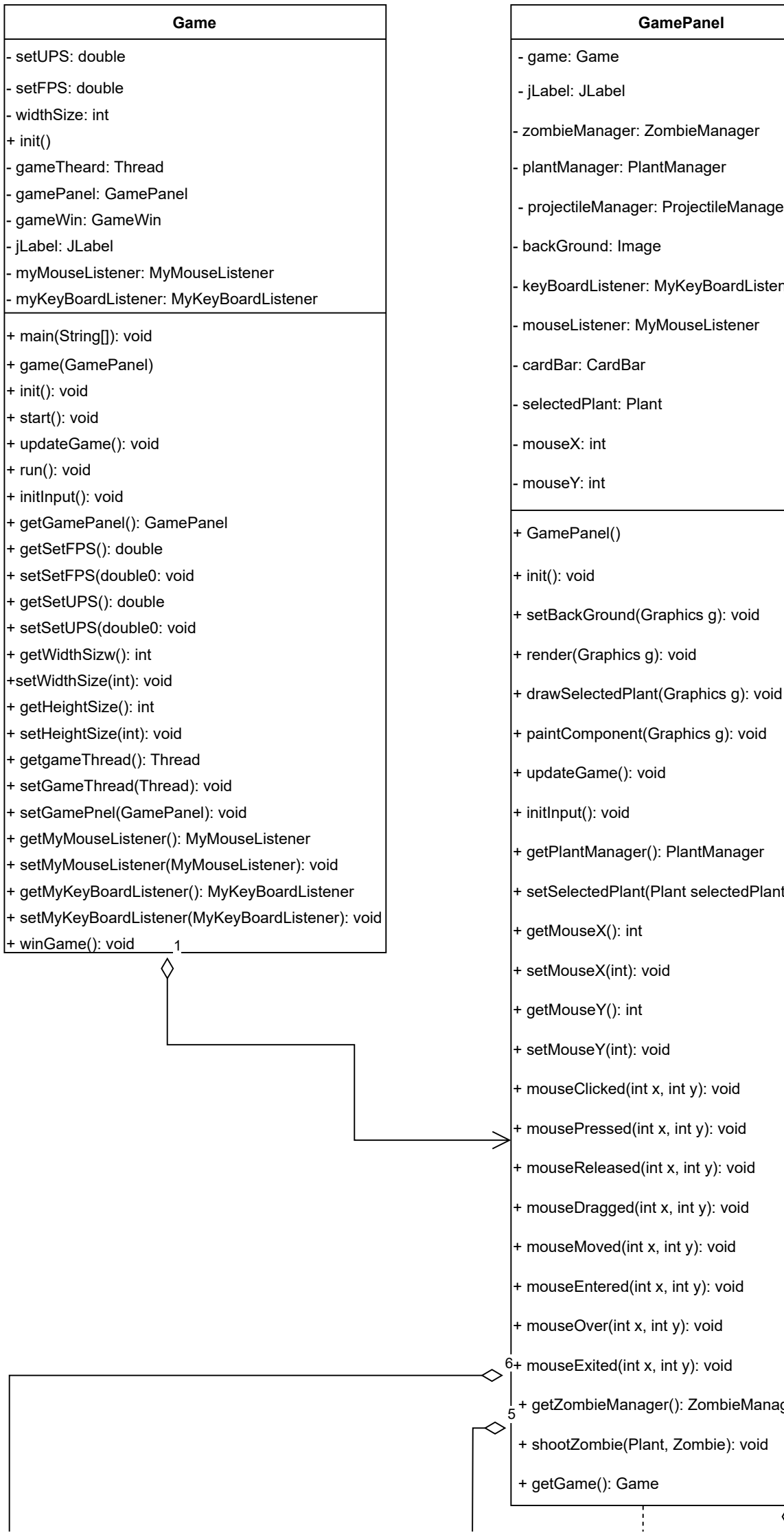
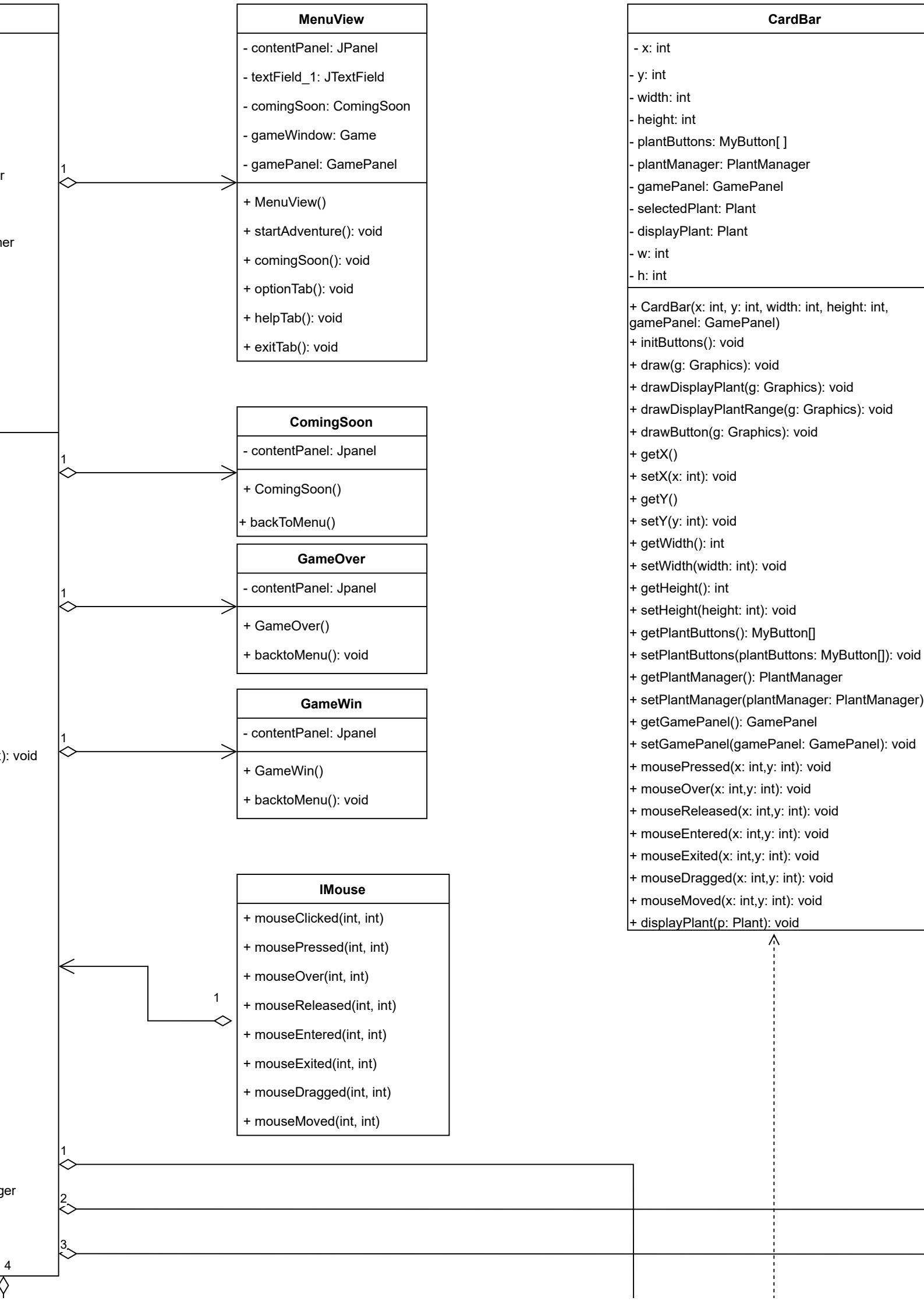
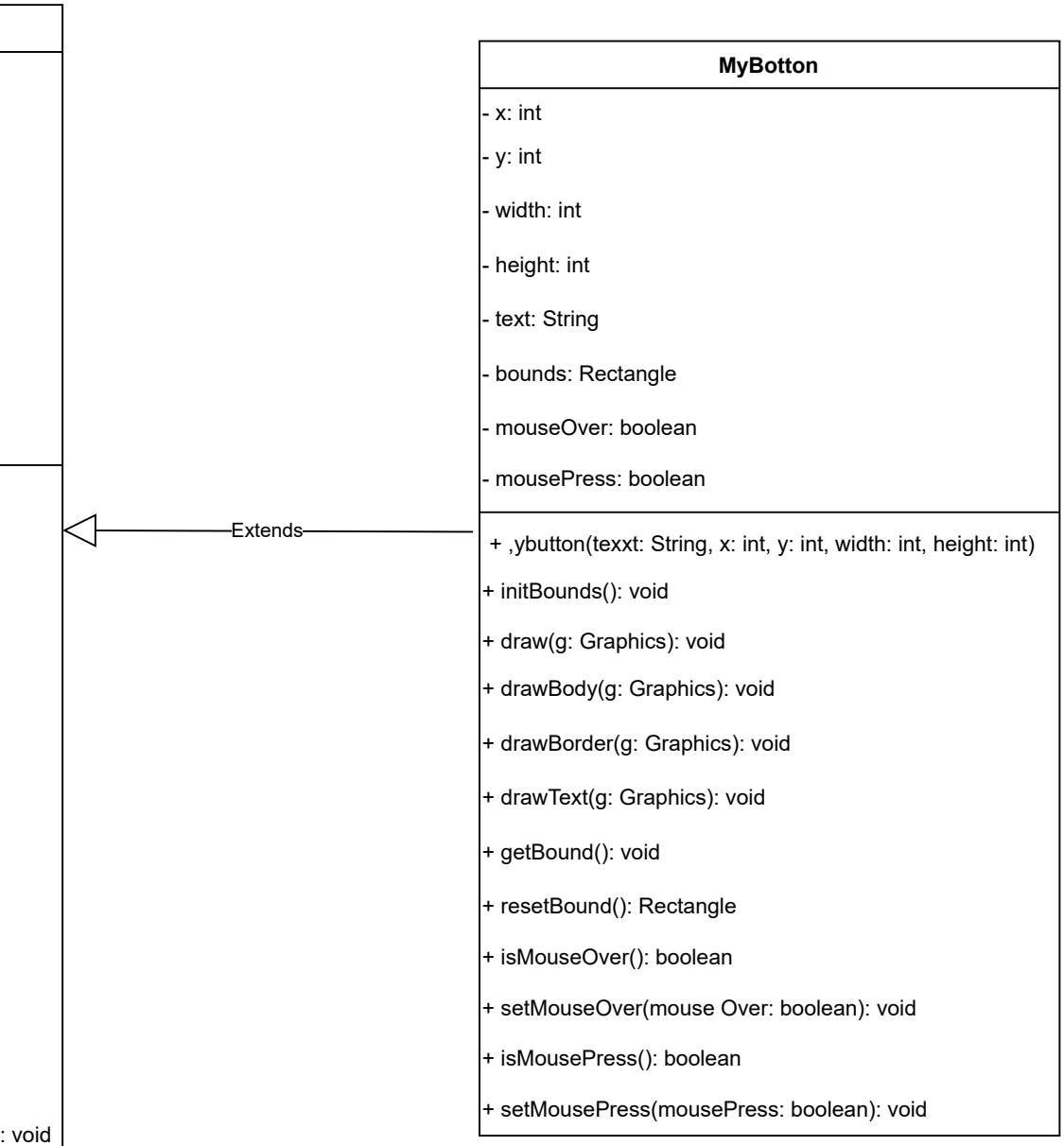
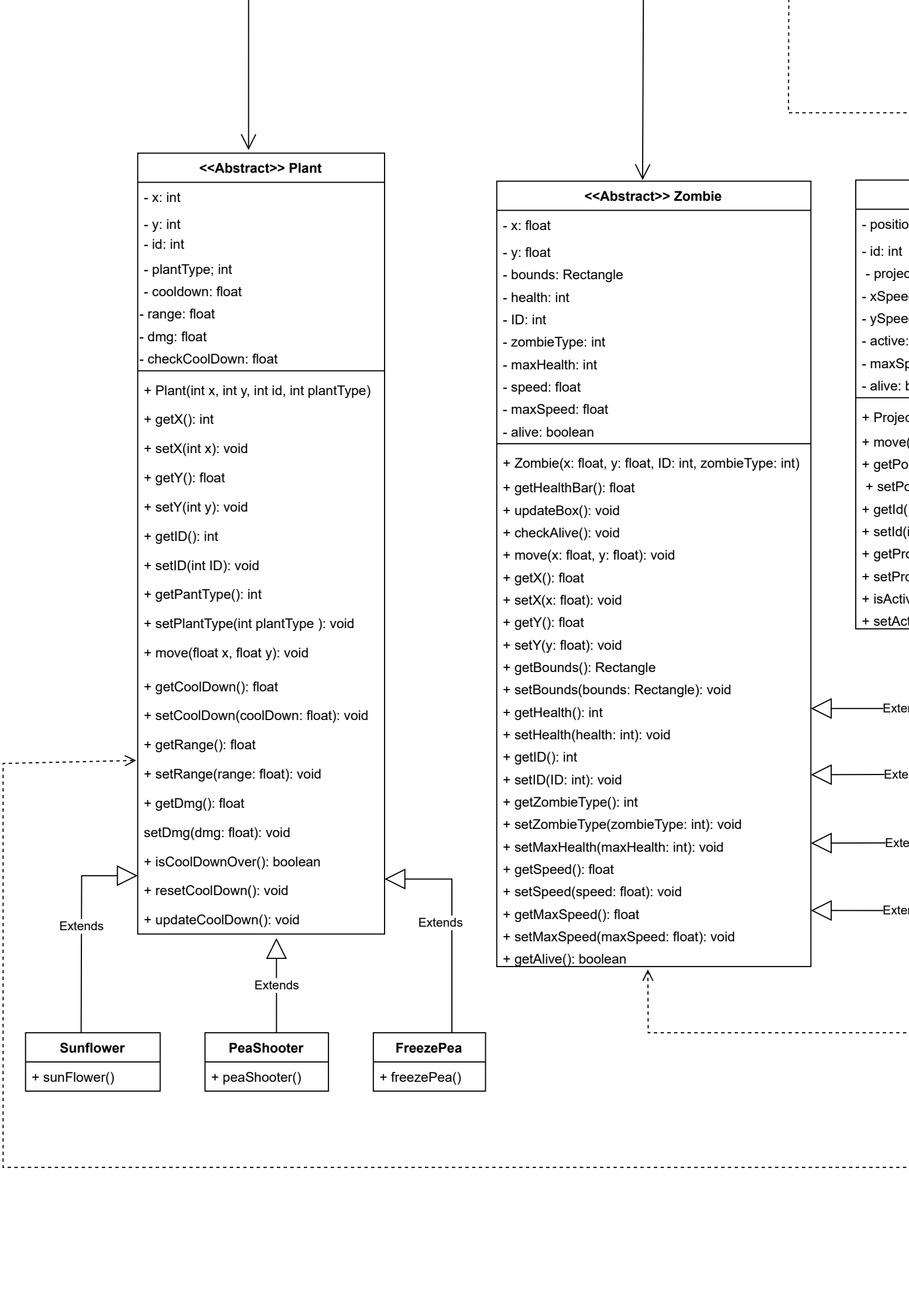
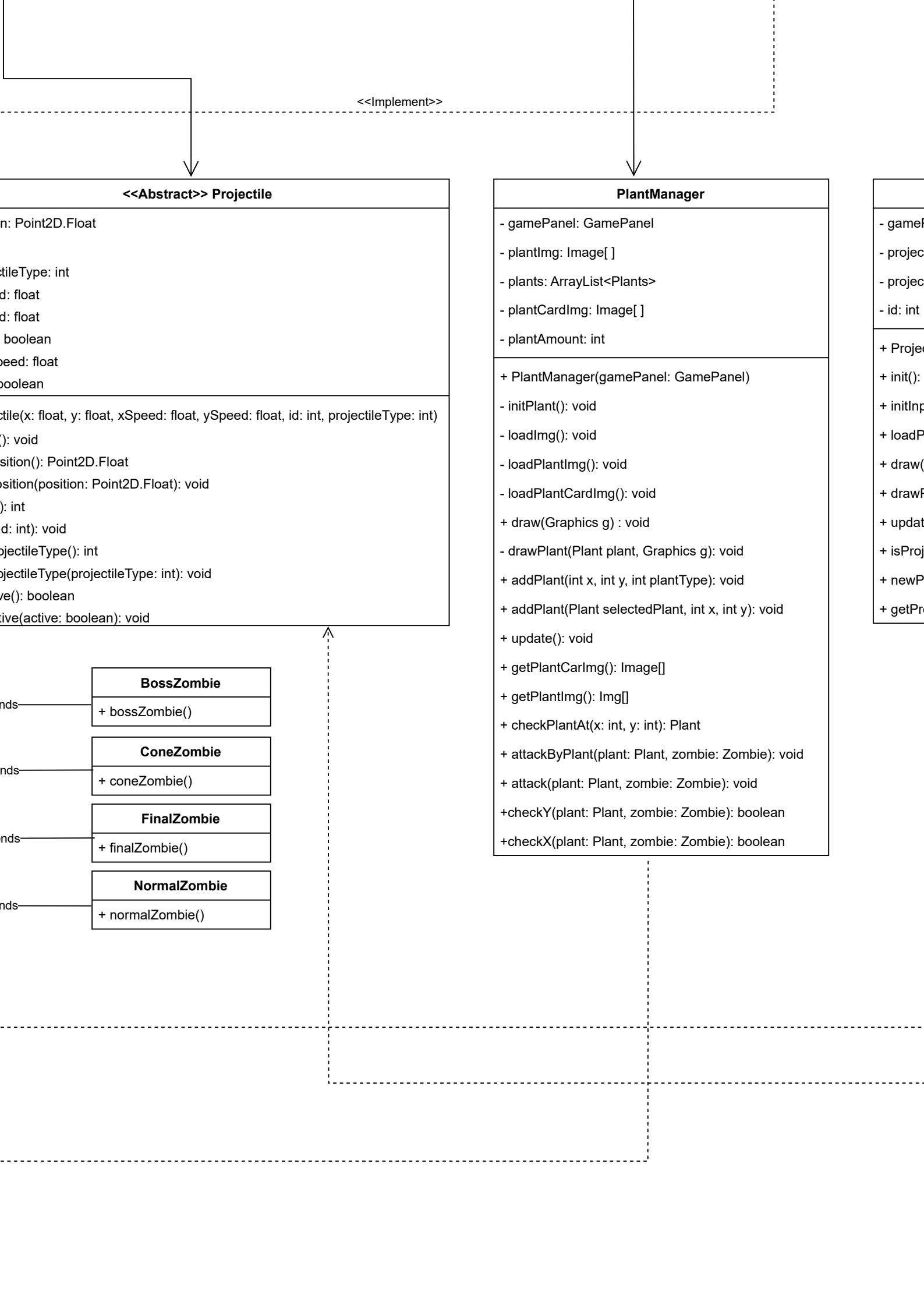
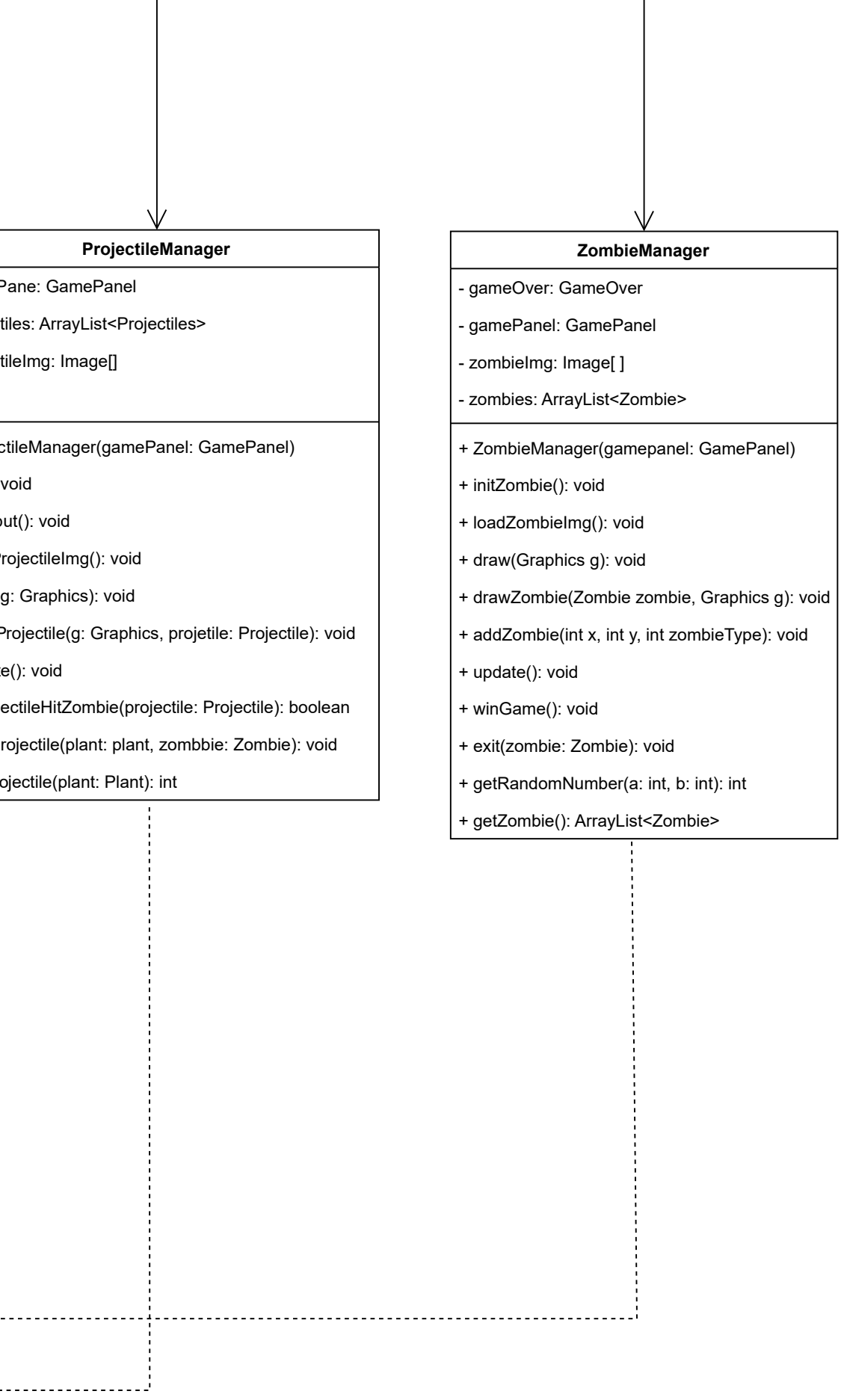

Game
- setUPS: double
- setFPS: double
- widthSize: int
+ init()
- gameTheard: Thread
- gamePanel: GamePanel
- gameWin: GameWin
- jLabel: JLabel
- myMouseListener: MyMouseListener
- myKeyBoardListener: MyKeyBoardListener
+ main(String[]): void
+ game(GamePanel)
+ init(): void
+ start(): void
+ updateGame(): void
+ run(): void
+ initInput(): void
+ getGamePanel(): GamePanel
+ getSetFPS(): double
+ setSetFPS(double0: void
+ getSetUPS(): double
+ setSetUPS(double0: void
+ getWidthSizw(): int
+setWidthSize(int): void
+ getHeightSize(): int
+ setHeightSize(int): void
+ getgameThread(): Thread
+ setGameThread(Thread): void
+ setGamePnel(GamePanel): void
+ getMyMouseListener(): MyMouseListener
+ setMyMouseListener(MyMouseListener): void
