Pixoul Documentation

Created by

Pimrak Apiratchatanon 6332031221

Pisitpong Chongpipattanakul 6332032921

2110215 Programming Methodology

Semester 2 Year 2020

Chulalongkorn University

Contents

Introduction **1**

How to play **1**

Game Controls **1**

Monster **1**

Item **3**

- Weapon 3
- Armor 3
- Potion 3

Ladder **4**

In-Game Screen **4**

Inventory **6**

Game Over **7**

Win The Game **7**

Setting **7**

Class Diagram **9**

Documentation **14**

Introduction

Pixoul is a pixel roguelike game inspired by Pixel Dungeon and Soul Knight. In the game, you are an adventurer who explores a dangerous dungeon. You have to collect items, kill monsters, and go deeper to reach the deepest floor of the dungeon where the priceless treasure is kept.

How to play

The game is split into two parts, the player's turn and the monster's turn. In the player's turn, the player can do one action (move, stay still, attack monster, use item, equip item, unequip item, or delete item). After the player does action, It will be the monster's turn. All monsters will move, attack, or stay still. Then it will be the player's turn again. The game will end if the player reaches the deepest floor or the player’s health point is zero.

Game Controls

- `A` `S` `D` `F` Move the player.
- `Q` Stay still.
- `Esc` Close pause pane or inventory.

Monster

There are six types of monsters which are listed below. Some monsters have special abilities. If the distance between the monster and player is not more than three cells, the movable monsters will walk towards the player.

Monsters can attack the player if the player is in one of the eight cells around the monster. The player can attack the monster by clicking on it if the monster is in one of the eight cells around the player.

- **Skeleton:** Skeletons don't have special abilities.
- **Pumpkin Head:** Pumpkin Heads don't have special abilities.
- **Haunted Maid:** Haunted Maids don't have special abilities.
- **Soul:** Souls cannot move. They randomly attack the player. But when they attack, the player's health point will be reduced to one.
- **Reaper:** Reapers can inflict an immobilized effect on the player. The effect will be removed once the reaper is killed.
- **Dark Mage:** Dark Mages cannot deal damage directly to the player and cannot move. But they will randomly warp the player to a random room on the same floor, inflict poison effect on the player, or summon Skeletons.

Item

There are three types of items, weapon, armor, and potion. Items will be generated on the floor. You can walk on the cell that the item is on to collect it. You cannot have more than sixteen items in the inventory.

1. Weapon

Weapon will increase the player's attack point when equipped. You can equip one weapon once.

2. Armor

Armor will increase the player's defense point when equipped. You can equip one armor once.

3. Potion

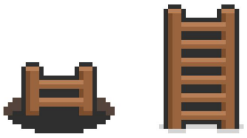
There are eight types of potions which are listed below. Some potions are permanent. Some have duration.

- **Instant Heal Potion:** Increases the player's current health point.
- **Permanent Health Potion:** Increases the player's max health point.
- **Protection Potion**: Increases the player's defense point.
- **Permanent Protection Potion**: Increases the player's defense point permanently.
- **Strength Potion:** Increases the player's attack point.

- **Permanent Strength Potion:** Increases the player's attack point permanently.
- **Regeneration Potion:** Increases the player's current health point in each turn for a certain duration.
- **Vision Potion:** Increases the player's vision for a certain duration.

Ladder

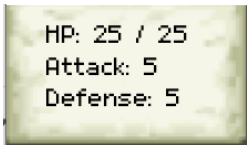
Ladders can move the player to the upper or the lower floor. You can walk on the ladder cell to climb it.



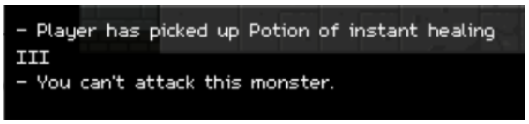
In-Game Screen

After the game starts, the player will be on the first floor of the dungeon. The player must find the ladder to go to the next floor. The player can walk on the item to collect it and click on the monster in one of the eight cells around the player to attack it.

The status pane at the top left shows the player's current health point, max health point, attack value, defense value.



The message pane at the bottom left shows the player and monster's actions.



You can click to pause the game and click to open inventory.

Inventory

The inventory pane shows all of the items you have. At the start of the game, you will get a rusty knife, wooden armor, and max healing potion.

