PvZ - CodeStrike

OOP Project

Advisor : Prof. Trần Thanh Tùng

TEAM MEMBERS

Nguyễn Thế Khoa

ITCSIU24042

Trần Khánh Bình

ITCSIU24015

Nguyễn Hưng

ITCSIU24030

Nguyễn Hữu Ninh

ITCSIU24066

Hoàng Triệu Nam

ITCSIU24059

CONTENTS

Introduction

In our simplified version of Plants vs Zombie game, the architecture adheres to Object-Oriented Programming (OOP) principles to maintain modularity, scalability, and ease of maintenance. Key OOP concepts applied include:

- **Encapsulation:** Game elements such as `Plant`, `Zombie`, and `Lawnmower` are structured with private attributes and controlled access via public methods.
- **Inheritance:** The `Zombie` class serves as a base type, with specialized subclasses such as `BrickheadZombie` inheriting core functionality while introducing unique mechanics.
- **Polymorphism:** Entities like `Zombie` and `Lawnmower` share a common interface, enabling dynamic behavior changes based on context without requiring explicit type-checking.

Additional gameplay enhancements:

- **Lawnmower mechanic** added, functioning as a last line of defense that automatically clears approaching enemies upon activation.
- **Level selection system** integrated, allowing players to choose specific stages rather than following a linear progression.

Gameplay

01. Characters

- Plants: Sunflower, Peashooter, Potato Mine, Walnut, and Cactus.
- Zombies: Basic Zombie, Conehead Zombie, Buckethead Zombie, and Brickhead Zombie.

02. Gameplay

- Collecting and using the sun to plant the Plants to defend your home.

03. Levels

- There are 4 levels to play through.
- After beating each level, you will achieve a new Plant.
- For the 4th level, this can be played by pressing “4” on the keyboard.

04. Running platform

- Using Greenfoot platform to execute the game.

