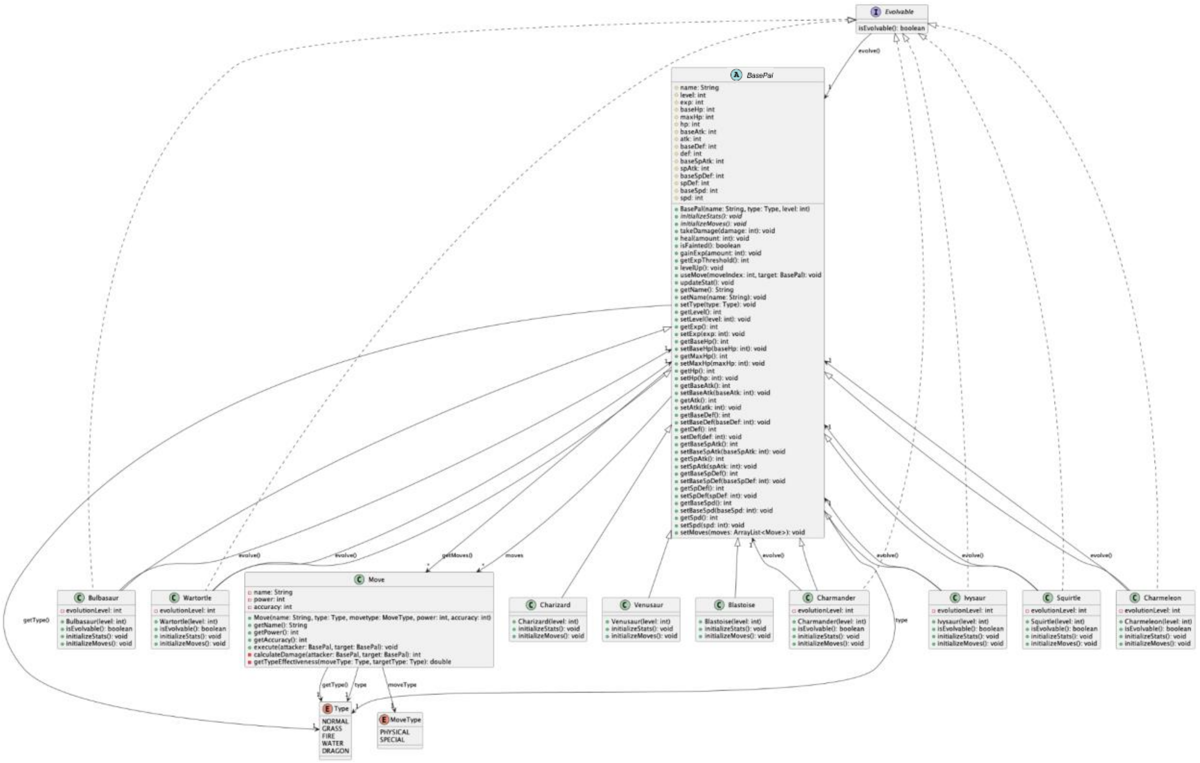


What hapeening???
Charmander is evolved into Charmeleon

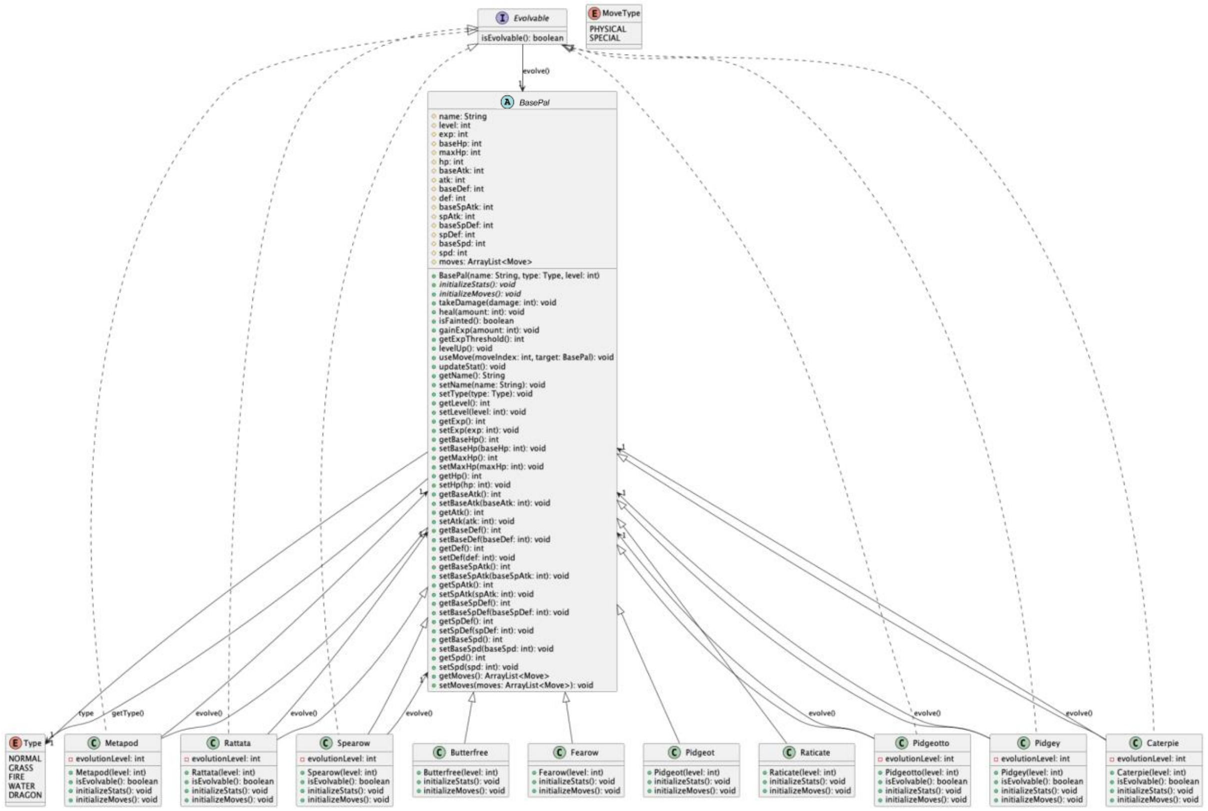


What will Charmeleon do?