+ getMyKeyBoardListener(): MyKeyBoardListener
+ setMyKeyBoardListener(MyKeyBoardListener): void
+ winGame(): void
1
GamePanel
- game: Game
- jLabel: JLabel
- zombieManager: ZombieManager
- plantManager: PlantManager
- projectileManager: ProjectileManage
- backGround: Image
- keyBoardListener: MyKeyBoardListen
- mouseListener: MyMouseListener
- cardBar: CardBar
- selectedPlant: Plant
- mouseX: int
- mouseY: int
+ GamePanel()
+ init(): void
+ setBackGround(Graphics g): void
+ render(Graphics g): void
+ drawSelectedPlant(Graphics g): void
+ paintComponent(Graphics g): void
+ updateGame(): void
+ initInput(): void
+ getPlantManager(): PlantManager
+ setSelectedPlant(Plant selectedPlant
+ getMouseX(): int
+ setMouseX(int): void
+ getMouseY(): int
+ setMouseY(int): void
+ mouseClicked(int x, int y): void
+ mousePressed(int x, int y): void
+ mouseReleased(int x, int y): void
+ mouseDragged(int x, int y): void
+ mouseMoved(int x, int y): void
+ mouseEntered(int x, int y): void
+ mouseOver(int x, int y): void
6
+ mouseExited(int x, int y): void
+ getZombieManager(): ZombieManag
5
+ shootZombie(Plant, Zombie): void
+ getGame(): Game


MenuView
- contentPanel: JPanel
- textField_1: JTextField
- comingSoon: ComingSoon
- gameWindow: Game
- gamePanel: GamePanel
+ MenuView()
+ startAdventure(): void
+ comingSoon(): void
+ optionTab(): void
+ helpTab(): void
+ exitTab(): void
CardBar
- x: int
- y: int
- width: int
- height: int
- plantButtons: MyButton[ ]
- plantManager: PlantManager
- gamePanel: GamePanel
- selectedPlant: Plant
- displayPlant: Plant
- w: int
- h: int
+ CardBar(x: int, y: int, width: int, height: int, gamePanel: GamePanel)
+ initButtons(): void
+ draw(g: Graphics): void
+ drawDisplayPlant(g: Graphics): void
+ drawDisplayPlantRange(g: Graphics): void
+ drawButton(g: Graphics): void
+ getX()
+ setX(x: int): void
+ getY()
+ setY(y: int): void
+ getWidth(): int
+ setWidth(width: int): void
+ getHeight(): int
+ setHeight(height: int): void
+ getPlantButtons(): MyButton[]