You can hover on the item to see the name and detail of the item.

You can double click on an item to use, equip or unequip it.

You can click on the bin to enter delete mode. In the delete mode, If you click on any item, it will be gone forever. Click on the bin again to exit delete mode.

Note: use, equip, unequip, or delete item will be considered as one turn.

Game Over

If the player's health points reach zero, the game will be over. You can start a new game, go back to the menu, or exit game

Win The Game

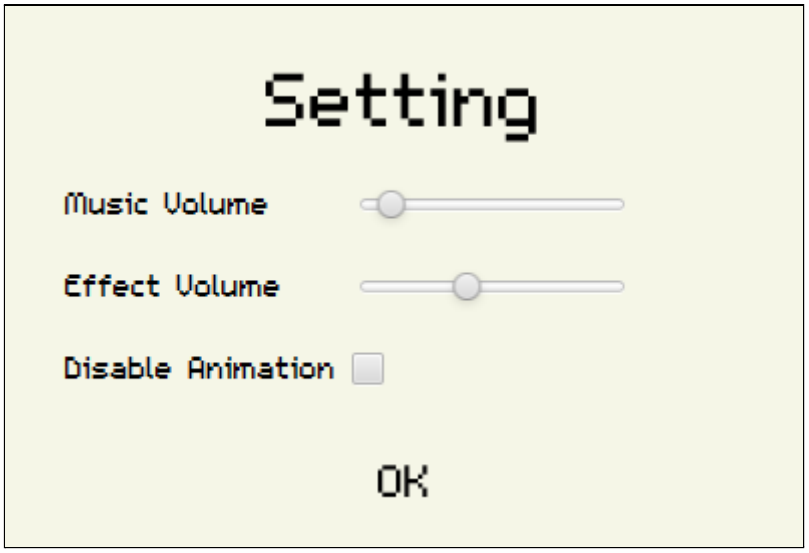
You will win the game if the player reaches the deepest floor. You can go back to the menu, or exit game

Setting

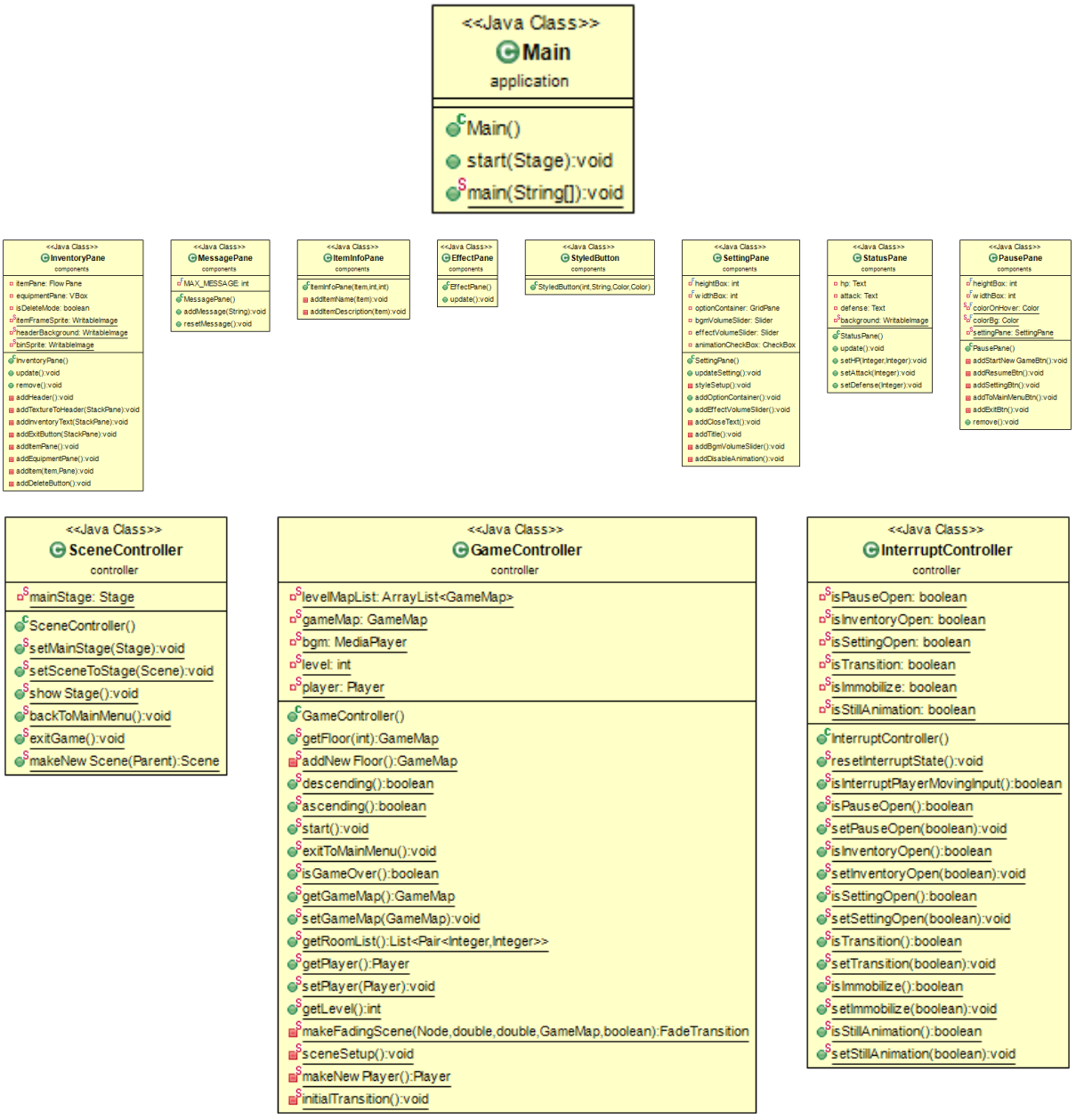
Pixoul has provided the setting to adjust some property in the game (List of the available setting are below)