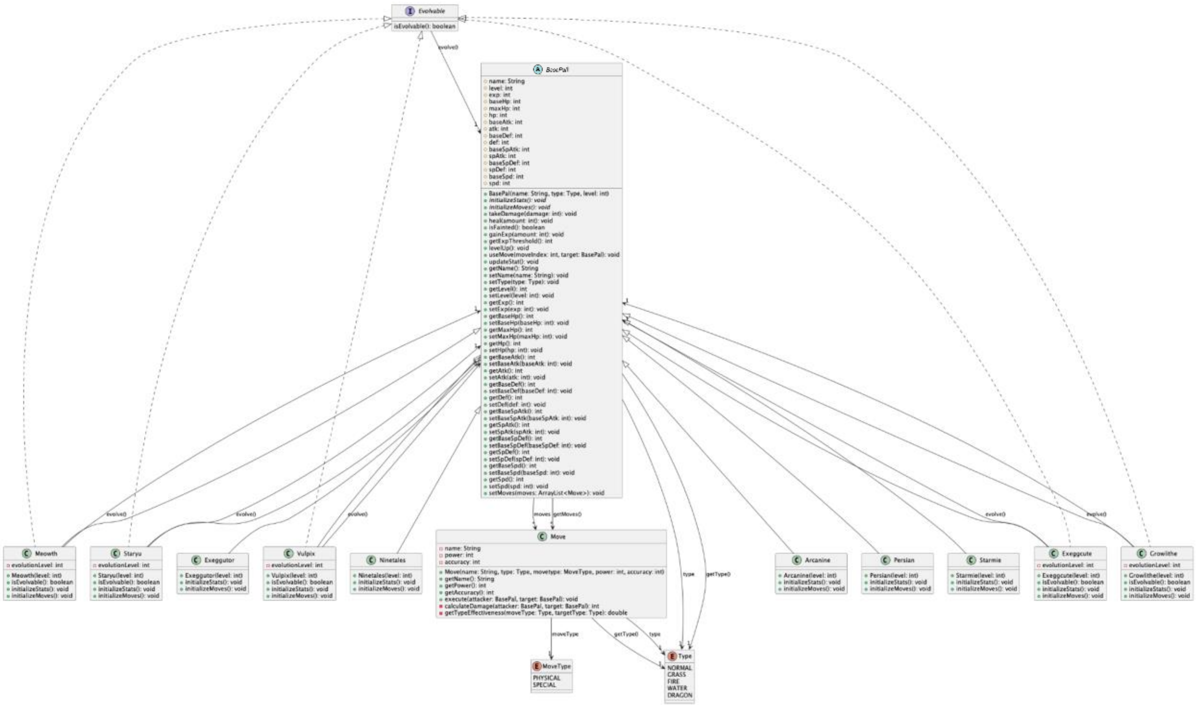
Object UML

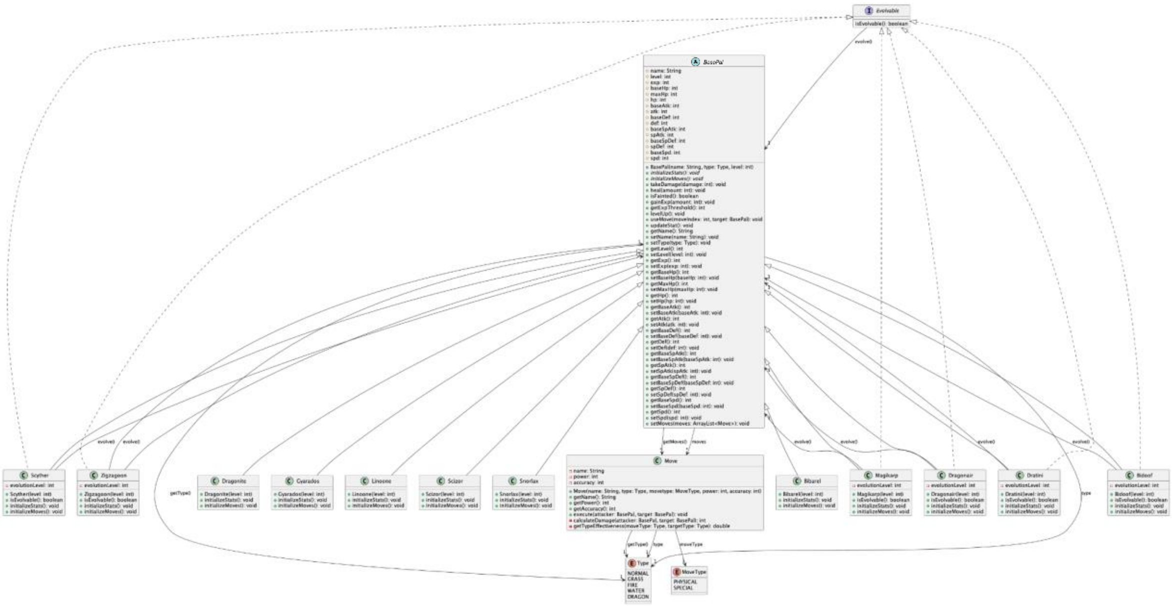


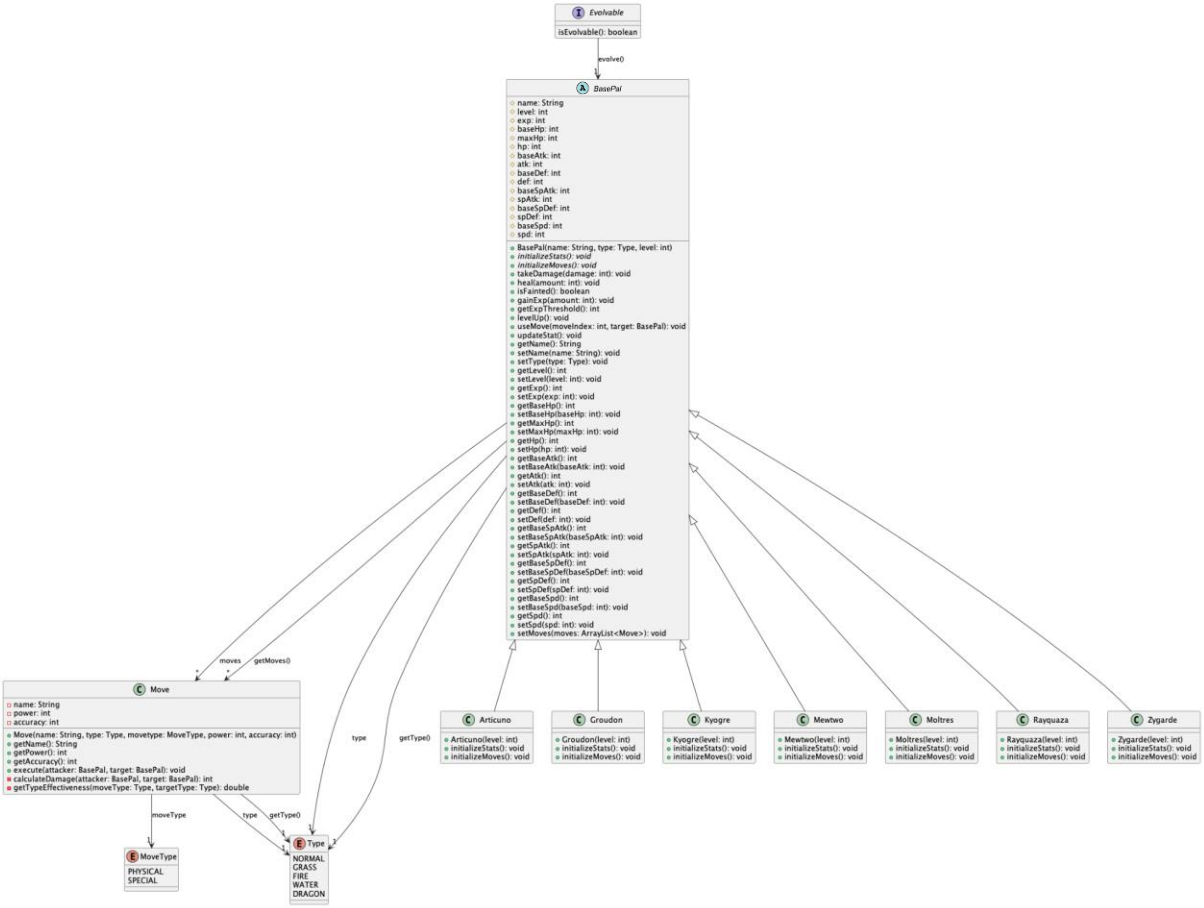
Evolvable
isEvolvable(): boolean
evolve()
1
BasePal
name: String
level: int
exp: int
baseHp: int
maxHp: int
hp: int
baseAtk: int
atk: int
baseDef: int
def: int
baseSpAtk: int
spAtk: int
baseSpDef: int
spDef: int
baseSpd: int
spd: int
BasePal(name: String, type: Type, level: int)
initializeStats(): void
initializeMoves(): void
takeDamage(damage: int): void
heal(amount: int): void
isFainted(): boolean
gainExp(amount: int): void
getExpThreshold(): int
levelUp(): void
useMove(moveIndex: int, target: BasePal): void
updateStat(): void
getName(): String
setName(name: String): void
setType(type: Type): void
getLevel(): int
setLevel(level: int): void
getExp(): int
setExp(exp: int): void
getBaseHp(): int
setBaseHp(baseHp: int): void
getMaxHp(): int
setMaxHp(maxHp: int): void
getHp(): int
setHp(hp: int): void
getBaseAtk(): int
setBaseAtk(baseAtk: int): void
getAtk(): int
setAtk(atk: int): void
getBaseDef(): int
setBaseDef(baseDef: int): void
getDef(): int
setDef(def: int): void
getBaseSpAtk(): int
setBaseSpAtk(baseSpAtk: int): void
getSpAtk(): int
setSpAtk(spAtk: int): void
getBaseSpDef(): int
setBaseSpDef(baseSpDef: int): void
getSpDef(): int
setSpDef(spDef: int): void
getBaseSpd(): int
setBaseSpd(baseSpd: int): void
getSpd(): int
setSpd(spd: int): void
setMoves(moves: ArrayList<Move>): void
evolve()
evolve()
getMoves()
moves
*
*
evolve()
evolve()
evolve()
evolve()
getType()
type
Bulbasaur
evolutionLevel: int
Bulbasaur(level: int)
isEvolvable(): boolean
initializeStats(): void
initializeMoves(): void
Wartortle
evolutionLevel: int
Wartortle(level: int)
isEvolvable(): boolean
initializeStats(): void
initializeMoves(): void
Move
name: String
power: int
accuracy: int
Move(name: String, type: Type, movetype: MoveType, power: int, accuracy: int)
getName(): String
getPower(): int
getAccuracy(): int
execute(attacker: BasePal, target: BasePal): void
calculateDamage(attacker: BasePal, target: BasePal): int
getTypeEffectiveness(moveType: Type, targetType: Type): double
Charizard
Charizard(level: int)
initializeStats(): void
initializeMoves(): void
Venusaur
Venusaur(level: int)
initializeStats(): void
initializeMoves(): void
Blastoise
Blastoise(level: int)
initializeStats(): void
initializeMoves(): void
Charmander
evolutionLevel: int
Charmander(level: int)
isEvolvable(): boolean
initializeStats(): void
initializeMoves(): void
Ivysaur
evolutionLevel: int
Ivysaur(level: int)
isEvolvable(): boolean
initializeStats(): void
initializeMoves(): void
Squirtle
evolutionLevel: int
Squirtle(level: int)
isEvolvable(): boolean
initializeStats(): void
initializeMoves(): void
Charmeleon
evolutionLevel: int
Charmeleon(level: int)
isEvolvable(): boolean
initializeStats(): void
initializeMoves(): void
getType()
type
moveType
1
Type
NORMAL
GRASS
FIRE
WATER
DRAGON
MoveType
PHYSICAL
SPECIAL