GREENFOOT

GREENFOOT APPLICATION

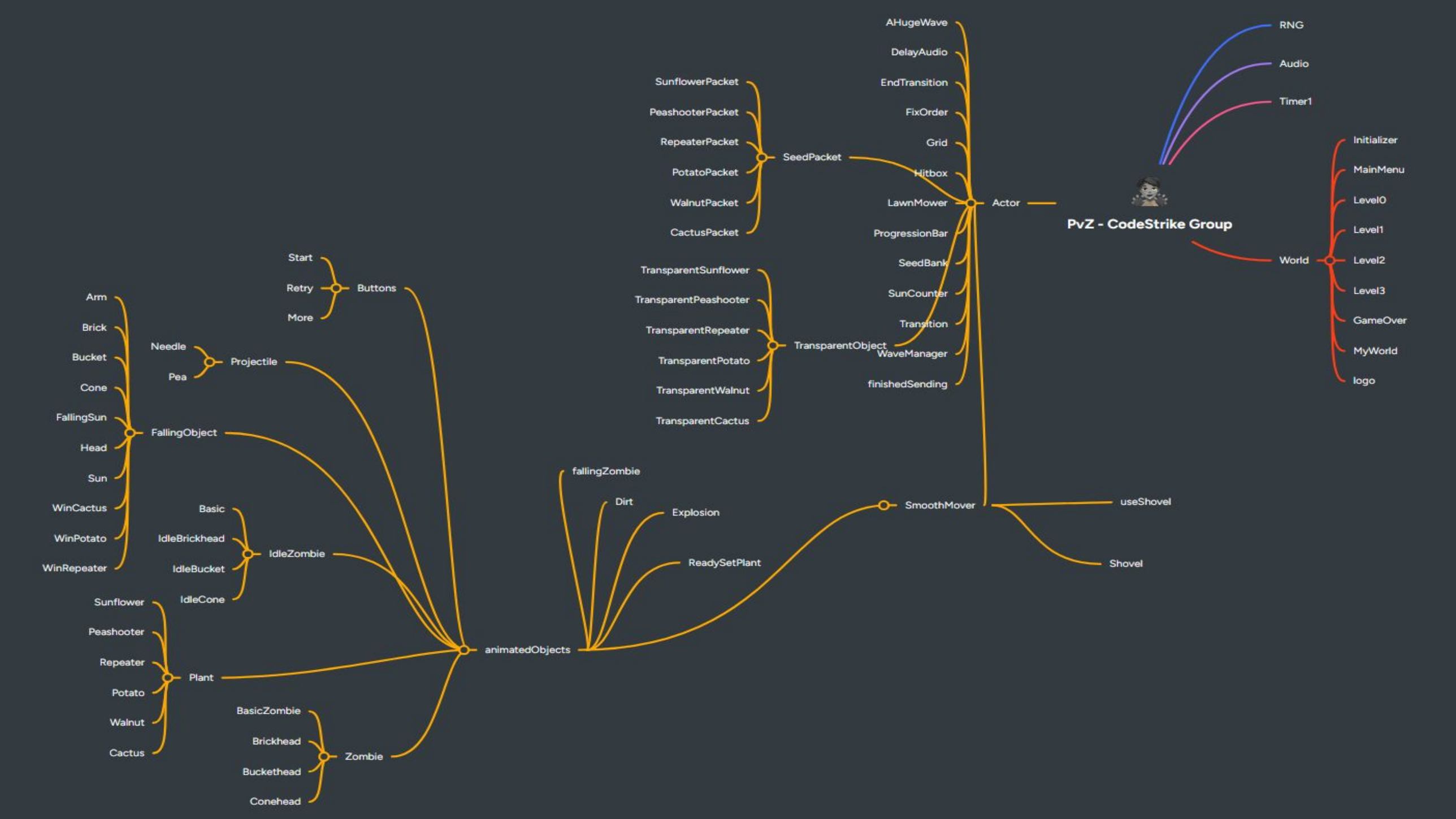
https://www.greenfoot.org/overview

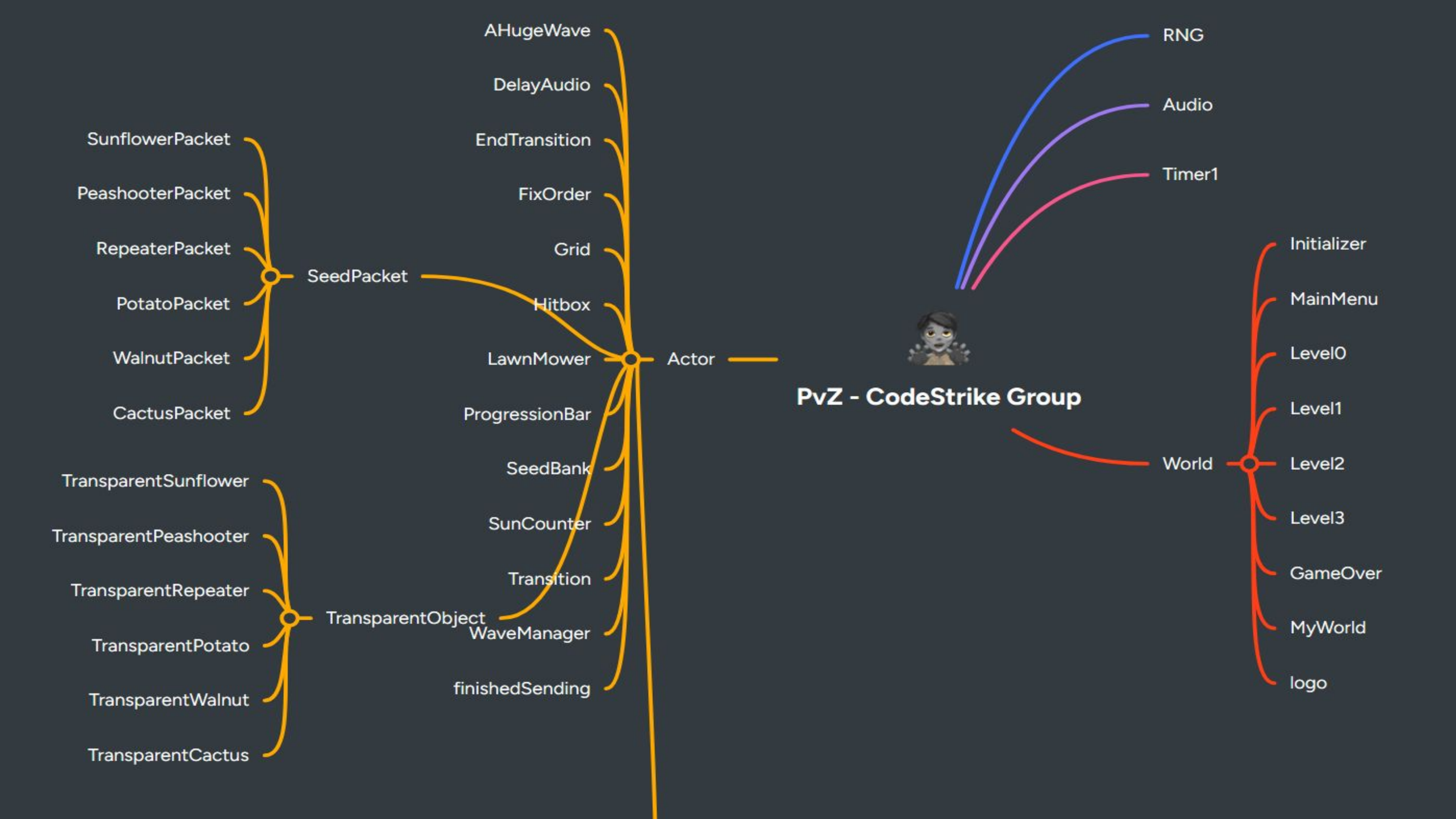
Greenfoot is an educational tool for learning Java
→ interactive
→ user-friendly
Highly recommended for budding coders !