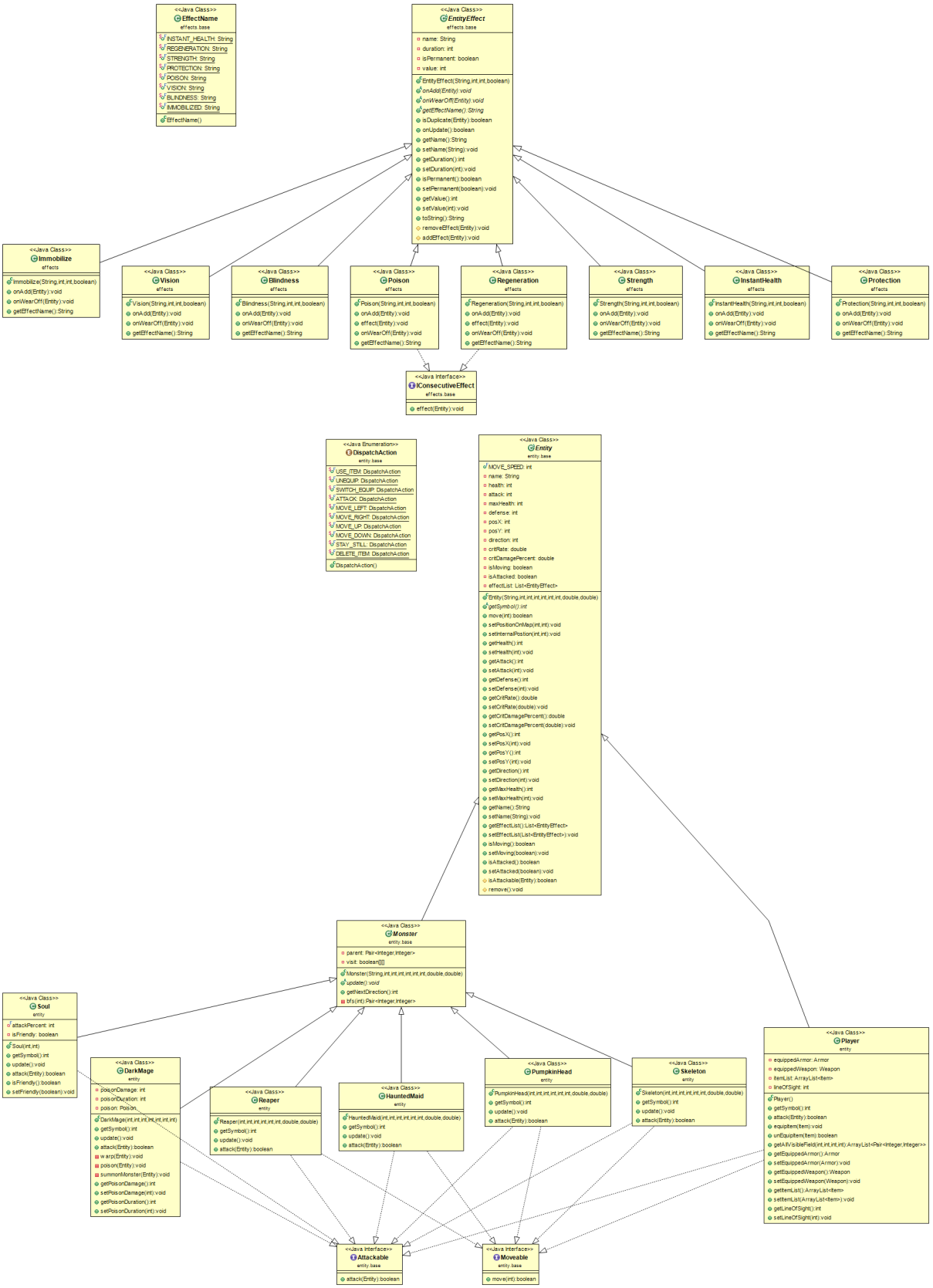
Available Setting

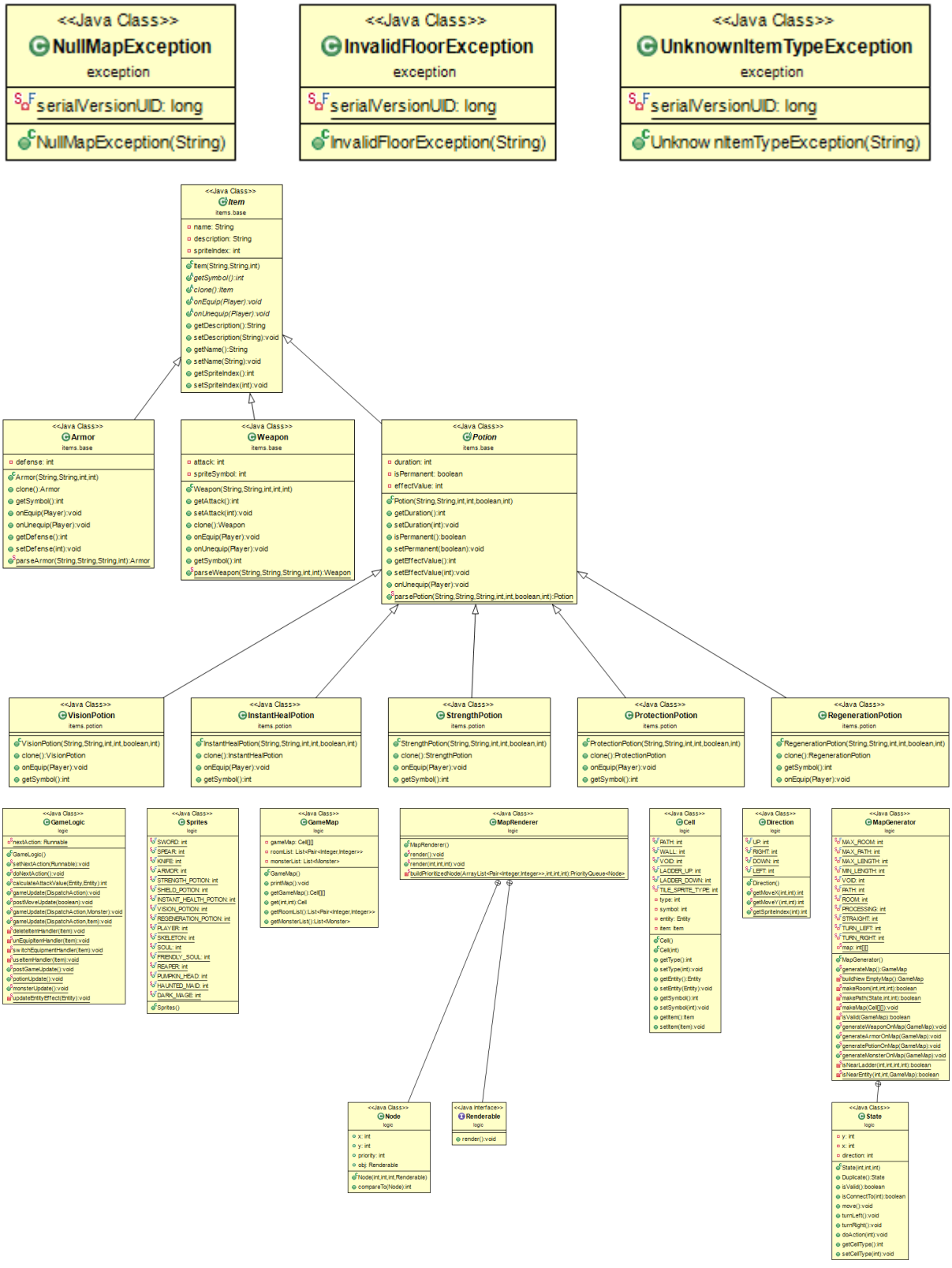
- Music Volume
- Effect Volume
- Disable Animation (In case of bad performance)