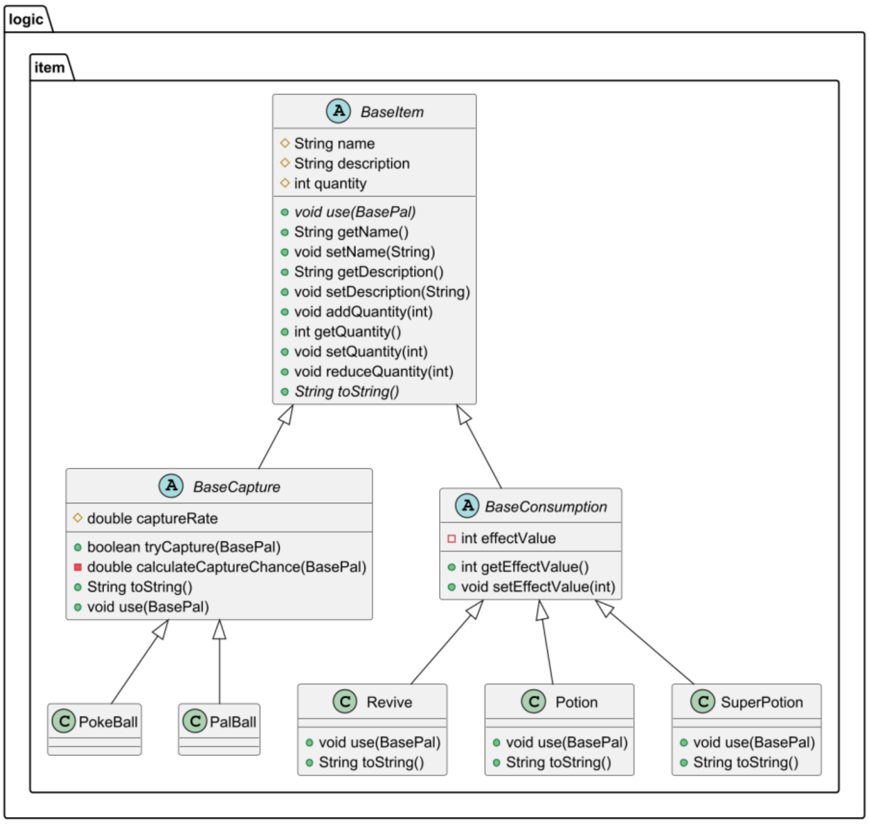


Evolvable
isEvolvable(): boolean
MoveType
PHYSICAL
SPECIAL
evolve()
1
BasePal
name: String
level: int
exp: int
baseHp: int
maxHp: int
hp: int
baseAtk: int
atk: int
baseDef: int
def: int
baseSpAtk: int
spAtk: int
baseSpDef: int
spDef: int
baseSpd: int
spd: int
moves: ArrayList<Move>
BasePal(name: String, type: Type, level: int)
initializeStats(): void
initializeMoves(): void
takeDamage(damage: int): void
heal(amount: int): void
isFainted(): boolean
gainExp(amount: int): void
getExpThreshold(): int
levelUp(): void
useMove(moveIndex: int, target: BasePal): void
updateStat(): void
getName(): String
setName(name: String): void
setType(type: Type): void
getLevel(): int
setLevel(level: int): void
getExp(): int
setExp(exp: int): void
getBaseHp(): int
setBaseHp(baseHp: int): void
getMaxHp(): int
setMaxHp(maxHp: int): void
getHp(): int
setHp(hp: int): void
getBaseAtk(): int
setBaseAtk(baseAtk: int): void
getAtk(): int
setAtk(atk: int): void
getBaseDef(): int
setBaseDef(baseDef: int): void
getDef(): int
setDef(def: int): void
getBaseSpAtk(): int
setBaseSpAtk(baseSpAtk: int): void
getSpAtk(): int
setSpAtk(spAtk: int): void
getBaseSpDef(): int
setBaseSpDef(baseSpDef: int): void
getSpDef(): int
setSpDef(spDef: int): void
getBaseSpd(): int
setBaseSpd(baseSpd: int): void
getSpd(): int
setSpd(spd: int): void
getMoves(): ArrayList<Move>
setMoves(moves: ArrayList<Move>): void
type
getType()
evolve()
evolve()
evolve()
evolve()
evolve()
evolve()
Type
NORMAL
GRASS
FIRE
WATER
DRAGON
Metapod
evolutionLevel: int
Metapod(level: int)
isEvolvable(): boolean
initializeStats(): void
initializeMoves(): void
Rattata
evolutionLevel: int
Rattata(level: int)
isEvolvable(): boolean
initializeStats(): void
initializeMoves(): void
Spearow
evolutionLevel: int
Spearow(level: int)
isEvolvable(): boolean
initializeStats(): void
initializeMoves(): void
Butterfree
Butterfree(level: int)
initializeStats(): void
initializeMoves(): void
Fearow
Fearow(level: int)
initializeStats(): void
initializeMoves(): void
Pidgeot
Pidgeot(level: int)
initializeStats(): void
initializeMoves(): void
Raticate
Raticate(level: int)
initializeStats(): void
initializeMoves(): void
Pidgeotto
evolutionLevel: int
Pidgeotto(level: int)
isEvolvable(): boolean
initializeStats(): void
initializeMoves(): void
Pidgey
evolutionLevel: int
Pidgey(level: int)
isEvolvable(): boolean
initializeStats(): void
initializeMoves(): void
Caterpie
evolutionLevel: int
Caterpie(level: int)
isEvolvable(): boolean
initializeStats(): void
initializeMoves(): void