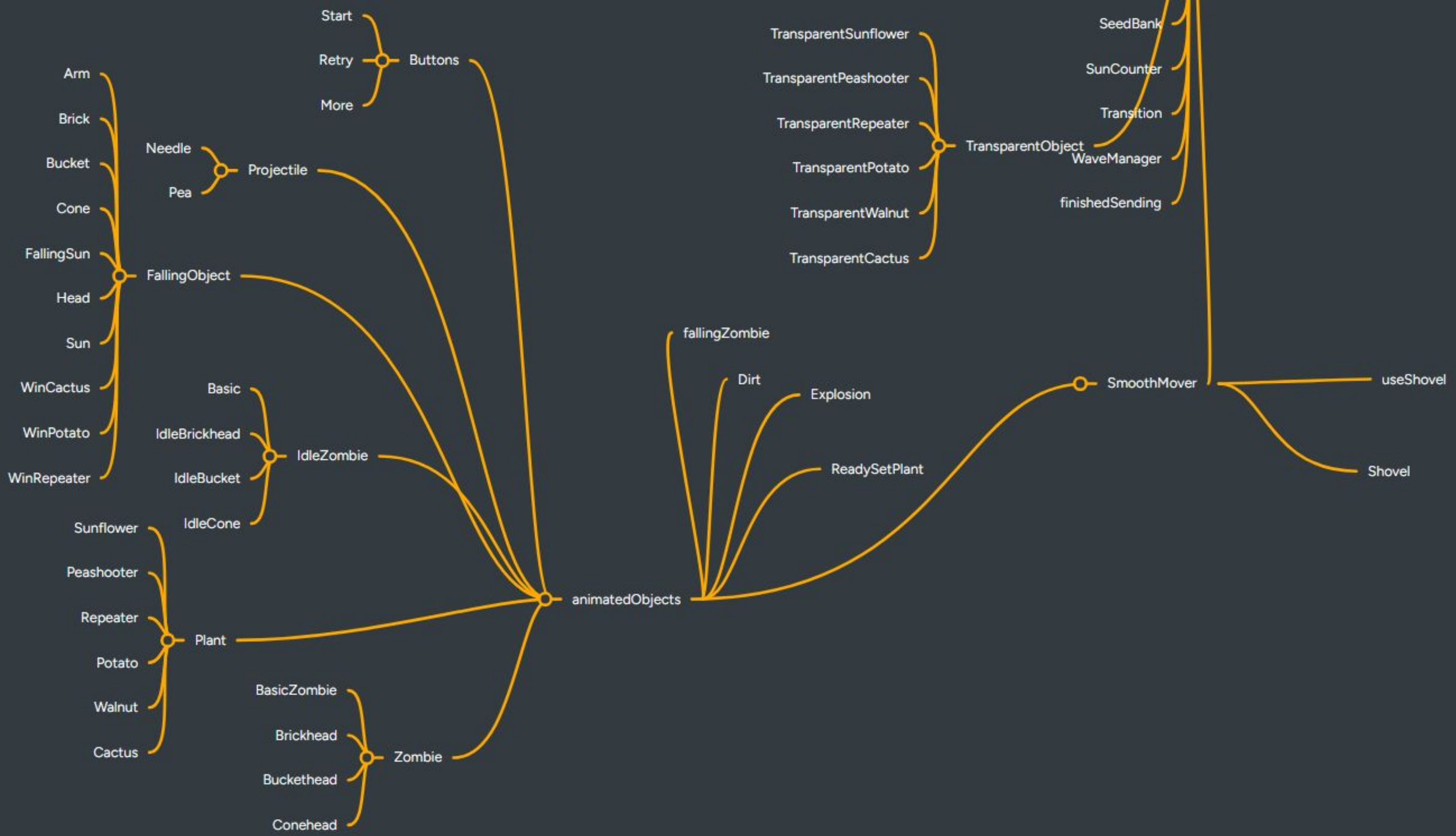
DEMO

https://www.youtube.com/watch?v=vKbh2-2IUVM

UML/Class Diagram


PvZ - CodeStrike Group
RNG
Audio
Timer1
World
Initializer
MainMenu
Level0
Level1
Level2
Level3
GameOver
MyWorld
logo
Actor
AHugeWave
DelayAudio
EndTransition
FixOrder
Grid
Hitbox
LawnMower
ProgressionBar
SeedBank
SunCounter
Transition
WaveManager
finishedSending
SeedPacket
SunflowerPacket
PeashooterPacket
RepeaterPacket
PotatoPacket
WalnutPacket
CactusPacket
TransparentObject
TransparentSunflower
TransparentPeashooter
TransparentRepeater
TransparentPotato
TransparentWalnut
TransparentCactus
SmoothMover
useShovel
Shovel
animatedObjects
fallingZombie
Dirt
Explosion
ReadySetPlant
Buttons
Start
Retry
More
Projectile
Needle
Pea
FallingObject
Arm
Brick
Bucket
Cone
FallingSun
Head
Sun
WinCactus
WinPotato
WinRepeater
IdleZombie
Basic
IdleBrickhead
IdleBucket
IdleCone
Plant
Sunflower
Peashooter
Repeater
Potato
Walnut
Cactus
Zombie
BasicZombie
Brickhead
Buckethead
Conehead