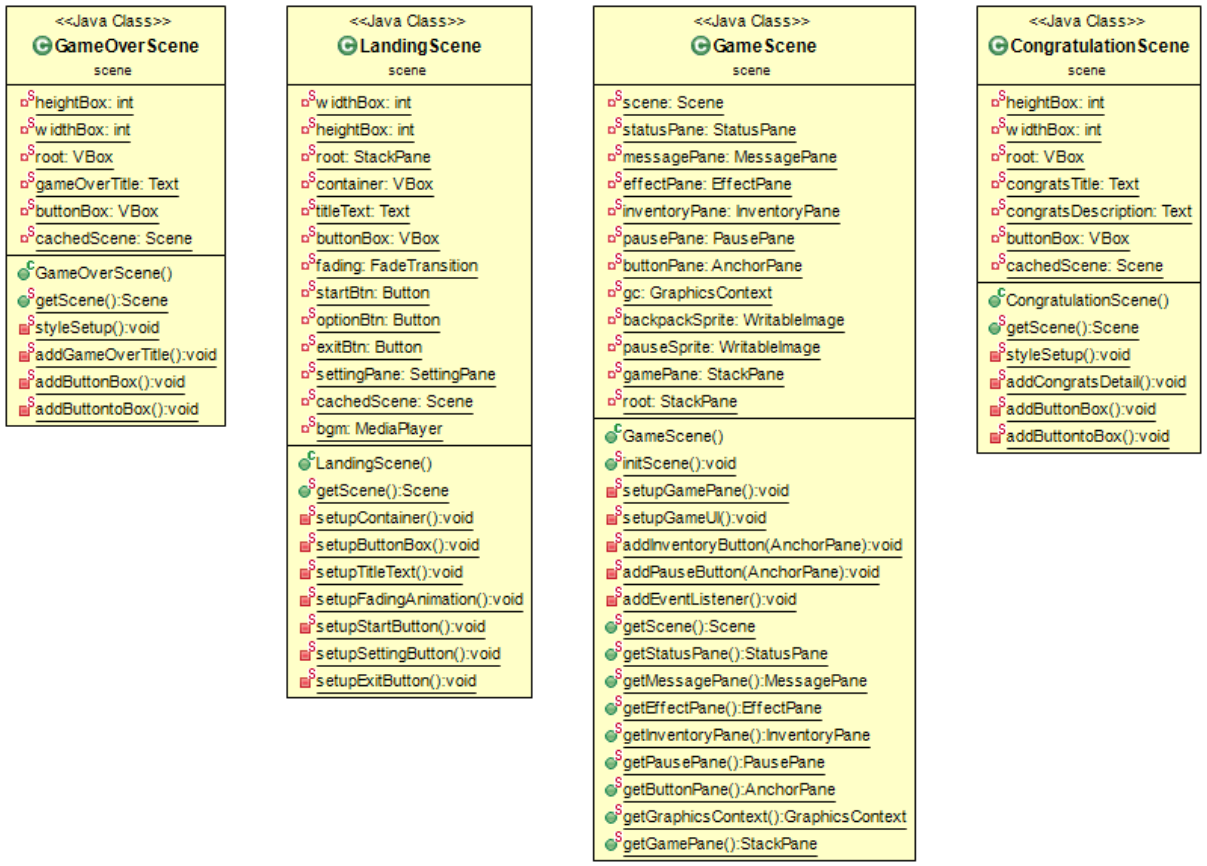


Class Diagram




EffectName
INSTANT_HEALTH: String
REGENERATION: String
STRENGTH: String
PROTECTION: String
POISON: String
VISION: String
BLINDNESS: String
IMMOBILIZED: String
EffectName()
EntityEffect
name: String
duration: int
isPermanent: boolean
value: int
EntityEffect(String,int,int,boolean)
onAdd(Entity): void
onWearOff(Entity): void
getEffectName(): String
isDuplicate(Entity): boolean
onUpdate(): boolean
getName(): String
setName(String): void
getDuration(): int
setDuration(int): void
isPermanent(): boolean
setPermanent(boolean): void
getValue(): int
setValue(int): void
toString(): String
removeEffect(Entity): void
addEffect(Entity): void
Immobilize
Vision
Blindness
Poison
Regeneration
Strength
InstantHealth
Protection
IConsecutiveEffect
effect(Entity): void
DispatchAction
USE_ITEM: DispatchAction
UNEQUIP: DispatchAction
SWITCH_EQUIP: DispatchAction
ATTACK: DispatchAction
MOVE_LEFT: DispatchAction
MOVE_RIGHT: DispatchAction
MOVE_UP: DispatchAction
MOVE_DOWN: DispatchAction
STAY_STILL: DispatchAction
DELETE_ITEM: DispatchAction
Entity
MOVE_SPEED: int
name: String
health: int
attack: int
maxHealth: int
defense: int
posX: int
posY: int
direction: int
critRate: double
critDamagePercent: double
isMoving: boolean
isAttacked: boolean
effectList: List<EntityEffect>
Monster
parent: Pair<Integer,Integer>
visit: boolean[][]
update(): void
getNextDirection(): int
bfs(int): Pair<Integer,Integer>
Soul
attackPercent: int
isFriendly: boolean
DarkMage
poisonDamage: int
poisonDuration: int
poison: Poison
Reaper
HauntedMaid
PumpkinHead
Skeleton
Player
equippedArmor: Armor
equippedWeapon: Weapon
itemList: ArrayList<Item>
lineOfSight: int
Attackable
attack(Entity): boolean
Moveable
move(int): boolean