Evolvable
isEvolvable(): boolean
evolve()
BasePal
name: String
level: int
exp: int
baseHp: int
maxHp: int
hp: int
baseAtk: int
atk: int
baseDef: int
def: int
baseSpAtk: int
spAtk: int
baseSpDef: int
spDef: int
baseSpd: int
spd: int
BasePal(name: String, type: Type, level: int)
initializeStats(): void
initializeMoves(): void
takeDamage(damage: int): void
heal(amount: int): void
isFainted(): boolean
gainExp(amount: int): void
getExpThreshold(): int
levelUp(): void
useMove(moveIndex: int, target: BasePal): void
updateStat(): void
getName(): String
setName(name: String): void
setType(type: Type): void
getLevel(): int
setLevel(level: int): void
getExp(): int
setExp(exp: int): void
getBaseHp(): int
setBaseHp(baseHp: int): void
getMaxHp(): int
setMaxHp(maxHp: int): void
getHp(): int
setHp(hp: int): void
getBaseAtk(): int
setBaseAtk(baseAtk: int): void
getAtk(): int
setAtk(atk: int): void
getBaseDef(): int
setBaseDef(baseDef: int): void
getDef(): int
setDef(def: int): void
getBaseSpAtk(): int
setBaseSpAtk(baseSpAtk: int): void
getSpAtk(): int
setSpAtk(spAtk: int): void
getBaseSpDef(): int
setBaseSpDef(baseSpDef: int): void
getSpDef(): int
setSpDef(spDef: int): void
getBaseSpd(): int
setBaseSpd(baseSpd: int): void
getSpd(): int
setSpd(spd: int): void
setMoves(moves: ArrayList<Move>): void
evolve()
evolve()
evolve()
moves
getMoves()
evolve()
evolve()
Meowth
evolutionLevel: int
Meowth(level: int)
isEvolvable(): boolean
initializeStats(): void
initializeMoves(): void
Staryu
evolutionLevel: int
Staryu(level: int)
isEvolvable(): boolean
initializeStats(): void
initializeMoves(): void
Exeggutor
Exeggutor(level: int)
initializeStats(): void
initializeMoves(): void
Vulpix
evolutionLevel: int
Vulpix(level: int)
isEvolvable(): boolean
initializeStats(): void
initializeMoves(): void
Ninetales
Ninetales(level: int)
initializeStats(): void
initializeMoves(): void
Move
name: String
power: int
accuracy: int
Move(name: String, type: Type, movetype: MoveType, power: int, accuracy: int)
getName(): String
getPower(): int
getAccuracy(): int
execute(attacker: BasePal, target: BasePal): void
calculateDamage(attacker: BasePal, target: BasePal): int
getTypeEffectiveness(moveType: Type, targetType: Type): double
type
getType()
Arcanine
Arcanine(level: int)
initializeStats(): void
initializeMoves(): void
Persian
Persian(level: int)
initializeStats(): void
initializeMoves(): void
Starmie
Starmie(level: int)
initializeStats(): void
initializeMoves(): void
Exeggcute
evolutionLevel: int
Exeggcute(level: int)
isEvolvable(): boolean
initializeStats(): void
initializeMoves(): void
Growlithe
evolutionLevel: int
Growlithe(level: int)
isEvolvable(): boolean
initializeStats(): void
initializeMoves(): void
moveType
getType()
type
MoveType
PHYSICAL
SPECIAL
Type
NORMAL
GRASS
FIRE
WATER
DRAGON


Evolvable
isEvolvable(): boolean
evolve()
BasePal
name: String
level: int
exp: int
baseHp: int
maxHp: int
hp: int
baseAtk: int
atk: int
baseDef: int
def: int
baseSpAtk: int
spAtk: int
baseSpDef: int
spDef: int
baseSpd: int
spd: int
BasePal(name: String, type: Type, level: int)
initializeStats(): void
initializeMoves(): void
takeDamage(damage: int): void
heal(amount: int): void
isFainted(): boolean
gainExp(amount: int): void
getExpThreshold(): int
levelUp(): void
useMove(moveIndex: int, target: BasePal): void
updateStat(): void
getName(): String
setName(name: String): void
setType(type: Type): void
getLevel(): int
setLevel(level: int): void
getExp(): int
setExp(exp: int): void
getBaseHp(): int
setBaseHp(baseHp: int): void
getMaxHp(): int
setMaxHp(maxHp: int): void
getHp(): int
setHp(hp: int): void
getBaseAtk(): int
setBaseAtk(baseAtk: int): void
getAtk(): int
setAtk(atk: int): void
getBaseDef(): int
setBaseDef(baseDef: int): void
getDef(): int
setDef(def: int): void
getBaseSpAtk(): int
setBaseSpAtk(baseSpAtk: int): void
getSpAtk(): int
setSpAtk(spAtk: int): void
getBaseSpDef(): int
setBaseSpDef(baseSpDef: int): void
getSpDef(): int
setSpDef(spDef: int): void
getBaseSpd(): int
setBaseSpd(baseSpd: int): void
getSpd(): int
setSpd(spd: int): void
setMoves(moves: ArrayList<Move>): void
evolve()
evolve()
getMoves()
moves
evolve()
evolve()
evolve()
evolve()
getType()
Scyther
evolutionLevel: int
Scyther(level: int)
isEvolvable(): boolean
initializeStats(): void
initializeMoves(): void
Zigzagoon
evolutionLevel: int
Zigzagoon(level: int)
isEvolvable(): boolean
initializeStats(): void
initializeMoves(): void
Dragonite
Dragonite(level: int)
initializeStats(): void
initializeMoves(): void
Gyarados
Gyarados(level: int)
initializeStats(): void
initializeMoves(): void
Linoone
Linoone(level: int)
initializeStats(): void
initializeMoves(): void
Scizor
Scizor(level: int)
initializeStats(): void
initializeMoves(): void
Snorlax
Snorlax(level: int)
initializeStats(): void
initializeMoves(): void
Move
name: String
power: int
accuracy: int
Move(name: String, type: Type, movetype: MoveType, power: int, accuracy: int)
getName(): String
getPower(): int
getAccuracy(): int
execute(attacker: BasePal, target: BasePal): void
calculateDamage(attacker: BasePal, target: BasePal): int
getTypeEffectiveness(moveType: Type, targetType: Type): double
Bibarel
Bibarel(level: int)
initializeStats(): void
initializeMoves(): void
Magikarp
evolutionLevel: int
Magikarp(level: int)
isEvolvable(): boolean
initializeStats(): void
initializeMoves(): void
Dragonair
evolutionLevel: int
Dragonair(level: int)
isEvolvable(): boolean
initializeStats(): void
initializeMoves(): void
Dratini
evolutionLevel: int
Dratini(level: int)
isEvolvable(): boolean
initializeStats(): void
initializeMoves(): void
type
Bidoof
evolutionLevel: int
Bidoof(level: int)
isEvolvable(): boolean
initializeStats(): void
initializeMoves(): void
getType()
type
moveType
Type
NORMAL
GRASS
FIRE
WATER
DRAGON
MoveType
PHYSICAL
SPECIAL