AHugeWave
DelayAudio
EndTransition
FixOrder
Grid
Hitbox
LawnMower
ProgressionBar
SeedBank
SunCounter
Transition
WaveManager
finishedSending
Actor
SeedPacket
SunflowerPacket
PeashooterPacket
RepeaterPacket
PotatoPacket
WalnutPacket
CactusPacket
TransparentObject
TransparentSunflower
TransparentPeashooter
TransparentRepeater
TransparentPotato
TransparentWalnut
TransparentCactus
RNG
Audio
Timer1
PvZ - CodeStrike Group
World
Initializer
MainMenu
Level0
Level1
Level2
Level3
GameOver
MyWorld
logo


Start
Retry
More
Buttons
Arm
Brick
Bucket
Cone
FallingSun
Head
Sun
WinCactus
WinPotato
WinRepeater
FallingObject
Needle
Pea
Projectile
Basic
IdleBrickhead
IdleBucket
IdleCone
IdleZombie
Sunflower
Peashooter
Repeater
Potato
Walnut
Cactus
Plant
BasicZombie
Brickhead
Buckethead
Conehead
Zombie
TransparentSunflower
TransparentPeashooter
TransparentRepeater
TransparentPotato
TransparentWalnut
TransparentCactus
TransparentObject
SeedBank
SunCounter
Transition
WaveManager
finishedSending
fallingZombie
Dirt
Explosion
ReadySetPlant
animatedObjects
SmoothMover
useShovel
Shovel

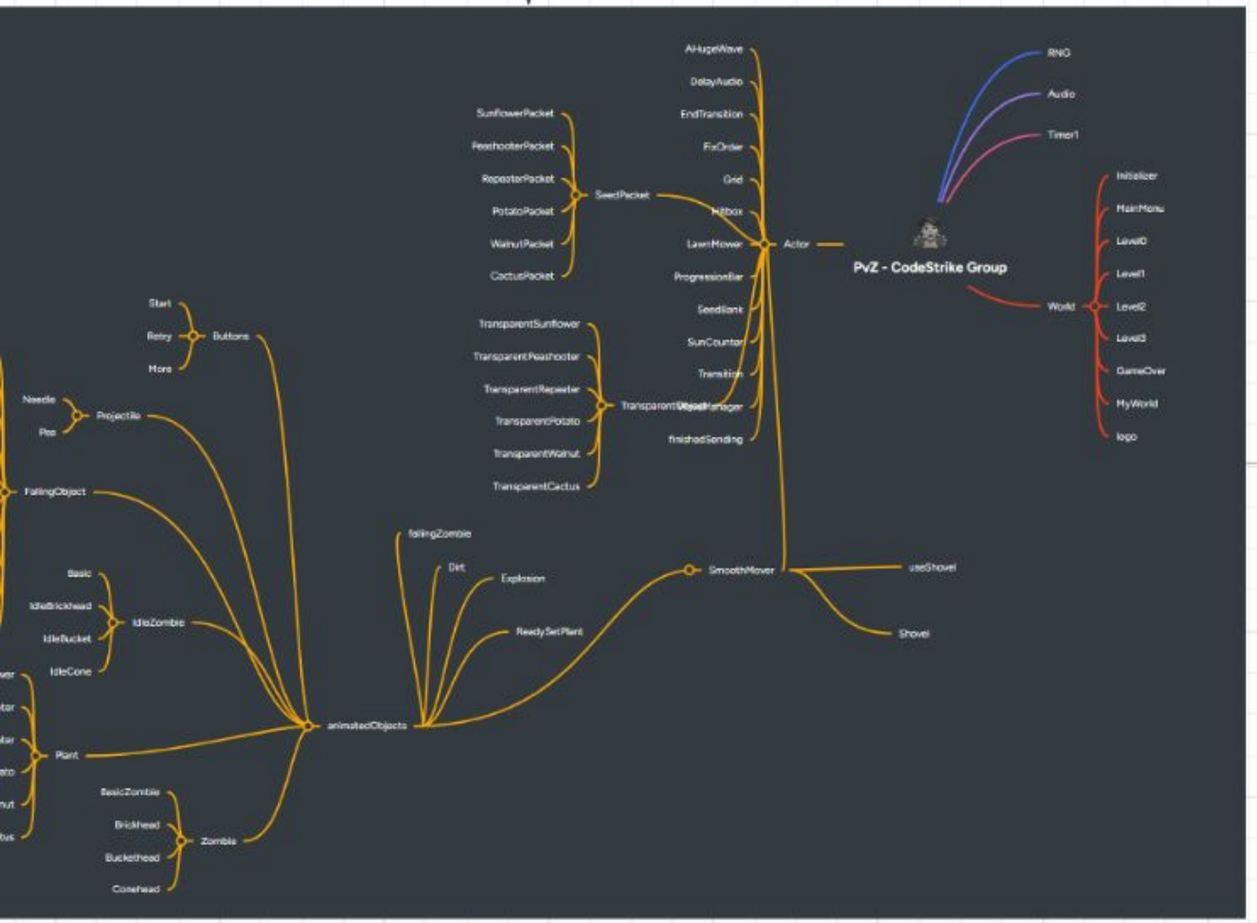
reenfoot
ion 2.0.0

sity of
nt

LA TROBE UNIVERSITY

Supported by Oracle

https://mm.tt/map/3724777316?t=MyTPSFwNut



World

World is the world that Actors live in. It is a two-dimensional grid of cells. All Actor are associated with a World and can get access to the world object. The size of cells can be specified at world creation time, and is constant after creation.