+ setPlantButtons(plantButtons: MyButton[]): void
+ getPlantManager(): PlantManager
+ setPlantManager(plantManager: PlantManager)
+ getGamePanel(): GamePanel
+ setGamePanel(gamePanel: GamePanel): void
+ mousePressed(x: int,y: int): void
+ mouseOver(x: int,y: int): void
+ mouseReleased(x: int,y: int): void
+ mouseEntered(x: int,y: int): void
+ mouseExited(x: int,y: int): void
+ mouseDragged(x: int,y: int): void
+ mouseMoved(x: int,y: int): void
+ displayPlant(p: Plant): void
ComingSoon
- contentPanel: Jpanel
+ ComingSoon()
+ backToMenu()
GameOver
- contentPanel: Jpanel
+ GameOver()
+ backtoMenu(): void
GameWin
- contentPanel: Jpanel
+ GameWin()
+ backtoMenu(): void
IMouse
+ mouseClicked(int, int)
+ mousePressed(int, int)
+ mouseOver(int, int)
+ mouseReleased(int, int)
+ mouseEntered(int, int)
+ mouseExited(int, int)
+ mouseDragged(int, int)
+ mouseMoved(int, int)
1
1
1
1
1
1
2
3
4


MyBotton
- x: int
- y: int
- width: int
- height: int
- text: String
- bounds: Rectangle