Evolvable
isEvolvable(): boolean
evolve()
1
BasePal
name: String
level: int
exp: int
baseHp: int
maxHp: int
hp: int
baseAtk: int
atk: int
baseDef: int
def: int
baseSpAtk: int
spAtk: int
baseSpDef: int
spDef: int
baseSpd: int
spd: int
BasePal(name: String, type: Type, level: int)
initializeStats(): void
initializeMoves(): void
takeDamage(damage: int): void
heal(amount: int): void
isFainted(): boolean
gainExp(amount: int): void
getExpThreshold(): int
levelUp(): void
useMove(moveIndex: int, target: BasePal): void
updateStat(): void
getName(): String
setName(name: String): void
setType(type: Type): void
getLevel(): int
setLevel(level: int): void
getExp(): int
setExp(exp: int): void
getBaseHp(): int
setBaseHp(baseHp: int): void
getMaxHp(): int
setMaxHp(maxHp: int): void
getHp(): int
setHp(hp: int): void
getBaseAtk(): int
setBaseAtk(baseAtk: int): void
getAtk(): int
setAtk(atk: int): void
getBaseDef(): int
setBaseDef(baseDef: int): void
getDef(): int
setDef(def: int): void
getBaseSpAtk(): int
setBaseSpAtk(baseSpAtk: int): void
getSpAtk(): int
setSpAtk(spAtk: int): void
getBaseSpDef(): int
setBaseSpDef(baseSpDef: int): void
getSpDef(): int
setSpDef(spDef: int): void
getBaseSpd(): int
setBaseSpd(baseSpd: int): void
getSpd(): int
setSpd(spd: int): void
setMoves(moves: ArrayList<Move>): void
moves
getMoves()
*
*
type
getType()
Move
name: String
power: int
accuracy: int
Move(name: String, type: Type, movetype: MoveType, power: int, accuracy: int)
getName(): String
getPower(): int
getAccuracy(): int
execute(attacker: BasePal, target: BasePal): void
calculateDamage(attacker: BasePal, target: BasePal): int
getTypeEffectiveness(moveType: Type, targetType: Type): double
moveType
type
getType()
1
1
1
1
1
MoveType
PHYSICAL
SPECIAL
Type
NORMAL
GRASS
FIRE
WATER
DRAGON
Articuno
Articuno(level: int)
initializeStats(): void
initializeMoves(): void
Groudon
Groudon(level: int)
initializeStats(): void
initializeMoves(): void
Kyogre
Kyogre(level: int)
initializeStats(): void
initializeMoves(): void
Mewtwo
Mewtwo(level: int)
initializeStats(): void
initializeMoves(): void
Moltres
Moltres(level: int)
initializeStats(): void
initializeMoves(): void
Rayquaza
Rayquaza(level: int)
initializeStats(): void
initializeMoves(): void
Zygarde
Zygarde(level: int)
initializeStats(): void
initializeMoves(): void


logic
item
BaseItem
String name
String description
int quantity
void use(BasePal)
String getName()
void setName(String)
String getDescription()
void setDescription(String)
void addQuantity(int)
int getQuantity()
void setQuantity(int)
void reduceQuantity(int)
String toString()
BaseCapture
double captureRate
boolean tryCapture(BasePal)
double calculateCaptureChance(BasePal)
String toString()
void use(BasePal)
BaseConsumption
int effectValue
int getEffectValue()
void setEffectValue(int)
PokeBall
PalBall
Revive
void use(BasePal)
String toString()
Potion
void use(BasePal)
String toString()
SuperPotion
void use(BasePal)
String toString()

1. Package item

1.1 Class BaseItem (Abstract)

Field

Name	Description
- String name	Name of the item.
- String description	Description of the item.
- int quantity	Quantity of the item.

Constructor

Name	Description
+ BaseItem(String name, String description)	Initialize an item with its name and description.
+ BaseItem(String name, String description, int quantity)	Initialize an item with its name, description and quantity.

Method

Name	Description
+ void use(BasePal target)	Use this item on target Pal.
+ void addQuantity(int amount)	Add the amount of this item.
+ void reduceQuantity(int amount)	Reduce the amount of this item.
+ String getName()	Return this item’s name.
+ void setName(String name)	Set name to this item’s name
+ String getDescription()	Return this item’s description.
+ void setDescription(String description)	Set name to this item’s description
+ int getQuantity()	Return quantity of this item.
+ void setQuantity(int amount)	Set the quantity of this item equal to the amount.

+ boolean equal(BaseItem otherItem)	Return true if each Field of this object equal to otherItem
+ String toString()	

1.2 Class BaseCapture (extends BaseItem)

Field

Name	Description
- int capturRate	Capture Rate of this item.

Constructor

Name	Description
+ BaseCapture(String name, String description, int captureRate, int quantity)	Initialize an item with its name, description, captureRate and quantity.

Method

Name	Description
+ boolean tryCapture(BasePal target)	Call calculateCaptureChance(target) If calculateCaptureChance(target) > random value Pal was captured. If Pal was captured add this Pal to instance.pals.
+ double calculateCaptureChance(BasePal target)	Calculate chance with HealthFactor
+ void use(BasePal target)	Call tryCapture(target)
+ int getCaptureRate()	Return this captureRate.
+ void setCaptureRate(int captureRate)	Set captureRate to this captureate.

1.3 Class Abstract BaseConsumption (extends BaseItem)

Field

Name	Description
- int effectValue	Effect effective percentage when use this item

Constructor

Name	Description
+ BaseConsumption(String name, String description, int quantity, int effectValue)	Initialize an item with its name, description, quantity and effectValue.

Method

Name	Description
+ int getEffectValue()	Return effectValue.
+ void setEffectValue(int effectValue)	Set effectValue to this.effectValue .