Greenfoot
https://www.greenfoot.org › files › javadoc › World

World (greenfoot API)

World

- worldWidth : int
- worldHeight : int
- cellSize : int
- bounded : boolean
- background : GreenfootImage

+ World(worldWidth: int, worldHeight: int, cellSize: int)
+ World(worldWidth: int, worldHeight: int, cellSize: int, bounded: boolean)
+ addObject(object: Actor, x: int, y: int) : void
+ removeObject(object: Actor) : void
+ removeObjects(objects: Collection<? extends Actor>) : void
+ getWidth() : int
+ getHeight() : int
+ getCellSize() : int
+ getObjects() : List<Actor>
+ getObjects(cls: Class) : List<Actor>
+ showText(text: String, x: int, y: int) : void
+ setBackground(image: GreenfootImage) : void
+ setBackground(filename: String) : void
+ repaint() : void
+ act() : void
+ started() : void
+ stopped() : void

SeedBank
<<extends Actor>>

+ MyWorld: MyWorld
+ suncounter: SunCounter = new SunCounter()
+ bank: SeedPacket[]
+ selectedPacket: SeedPacket = null
+ image: TransparentObject = null
+ transparent: TransparentObject = null

+ static final x1: int = 182
+ static final x2: int = 702
+ static final xSpacing: int = Grid.xSpacing
+ static final y1: int = 62
+ static final y2: int = 417
+ static final ySpacing: int = Grid.ySpacing

+ SeedBank(bank: SeedPacket[])
+ act(): void
+ addedToWorld(world: World): void

SeedPacket
<<extends Actor>>

+ deltaTime: long
+ deltaTime2: long
+ lastFrame: long = System.nanoTime()
+ lastFrame2: long = System.nanoTime()
+ rechargeTime: long
+ currentFrame: long = System.nanoTime()
+ sunCost: int
+ recharged: boolean = false
+ selected: boolean = false
+ doneRechargeTime: boolean = false
+ recharge: GreenfootImage
+ image1: GreenfootImage
+ image2: GreenfootImage
+ name: String
+ myWorld: MyWorld

+ SeedPacket(rechargeTime: long, recharged: boolean, sunCost: int, name: String)
+ addedToWorld(world: World): void
+ act(): void
+ handleRechargeProgress(): void
+ startRecharge(): void
+ setRecharged(charge: boolean): void
+ setSelected(selected: boolean): void
+ getCharge(): boolean
+ getSelected(): boolean
+ addImage(): TransparentObject
+ getPlant(): Plant

LawnMower
<<extends Actor>>

- activated: boolean
- speed: int = 7
- lawnmowerImage: GreenfootImage

+ LawnMower()
+ act() : void
- activate() : void

ProgressionBar
<<extends Actor>>

- level: WaveManager
- frameWidth: int = 458
- frameHeight: int = 34
- fillAreaWidth: int = 438
- fillAreaHeight: int = 14
- fillOffsetX: int = (frameWidth - fillAreaWidth) / 2
- fillOffsetY: int = (frameHeight - fillAreaHeight) / 2
- frame: GreenfootImage
- bar: GreenfootImage