<<Java Class>>
NullMapException
exception
serialVersionUID: long
NullMapException(String)
<<Java Class>>
InvalidFloorException
exception
serialVersionUID: long
InvalidFloorException(String)
<<Java Class>>
UnknownItemTypeException
exception
serialVersionUID: long
UnknownItemTypeException(String)
<<Java Class>>
Item
items.base
name: String
description: String
spriteIndex: int
Item(String,String,int)
getSymbol():int
clone():Item
onEquip(Player):void
onUnequip(Player):void
getDescription():String
setDescription(String):void
getName():String
setName(String):void
getSpriteIndex():int
setSpriteIndex(int):void
<<Java Class>>
Armor
items.base
defense: int
Armor(String,String,int,int)
clone():Armor
getSymbol():int
onEquip(Player):void
onUnequip(Player):void
getDefense():int
setDefense(int):void
parseArmor(String,String,String,int):Armor
<<Java Class>>
Weapon
items.base
attack: int
spriteSymbol: int
Weapon(String,String,int,int,int)
getAttack():int
setAttack(int):void
clone():Weapon
onEquip(Player):void
onUnequip(Player):void
getSymbol():int
parseWeapon(String,String,String,int,int):Weapon
<<Java Class>>
Potion
items.base
duration: int
isPermanent: boolean
effectValue: int
Potion(String,String,int,int,boolean,int)
getDuration():int
setDuration(int):void
isPermanent():boolean
setPermanent(boolean):void
getEffectValue():int
setEffectValue(int):void
onUnequip(Player):void
parsePotion(String,String,String,int,int,boolean,int):Potion
<<Java Class>>
VisionPotion
items.potion
VisionPotion(String,String,int,int,boolean,int)
clone():VisionPotion
onEquip(Player):void
getSymbol():int
<<Java Class>>
InstantHealPotion
items.potion
InstantHealPotion(String,String,int,int,boolean,int)
clone():InstantHealPotion
onEquip(Player):void
getSymbol():int
<<Java Class>>
StrengthPotion
items.potion
StrengthPotion(String,String,int,int,boolean,int)
clone():StrengthPotion
onEquip(Player):void
getSymbol():int
<<Java Class>>
ProtectionPotion
items.potion
ProtectionPotion(String,String,int,int,boolean,int)
clone():ProtectionPotion
onEquip(Player):void
getSymbol():int
<<Java Class>>
RegenerationPotion
items.potion
RegenerationPotion(String,String,int,int,boolean,int)
clone():RegenerationPotion
getSymbol():int
onEquip(Player):void
<<Java Class>>
GameLogic
logic
nextAction: Runnable
GameLogic()
setNextAction(Runnable):void
doNextAction():void
calculateAttackValue(Entity,Entity):int
gameUpdate(DispatchAction):void
postMoveUpdate(boolean):void
gameUpdate(DispatchAction,Monster):void