1.4 Class PalBall (extends BaseCapture)

Constructor

Name	Description
+ PalBall(int quantity)	Call super("Palball", "Lowest rate for capture Pal.", 1, quantity);

1.5 Class GreatBall (extends BaseCapture)

Constructor

Name	Description
+ GreatBall(int quantity)	Call super("GreatBall", "Highest rate for capture Pal.", 2, quantity);

1.6 Class Potion (extends BaseConsumption)

Constructor

Name	Description
+ Potion(int quantity)	Call super("Potion", "Restore some of your Pal's health", quantity, 25);

Method

Name	Description
+ void use(BasePal target)	Check if target is fainted - do nothing if target still alive Call target.heal(target.MaxHp x effectiveValue/100)

1.7 Class SuperPotion (extends BaseConsumption)

Constructor

Name	Description
+ SuperPotion(int quantity)	Call super("Super Potion", "Restores a significant amount of your Pal's health.", quantity, 50);

Method

Name	Description
+ void use(BasePal target)	Check if target is fainted - do nothing if target still alive Call target.heal(target.MaxHp x effectiveValue/100)

1.8 Class Revive (extends BaseConsumption)

Constructor

Name	Description
+ Revive(int quantity)	Call super("Revive", "Revive fainted Pal and restore some health", quantity, 50);

Method

Name	Description
+ void use(BasePal target)	Check if target is fainted Call target.heal(target.MaxHp x effectiveValue/100)

2. Package move

2.1 Class Move

Field

Name	Description
- String name	Name of this move.
- Type type	Type of this move.
- MoveType moveType	MoveType of this move.
- int power	Power of this move.
- int accuracy	Accuracy of this move.

Constructor

Name	Description
+ Move(String name, Type type, MoveType movetype, int power, int accuracy)	Initialize an move with its name, type, movetype, power and accuracy.

Method

Name	Description
+ void execute(BasePal attacker, BasePal target)	- Check accuracy - Call calculateDamage() - attacker dealt damage to target.
+ int calculateDamage(BasePal attacker, BasePal target)	Return how much damage attacker Pal dealt.
+ int getTypeEffectiveness(Type moveType, Type targetType)	Return TypeEffectiveness
+ String getName()	Return this.name .
+ void setName(String name)	Set name to this.name .
+ Type getType()	Return this.type .
+ void setType(Type type)	Set type to this.type .
+ int getPower()	Return this.power .

+ void setPower(int power)	Set power to this.power .
+ int getAccuracy()	Return this.accuracy .
+ void setAccuracy(int accuracy)	Set accuracy to this.accuracy .

3. Package pal

3.1 Abstract Class BasePal

Field

Name	Description
- String name	Name of this Pal.
- Type type	Type of this Pal.
- int level	Level of this Pal.
- int baseHp	BaseHP of this Pal.
- int maxHp	MaxHP of this Pal.
- int hp	HP of this Pal.
- int baseAtk	BaseAtk of this Pal.
- int atk	Atk of this Pal.
- int baseDef	BaseDef of this Pal.
- int def	Def of this Pal.
- int baseSpAtk	BaseSpAtk of this Pal.
- int spAtk	SpAtk of this Pal.
- int baseSpDef	BaseSpDef of this Pal.
- int spDef	SpDef of this Pal.
- int baseSpd	BaseSpeed of this Pal.
- int spd	Speed of this Pal.
- ArrayList<Move> moves	This Pal moves.

Constructor

Name	Description

+ BasePal(String name, Type type, int level)	Initialize Pal with its name, type and level.

Method

Name	Description
+ void initailizaStats()	Set all stats.
+ void initailizaMoves()	Set moves.
+ void takeDamage(int damage)	Decrease this.hp by damage.
+ void heal(int amount)	Increase this.hp by amount.
+ boolean isFainted()	Return this.HP <= 0
+ void levelUp()	This.level+=1
+ void useMove(int moveIndex, BasePal target)	Call use moves[moveIndex] on target Pal.

3.2 Class Bulbasaur (extends BasePal implements Evolvable)

Field

Name	Description
- EVOLUTION_LEVEL	

Constructor

Name	Description
+ Bulbasaur(int level)	Create new Pal at level = level

Method

Name	Description
+ boolean isEvolvable()	
+ BasePal evolve()	
+ void initailizaStats()	
+ void initailizaMoves()	

3.3 Class Ivysaur (extends BasePal implements Evolvable)

Field

Name	Description
- EVOLUTION_LEVEL	

Constructor

Name	Description
+ Ivysaur(int level)	Create new Pal at level = level

Method

Name	Description
+ boolean isEvolvable()	
+ BasePal evolve()	
+ void initailizaStats()	
+ void initailizaMoves()	

3.4 Class Venusaur (extends BasePal)

Constructor

Name	Description
+ Venusaur(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.5 Class Charmander (extends BasePal implements Evolvable)

Field

Name	Description
- EVOLUTION_LEVEL	

Constructor

Name	Description
+ Charmander(int level)	Create new Pal at level = level

Method

Name	Description
+ boolean isEvolvable()	
+ BasePal evolve()	
+ void initailizaStats()	
+ void initailizaMoves()	

3.6 Class Charmeleon (extends BasePal implements Evolvable)

Field

Name	Description
- EVOLUTION_LEVEL	

Constructor

Name	Description
+ Charmeleon(int level)	Create new Pal at level = level

Method

Name	Description
+ boolean isEvolvable()	
+ BasePal evolve()	
+ void initailizaStats()	
+ void initailizaMoves()	

3.7 Class Charizard (extends BasePal)

Constructor

Name	Description

+ Venusaur(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.8 Class Squirtle (extends BasePal implements Evolvable)

Field

Name	Description
- EVOLUTION_LEVEL	

Constructor

Name	Description
+ Squirtle(int level)	Create new Pal at level = level

Method

Name	Description
+ boolean isEvolvable()	
+ BasePal evolve()	
+ void initailizaStats()	
+ void initailizaMoves()	

3.9 Class Wartortle (extends BasePal implements Evolvable)

Field

Name	Description
- EVOLUTION_LEVEL	

Constructor

Name	Description

+ Wartortle(int level)	Create new Pal at level = level

Method

Name	Description
+ boolean isEvolvable()	
+ BasePal evolve()	
+ void initailizaStats()	
+ void initailizaMoves()	

3.10 Class Blastoise (extends BasePal)

Constructor

Name	Description
+ Blastoise(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.11 Class Caterpie (extends BasePal implements Evolvable)

Field

Name	Description
- EVOLUTION_LEVEL	

Constructor

Name	Description
+ Caterpie(int level)	Create new Pal at level = level

Method

Name	Description
+ boolean isEvolvable()	
+ BasePal evolve()	
+ void initailizaStats()	
+ void initailizaMoves()	

3.12 Class Metapod (extends BasePal implements Evolvable)

Field

Name	Description
- EVOLUTION_LEVEL	

Constructor

Name	Description
+ Metapod(int level)	Create new Pal at level = level

Method

Name	Description
+ boolean isEvolvable()	
+ BasePal evolve()	
+ void initailizaStats()	
+ void initailizaMoves()	

3.13 Class Butterfree (extends BasePal)

Constructor

Name	Description
+ Butterfree(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.14 Class Pidgey (extends BasePal implements Evolvable)

Field

Name	Description
- EVOLUTION_LEVEL	

Constructor

Name	Description
+ Pidgey(int level)	Create new Pal at level = level

Method

Name	Description
+ boolean isEvolvable()	
+ BasePal evolve()	
+ void initailizaStats()	
+ void initailizaMoves()	

3.15 Class Pidgeotto (extends BasePal implements Evolvable)

Field

Name	Description
- EVOLUTION_LEVEL	

Constructor

Name	Description
+ Pidgeotto(int level)	Create new Pal at level = level

Method

Name	Description
+ boolean isEvolvable()	
+ BasePal evolve()	
+ void initailizaStats()	
+ void initailizaMoves()	

3.16 Class Pidgeot (extends BasePal)

Constructor

Name	Description
+ Pidgeot(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.17 Class Rattata (extends BasePal implements Evolvable)