+ ProgressionBar(level: WaveManager)
+ act(): void

Actor

Actor

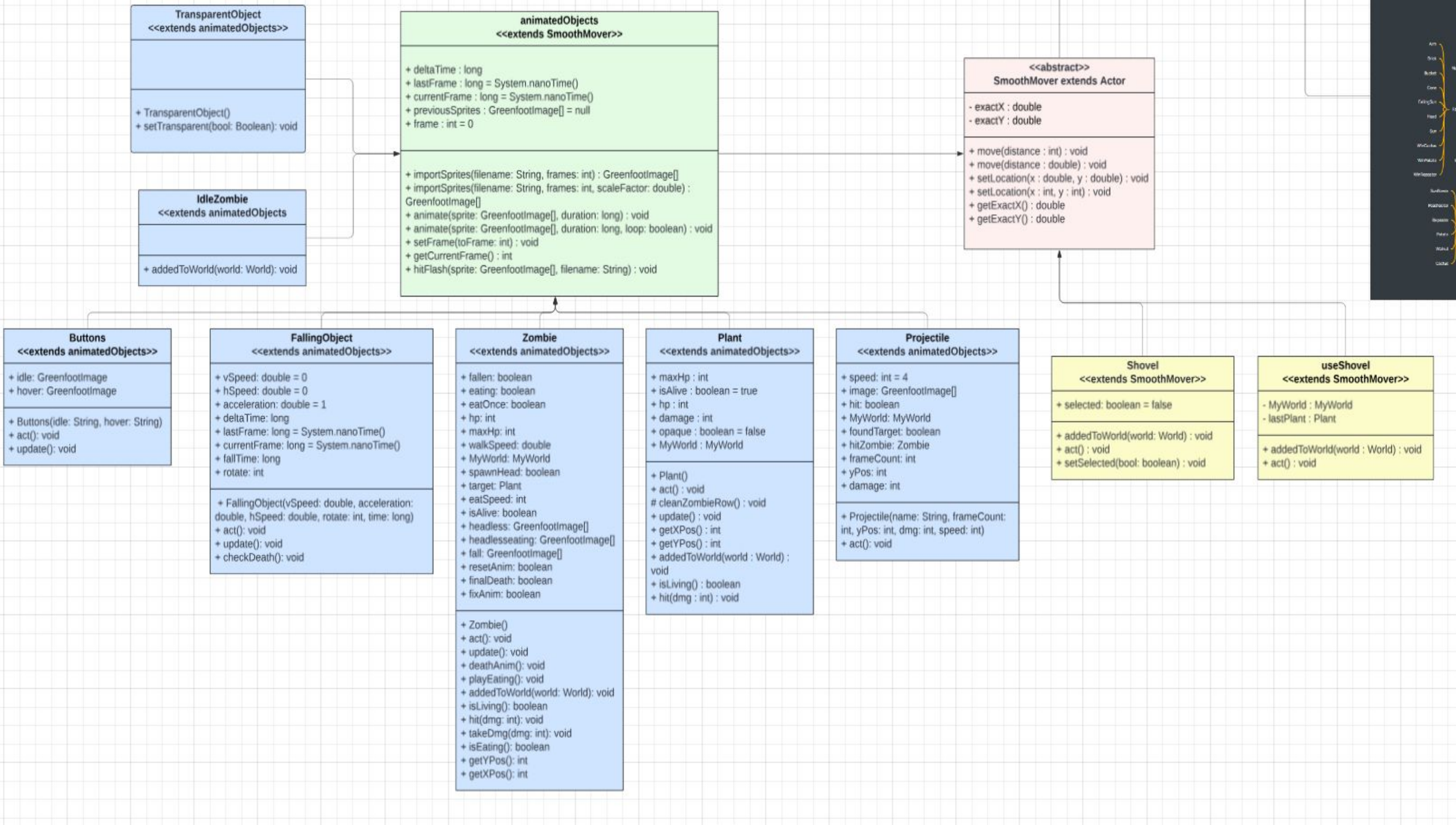
- image : GreenfootImage
- world : World
- x : int
- y : int

+ act() : void
+ getWorld() : World
+ setLocation(x: int, y: int) : void
+ move(distance: int) : void
+ turn(angle: int) : void
+ getX() : int
+ getY() : int
+ getImage() : GreenfootImage
+ setImage(image: GreenfootImage) : void
+ getRotation() : int
+ setRotation(angle: int) : void
+ isAtEdge() : boolean

An Actor is an object that exists in the Greenfoot world. Every Actor has a location in the world, and an appearance (that is: an icon). An Actor is not normally instantiated, but instead used as a superclass to more specific objects in the world. Every object that is intended to appear in the world must extend Actor.