gameUpdate(DispatchAction,Item):void
deleteItemHandler(Item):void
unEquipItemHandler(Item):void
switchEquipmentHandler(Item):void
useItemHandler(Item):void
postGameUpdate():void
potionUpdate():void
monsterUpdate():void
updateEntityEffect(Entity):void
<<Java Class>>
Sprites
logic
SWORD: int
SPEAR: int
KNIFE: int
ARMOR: int
STRENGTH_POTION: int
SHIELD_POTION: int
INSTANT_HEALTH_POTION: int
VISION_POTION: int
REGENERATION_POTION: int
PLAYER: int
SKELETON: int
SOUL: int
FRIENDLY_SOUL: int
REAPER: int
PUMPKIN_HEAD: int
HAUNTED_MAID: int
DARK_MAGE: int
Sprites()
<<Java Class>>
GameMap
logic
gameMap: Cell[][]
roomList: List<Pair<Integer,Integer>>
monsterList: List<Monster>
GameMap()
printMap():void
getGameMap():Cell[][]
get(int,int):Cell
getRoomList():List<Pair<Integer,Integer>>
getMonsterList():List<Monster>
<<Java Class>>
MapRenderer
logic
MapRenderer()
render():void
render(int,int,int):void
buildPrioritizedNode(ArrayList<Pair<Integer,Integer>>,int,int,int):PriorityQueue<Node>
<<Java Class>>
Cell
logic
PATH: int
WALL: int
VOID: int
LADDER_UP: int
LADDER_DOWN: int
TILE_SPRITE_TYPE: int
type: int
symbol: int
entity: Entity
item: Item
Cell()
Cell(int)
getType():int
setType(int):void
getEntity():Entity
setEntity(Entity):void
getSymbol():int
setSymbol(int):void
getItem():Item
setItem(Item):void
<<Java Class>>
Direction
logic
UP: int
RIGHT: int
DOWN: int
LEFT: int
Direction()
getMoveX(int,int):int
getMoveY(int,int):int
getSpriteIndex(int):int
<<Java Class>>
MapGenerator
logic
MAX_ROOM: int
MAX_PATH: int
MAX_LENGTH: int
MIN_LENGTH: int
VOID: int
PATH: int
ROOM: int
PROCESSING: int
STRAIGHT: int
TURN_LEFT: int
TURN_RIGHT: int
map: int[][]
MapGenerator()
generateMap():GameMap
buildNewEmptyMap():GameMap
makeRoom(int,int,int):boolean
makePath(State,int,int):boolean
makeMap(Cell[][]):void
isValid(GameMap):boolean
generateWeaponOnMap(GameMap):void
generateArmorOnMap(GameMap):void
generatePotionOnMap(GameMap):void
generateMonsterOnMap(GameMap):void
isNearLadder(int,int,int):boolean
isNearEntity(int,int,GameMap):boolean
<<Java Class>>
Node
logic
x: int
y: int
priority: int
obj: Renderable
Node(int,int,int,Renderable)
compareTo(Node):int
<<Java Interface>>
Renderable
logic
render():void
<<Java Class>>
State
logic
y: int
x: int
direction: int
State(int,int,int)
Duplicate():State
isValid():boolean
isConnectTo(int):boolean
move():void
turnLeft():void
turnRight():void
doAction(int):void
getCellType():int
setCellType(int):void



Documentation

View the Javadoc at: https://pixoul-javadoc.web.app/