Field

Name	Description
- EVOLUTION_LEVEL	

Constructor

Name	Description
+ Rattata(int level)	Create new Pal at level = level

Method

Name	Description
+ boolean isEvolvable()	
+ BasePal evolve()	
+ void initailizaStats()	
+ void initailizaMoves()	

3.18 Class Raticate (extends BasePal)

Constructor

Name	Description
+ Raticate(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.19 Class Spearow (extends BasePal implements Evolvable)

Field

Name	Description
- EVOLUTION_LEVEL	

Constructor

Name	Description
+ Spearow(int level)	Create new Pal at level = level

Method

Name	Description
+ boolean isEvolvable()	
+ BasePal evolve()	
+ void initailizaStats()	
+ void initailizaMoves()	

3.20 Class Fearow (extends BasePal)

Constructor

Name	Description
+ Fearow(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.21 Class Vulpix (extends BasePal implements Evolvable)

Field

Name	Description
- EVOLUTION_LEVEL	

Constructor

Name	Description
+ Vulpix(int level)	Create new Pal at level = level

Method

Name	Description
+ boolean isEvolvable()	
+ BasePal evolve()	
+ void initailizaStats()	
+ void initailizaMoves()	

3.22 Class Ninetales (extends BasePal)

Constructor

Name	Description
+ Ninetales(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.23 Class Meowth (extends BasePal implements Evolvable)

Field

Name	Description
- EVOLUTION_LEVEL	

Constructor

Name	Description
+ Meowth(int level)	Create new Pal at level = level

Method

Name	Description
+ boolean isEvolvable()	
+ BasePal evolve()	
+ void initailizaStats()	
+ void initailizaMoves()	

3.24 Class Persian (extends BasePal)

Constructor

Name	Description
+ Persian(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.25 Class Growlithe (extends BasePal implements Evolvable)

Field

Name	Description
- EVOLUTION_LEVEL	

Constructor

Name	Description
+ Growlithe(int level)	Create new Pal at level = level

Method

Name	Description
+ boolean isEvolvable()	
+ BasePal evolve()	
+ void initailizaStats()	
+ void initailizaMoves()	

3.26 Class Arcanine (extends BasePal)

Constructor

Name	Description
+ Arcanine(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.27 Class Exeggcute (extends BasePal implements Evolvable)

Field

Name	Description
- EVOLUTION_LEVEL	

Constructor

Name	Description
+ Exeggcute(int level)	Create new Pal at level = level

Method

Name	Description
+ boolean isEvolvable()	
+ BasePal evolve()	
+ void initailizaStats()	
+ void initailizaMoves()	

3.28 Class Exeggutor (extends BasePal)

Constructor

Name	Description
+ Exeggutor(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.29 Class Staryu (extends BasePal implements Evolvable)

Field

Name	Description
- EVOLUTION_LEVEL	

Constructor

Name	Description
+ Staryu(int level)	Create new Pal at level = level

Method

Name	Description
+ boolean isEvolvable()	
+ BasePal evolve()	
+ void initailizaStats()	
+ void initailizaMoves()	

3.30 Class Starmie (extends BasePal)

Constructor

Name	Description
+ Starmie(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.31 Class Scyther (extends BasePal)

Constructor

Name	Description
+ Scyther(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.32 Class Magikarp (extends BasePal implements Evolvable)

Field

Name	Description
- EVOLUTION_LEVEL	

Constructor

Name	Description
+ Magikarp(int level)	Create new Pal at level = level

Method

Name	Description
+ boolean isEvolvable()	
+ BasePal evolve()	
+ void initailizaStats()	
+ void initailizaMoves()	

3.33 Class Gyarados (extends BasePal)

Constructor

Name	Description
+ Gyarados(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.34 Class Snorlax (extends BasePal)

Constructor

Name	Description
+ Snorlax(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.35 Class Articuno (extends BasePal)

Constructor

Name	Description
+ Articuno(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.36 Class Moltres (extends BasePal)

Constructor

Name	Description
+ Moltres(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.37 Class Mewtwo (extends BasePal)

Constructor

Name	Description
+ Mewtwo(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.38 Class Dratini (extends BasePal implements Evolvable)

Field

Name	Description
- EVOLUTION_LEVEL	

Constructor

Name	Description
+ Dratini(int level)	Create new Pal at level = level

Method

Name	Description
+ boolean isEvolvable()	
+ BasePal evolve()	
+ void initailizaStats()	
+ void initailizaMoves()	

3.39 Class Dragonair (extends BasePal implements Evolvable)

Field

Name	Description
- EVOLUTION_LEVEL	

Constructor

Name	Description
+ Dragonair(int level)	Create new Pal at level = level

Method

Name	Description
+ boolean isEvolvable()	
+ BasePal evolve()	
+ void initailizaStats()	
+ void initailizaMoves()	

3.40 Class Dragonite (extends BasePal)

Constructor

Name	Description
+ Dragonite(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.41 Class Scizor (extends BasePal)

Constructor

Name	Description
+ Scizor(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.42 Class Zigzagoon (extends BasePal implements Evolvable)

Field

Name	Description
- EVOLUTION_LEVEL	

Constructor

Name	Description
+ Zigzagoon(int level)	Create new Pal at level = level

Method

Name	Description
+ boolean isEvolvable()	
+ BasePal evolve()	
+ void initailizaStats()	
+ void initailizaMoves()	

3.43 Class Linoone (extends BasePal)

Constructor

Name	Description
+ Dragonite(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.44 Class Bidoof (extends BasePal implements Evolvable)

Field

Name	Description
- EVOLUTION_LEVEL	

Constructor

Name	Description
+ Bidoof(int level)	Create new Pal at level = level

Method

Name	Description
+ boolean isEvolvable()	
+ BasePal evolve()	
+ void initailizaStats()	
+ void initailizaMoves()	

3.45 Class Bibarel (extends BasePal)

Constructor

Name	Description
+ Bibarel(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.46 Class Kyogre (extends BasePal)

Constructor

Name	Description
+ Kyogre(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.47 Class Groudon (extends BasePal)

Constructor

Name	Description
+ Groudon(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

3.48 Class Rayquaza (extends BasePal)

Constructor

Name	Description
+ Rayquaza(int level)	Create new Pal at level = level

Method

Name	Description
+ void initailizaStats()	
+ void initailizaMoves()	

4. Package state

4.1 Class StateRoute1

Field

Name	Description
- Class<?>[] PAL_CLASSES	Contain what Pal can generate by this state.

Method

Name	Description
+ static BasePal generateRandomPal(int level)	Random create new BasePal object that is instance of class in PAL_CLASSES

4.2 Class StateRoute2

Field

Name	Description
- Class<?>[] PAL_CLASSES	Contain what Pal can generate by this state.

Method

Name	Description
+ static BasePal generateRandomPal(int level)	Random create new BasePal object that is instance of class in PAL_CLASSES

4.3 Class StateRoute3

Field

Name	Description
- Class<?>[] PAL_CLASSES	Contain what Pal can generate by this state.

Method

Name	Description
+ static BasePal generateRandomPal(int level)	Random create new BasePal object that is instance of class in PAL_CLASSES

4.4 Class StateRoute4

Field

Name	Description
- Class<?>[] PAL_CLASSES	Contain what Pal can generate by this state.

Method

Name	Description
+ static BasePal generateRandomPal(int level)	Random create new BasePal object that is instance of class in PAL_CLASSES

4.5 Class StateRoute5

Field

Name	Description
- Class<?>[] PAL_CLASSES	Contain what Pal can generate by this state.