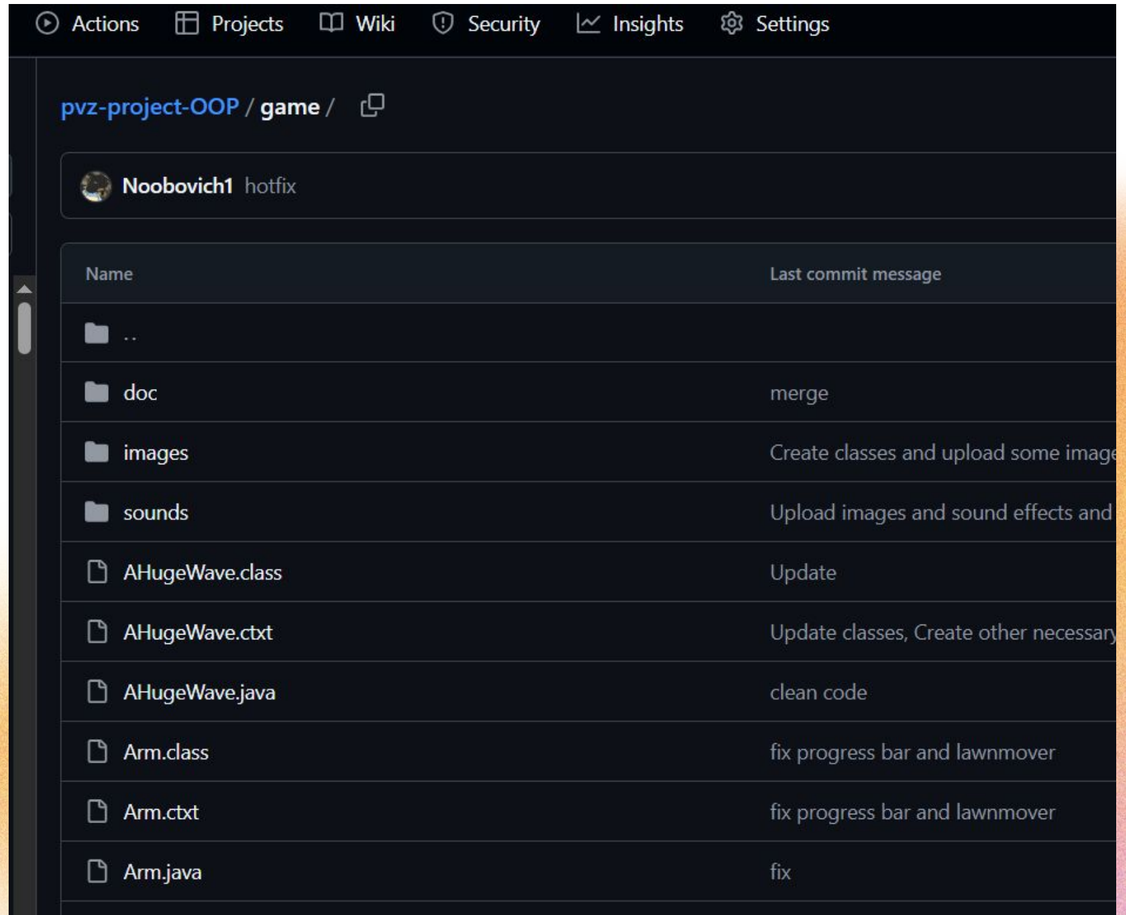
Greenfoot
https://www.greenfoot.org › files › javadoc › Actor

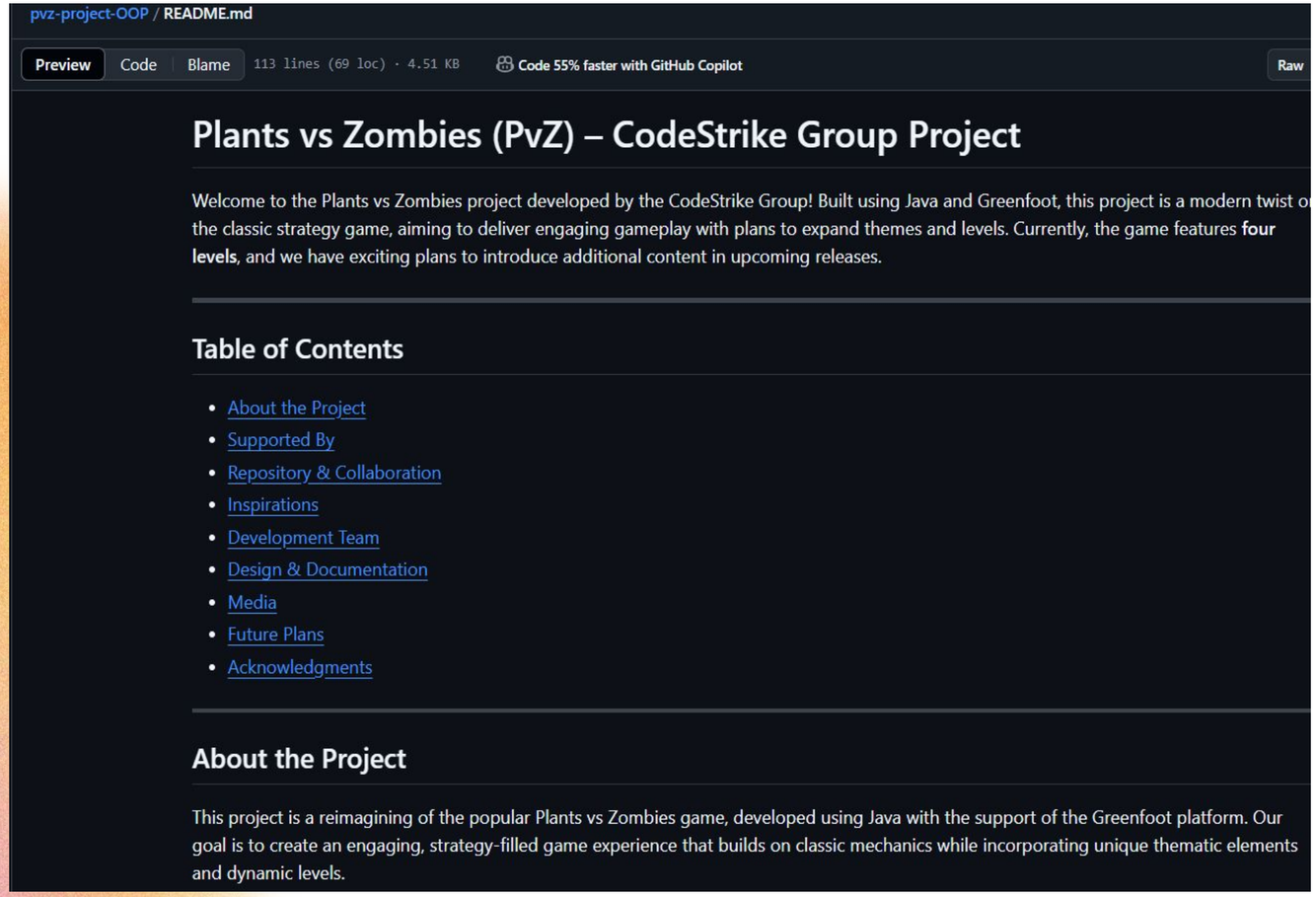
Actor (greenfoot API)