- mouseOver: boolean
- mousePress: boolean
+ ,ybutton(texxt: String, x: int, y: int, width: int, height: int)
+ initBounds(): void
+ draw(g: Graphics): void
+ drawBody(g: Graphics): void
+ drawBorder(g: Graphics): void
+ drawText(g: Graphics): void
+ getBound(): void
+ resetBound(): Rectangle
+ isMouseOver(): boolean
+ setMouseOver(mouse Over: boolean): void
+ isMousePress(): boolean
+ setMousePress(mousePress: boolean): void
Extends
: void


<<Abstract>> Plant
- x: int
- y: int
- id: int
- plantType; int
- cooldown: float
- range: float
- dmg: float
- checkCoolDown: float
+ Plant(int x, int y, int id, int plantType)
+ getX(): int
+ setX(int x): void
+ getY(): float
+ setY(int y): void
+ getID(): int
+ setID(int ID): void
+ getPantType(): int
+ setPlantType(int plantType): void
+ move(float x, float y): void
+ getCoolDown(): float
+ setCoolDown(coolDown: float): void
+ getRange(): float
+ setRange(range: float): void
+ getDmg(): float
setDmg(dmg: float): void
+ isCoolDownOver(): boolean
+ resetCoolDown(): void
+ updateCoolDown(): void
<<Abstract>> Zombie
- x: float
- y: float
- bounds: Rectangle
- health: int
- ID: int
- zombieType: int
- maxHealth: int
- speed: float
- maxSpeed: float
- alive: boolean
+ Zombie(x: float, y: float, ID: int, zombieType: int)
+ getHealthBar(): float
+ updateBox(): void
+ checkAlive(): void
+ move(x: float, y: float): void
+ getX(): float
+ setX(x: float): void
+ getY(): float
+ setY(y: float): void
+ getBounds(): Rectangle
+ setBounds(bounds: Rectangle): void
+ getHealth(): int
+ setHealth(health: int): void
+ getID(): int
+ setID(ID: int): void
+ getZombieType(): int
+ setZombieType(zombieType: int): void
+ setMaxHealth(maxHealth: int): void
+ getSpeed(): float
+ setSpeed(speed: float): void
+ getMaxSpeed(): float
+ setMaxSpeed(maxSpeed: float): void
+ getAlive(): boolean
Extends
Extends
Extends
Sunflower
+ sunFlower()
PeaShooter
+ peaShooter()
FreezePea
+ freezePea()


<<Implement>>
<<Abstract>> Projectile
n: Point2D.Float