Method

Name	Description
+ static BasePal generateRandomPal(int level)	Random create new BasePal object that is instance of class in PAL_CLASSES

5. Package utils

5.1 Enum Type

Represents the different Pal’s type. There are GRASS, FIRE, WATER, NORMAL and DRAGON.

5.2 Enum MoveType

Represents the different Pal’s move type. There are PHYSICAL and SPECIAL.

5.3 Interface Evolvable

Method

Name	Description
boolean isEvolvable()	Return this.level >= EVOLUTION_LEVEL
BasePal evolve()	Return next evolution of each Pal.

6. Package game

6.1 Class GameController

Field

Name	Description
- int money	Currency use to upgrade player stat.
- int pocket	Currency use to buy item in reach run.
- int xHp	HP stat multiplier.
- int xAtt	Atk stat multiplier.
- int xDef	Def stat multiplier.
- int xSpAtt	SpAtk stat multiplier.

- int xSpDef	SpDef stat multiplier.
- int xSpeed	Speed stat multiplier.
- int wave	Current wave.
- ArrayList<BasePal> pals	List of player Pals.
- ArrayList<BasePal> enemy	List of enemy.
- LinkedHashMap<String, Integer> items	List of items in player bag.
- int selectPal	Index of displayed player Pal.
- GameController *instance*	game instance.

Constructor

Name	Description
- GameController()	create an instance if it doesn't exist.

Method

Name	Description
+ GameController getInstance()	Return instance.

- void initGame()	Set stat for start the game
	this.money = 100; // Reset money
	this.pocket = 1000; // Reset pocket
	this.selectPal = 0;
	this.wave = 1;
	this.xHp = this.xAtt = this.xDef = this.xSpAtt = this.xSpDef = this.xSpeed = 0; // Reset stats
	this.enemy.add(StateRoute1.generateRandomPal(1));
	addItem(new PalBall(20));
	addItem(new PokeBall(10));
	addItem(new Potion(20));
	addItem(new SuperPotion(10));
	addItem(new Revive(5));

+ void resetGame()	this.pals.clear();
	this.enemy.clear();

	this.items.clear();
	
	// Keep the values of money and stats intact
	this.selectPal = 0;
	this.wave = 1;
	
	// Generate a new set of enemies and items
	
	this.enemy.add(StateRoute1.*generateRandomPal*(1));
	addItem(new PalBall(20));
	addItem(new PokeBall(10));
	addItem(new Potion(20));
	addItem(new SuperPotion(10));
	addItem(new Revive(5));

+ void setSelectPal(int selectPal)	this.selectPal = selectPal.
+ void addPal(BasePal newPal)	Add newPal to instance.pals
+ void addItem(BaseItem newItem)	Add newItem amount to instance.items
+ boolean isGameOver()	Return false if any Pal is still alive.
+ void nextWave()	Go to next wave

7. Package gui

7.1 Class MenuScene

Method

Name	Description
+ Parent createContent()	Create a scene to be a menu of the game
- MenuBox extends VBox	Group created menu items
- MenuItems extends StackPane	Method to create a rectangle that have text that will addClickevent to go to another Page
- Title extends StackPane	Create title of the scene

7.2 Class UpgradePage

Name	Description
+ Parent cratePage()	Create UpgradePage for buy stat multiplier

7.3 Class SelectPalPage

Name	Description
+ Parent createContent()	Create a page for select starter Pal.
- MenuBox extends VBox	Group created menu items
- MenuItems extends StackPane	Method to create a rectangle that have text that will addClickevent to go to another Page
- Title extends StackPane	Create title of the scene
- UpgradeItem extends VBox	Method to link between UI and GameController eg.when click to buy and UI update GameController also update value

7.4 Class StartPage

Field

Name	Description
- ImageView backgroundImage	Static value of UI component to change in game loop
- ImageView frontImage	Static value of UI component to change in game loop
- ImageView backImage	Static value of UI component to change in game loop
- ProgressBar hpProgress1	Static value of UI component to change in game loop
- ProgressBar hpProgress2	Static value of UI component to change

	in game loop
- ProgressBar expProgress	Static value of UI component to change in game loop
- ImageView playerType	Static value of UI component to change in game loop
- ImageView enemyType	Static value of UI component to change in game loop
- Text pocket	Static value of UI component to change in game loop
- Text wave	Static value of UI component to change in game loop
- Text context	Static value of UI component to change in game loop
- Text playerName	Static value of UI component to change in game loop
- Text playerLevel	Static value of UI component to change in game loop
- Text enemyName	Static value of UI component to change in game loop
- Text enemyLevel	Static value of UI component to change in game loop
- Text levelFractionText	Static value of UI component to change in game loop
- Boolean playerTurn	Static value to check palyerturn
- boolean isAnimating	Static value to check is animation finished
- Button fightButton	Static value of UI component to change in game loop
- Button palButton	Static value of UI component to change in game loop
- Button cageButton	Static value of UI component to change in game loop

- Button skill1Button	Button to use Move 1 of BasePal
- Button skill2Button	Button to use Move 2 of BasePal
- Button skill3Button	Button to use Move 3 of BasePal
- Button skill4Button	Button to use Move 4 of BasePal
- Button item1	Button to use item 1 of BasePal
- Button item2	Button to use item 2 of BasePal
- Button item3	Button to use item 3 of BasePal
- Button item4	Button to use item 4 of BasePal
- Button item5	Button to use item 5 of BasePal
- Move skill	Move that player choose
- int skillindex	Index of move that player choose
- BasePal player	BasePal that player currently use
- int button	Static value to check bag open or close
- Boolean capture	Check last activity does player caught Pal
- Timeline gameLoop	Loop of the game

Method

Name	Description
+ Parent createPage()	Create page to play a game load photo of player ,enemy, etc.
- void startGameLoop()	Loop of the game that update all the value in the game
- void handleSkillSelection(int index)	Use index to call move by index
- void animateHpBar(ProgressBar progressBar, double newProgress, Duration duration)	Method to create animation of hp bar
- void loadNextEnemy()	Method to get next enemy and tell player that last Pal have been defeated
- void addWriteOnAnimation(Text	Method to make text appear one by one

textNode, String fullText, Duration duration, Runnable onComplete)	
- void setPlayerControlsEnabled(boolean enabled)	Method to disable player button
- void addSlideInAnimation(ImageView imageView, double targetX, double targetY, double screenWidth)	Method that make picture slide in the frame

7.5 Class SelectPalPage

Name	Description
+ Parent createPage()	Create a page for choose a starter Pal
- VBox createPokemonButton(BasePal pal)	Create a button to choose start Pal

7.6 Class ItemSelectPage

Field

Name	Description
- BasePal target;	Target of item
- int position	Mode of the item 1 is caught item 2 is consumption item
- BaseConsumption *item*;	Item that been use

Method

Name	Description
+ Parent createContent(int index,BasePal entity)	Create page for select Pal to release
+ Parent createContent(int index,BaseConsumption loaditem)	Create a page for the player to choose pal to use the item on.
- Pane createPokemonPanel(double x, double y,int index)	Method to create Pal box that contain name type hp etc.

7.7 Class EvolvePage

Name	Description
+ Parent createPage(BasePal)	Create a page to show what BasePal evolve into

7.8 Class GameoverPage

Name	Description
+ Parent createPage()	Create a page for showing when the run is over

7.9 Class WinPage

Name	Description
+ Parent createPage()	Create a page for showing when win