TransparentObject
<<extends animatedObjects>>
+ TransparentObject()
+ setTransparent(bool: Boolean): void
IdleZombie
<<extends animatedObjects
+ addedToWorld(world: World): void
animatedObjects
<<extends SmoothMover>>
+ deltaTime : long
+ lastFrame : long = System.nanoTime()
+ currentFrame : long = System.nanoTime()
+ previousSprites : GreenfootImage[] = null
+ frame : int = 0
+ importSprites(filename: String, frames: int) : GreenfootImage[]
+ importSprites(filename: String, frames: int, scaleFactor: double) : GreenfootImage[]
+ animate(sprite: GreenfootImage[], duration: long) : void
+ animate(sprite: GreenfootImage[], duration: long, loop: boolean) : void
+ setFrame(toFrame: int) : void
+ getCurrentFrame() : int
+ hitFlash(sprite: GreenfootImage[], filename: String) : void
<<abstract>>
SmoothMover extends Actor
- exactX : double
- exactY : double
+ move(distance : int) : void
+ move(distance : double) : void
+ setLocation(x : double, y : double) : void
+ setLocation(x : int, y : int) : void
+ getExactX() : double
+ getExactY() : double
Buttons
<<extends animatedObjects>>
+ idle: GreenfootImage
+ hover: GreenfootImage
+ Buttons(idle: String, hover: String)
+ act(): void
+ update(): void
FallingObject
<<extends animatedObjects>>
+ vSpeed: double = 0
+ hSpeed: double = 0
+ acceleration: double = 1
+ deltaTime: long
+ lastFrame: long = System.nanoTime()
+ currentFrame: long = System.nanoTime()
+ fallTime: long
+ rotate: int
+ FallingObject(vSpeed: double, acceleration: double, hSpeed: double, rotate: int, time: long)
+ act(): void
+ update(): void
+ checkDeath(): void
Zombie
<<extends animatedObjects>>
+ fallen: boolean
+ eating: boolean
+ eatOnce: boolean
+ hp: int
+ maxHp: int
+ walkSpeed: double
+ MyWorld: MyWorld
+ spawnHead: boolean
+ target: Plant
+ eatSpeed: int
+ isAlive: boolean
+ headless: GreenfootImage[]
+ headlesseating: GreenfootImage[]
+ fall: GreenfootImage[]
+ resetAnim: boolean
+ finalDeath: boolean
+ fixAnim: boolean
+ Zombie()
+ act(): void
+ update(): void
+ deathAnim(): void
+ playEating(): void
+ addedToWorld(world: World): void
+ isLiving(): boolean
+ hit(dmg: int): void
+ takeDmg(dmg: int): void
+ isEating(): boolean
+ getYPos(): int
+ getXPos(): int
Plant
<<extends animatedObjects>>
+ maxHp : int
+ isAlive : boolean = true
+ hp : int
+ damage : int
+ opaque : boolean = false
+ MyWorld : MyWorld
+ Plant()
+ act() : void
# cleanZombieRow() : void
+ update() : void
+ getXPos() : int
+ getYPos() : int
+ addedToWorld(world : World) : void
+ isLiving() : boolean
+ hit(dmg : int) : void
Projectile
<<extends animatedObjects>>
+ speed: int = 4
+ image: GreenfootImage[]
+ hit: boolean
+ MyWorld: MyWorld
+ foundTarget: boolean
+ hitZombie: Zombie
+ frameCount: int
+ yPos: int
+ damage: int
+ Projectile(name: String, frameCount: int, yPos: int, dmg: int, speed: int)
+ act(): void
Shovel
<<extends SmoothMover>>
+ selected: boolean = false
+ addedToWorld(world: World) : void
+ act() : void
+ setSelected(bool: boolean) : void
useShovel
<<extends SmoothMover>>
- MyWorld : MyWorld
- lastPlant : Plant
+ addedToWorld(world : World) : void
+ act() : void

GITHUB

Our github link: https://github.com/trankhanhbinh/pvz-project-OOP.git





CONCLUSION

By implementing a well-researched and innovative sales strategy, our goal is not only to boost immediate sales figures but also to establish a sustainable framework for continued growth and success

THANK YOU!