ctileType: int
d: float
d: float
boolean
peed: float
boolean
ctile(x: float, y: float, xSpeed: float, ySpeed: float, id: int, projectileType: int)
(): void
sition(): Point2D.Float
osition(position: Point2D.Float): void
): int
d: int): void
ojectileType(): int
ojectileType(projectileType: int): void
ve(): boolean
tive(active: boolean): void
PlantManager
- gamePanel: GamePanel
- plantImg: Image[ ]
- plants: ArrayList<Plants>
- plantCardImg: Image[ ]
- plantAmount: int
+ PlantManager(gamePanel: GamePanel)
- initPlant(): void
- loadImg(): void
- loadPlantImg(): void
- loadPlantCardImg(): void
+ draw(Graphics g) : void
- drawPlant(Plant plant, Graphics g): void
+ addPlant(int x, int y, int plantType): void
+ addPlant(Plant selectedPlant, int x, int y): void
+ update(): void
+ getPlantCarImg(): Image[]
+ getPlantImg(): Img[]
+ checkPlantAt(x: int, y: int): Plant
+ attackByPlant(plant: Plant, zombie: Zombie): void
+ attack(plant: Plant, zombie: Zombie): void
+checkY(plant: Plant, zombie: Zombie): boolean
+checkX(plant: Plant, zombie: Zombie): boolean
BossZombie
+ bossZombie()
ConeZombie
+ coneZombie()
FinalZombie
+ finalZombie()
NormalZombie
+ normalZombie()


ProjectileManager
Pane: GamePanel
tiles: ArrayList<Projectiles>
tileImg: Image[]
ctileManager(gamePanel: GamePanel)
void
out(): void
rojectileImg(): void
g: Graphics): void
Projectile(g: Graphics, projetile: Projectile): void
te(): void
ectileHitZombie(projectile: Projectile): boolean
rojectile(plant: plant, zombbie: Zombie): void
ojectile(plant: Plant): int
ZombieManager
- gameOver: GameOver
- gamePanel: GamePanel
- zombieImg: Image[ ]
- zombies: ArrayList<Zombie>
+ ZombieManager(gamepanel: GamePanel)
+ initZombie(): void
+ loadZombieImg(): void
+ draw(Graphics g): void
+ drawZombie(Zombie zombie, Graphics g): void
+ addZombie(int x, int y, int zombieType): void
+ update(): void
+ winGame(): void
+ exit(zombie: Zombie): void
+ getRandomNumber(a: int, b: int): int
+ getZombie(): ArrayList<Zombie>