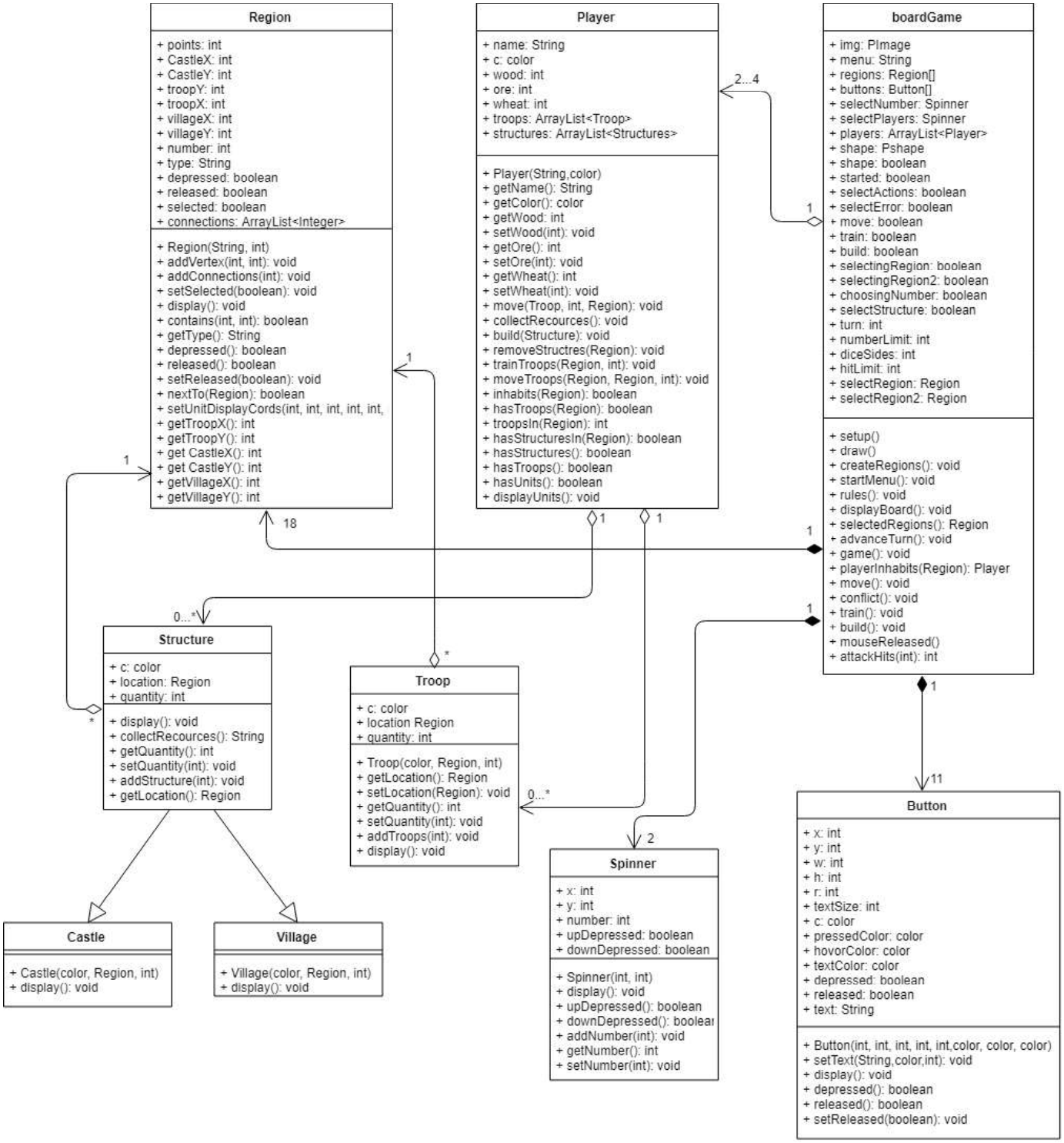
UML Diagram


Region
+ points: int
+ CastleX: int
+ CastleY: int
+ troopY: int
+ troopX: int
+ villageX: int
+ villageY: int
+ number: int
+ type: String
+ depressed: boolean
+ released: boolean
+ selected: boolean
+ connections: ArrayList<Integer>
+ Region(String, int)
+ addVertex(int, int): void
+ addConnections(int): void
+ setSelected(boolean): void
+ display(): void
+ contains(int, int): boolean
+ getType(): String
+ depressed(): boolean
+ released(): boolean
+ setReleased(boolean): void
+ nextTo(Region): boolean
+ setUnitDisplayCords(int, int, int, int, int,
+ getTroopX(): int
+ getTroopY(): int
+ get CastleX(): int
+ get CastleY(): int
+ getVillageX(): int
+ getVillageY(): int
Player
+ name: String
+ c: color
+ wood: int
+ ore: int
+ wheat: int
+ troops: ArrayList<Troop>
+ structures: ArrayList<Structures>
+ Player(String,color)
+ getName(): String
+ getColor(): color
+ getWood: int
+ setWood(int): void
+ getOre(): int
+ setOre(int): void
+ getWheat(): int
+ setWheat(int): void
+ move(Troop, int, Region): void
+ collectRecources(): void
+ build(Structure): void
+ removeStructres(Region): void
+ trainTroops(Region, int): void
+ moveTroops(Region, Region, int): void
+ inhabits(Region): boolean
+ hasTroops(Region): boolean
+ troopsIn(Region): int
+ hasStructuresIn(Region): boolean
+ hasStructures(): boolean
+ hasTroops(): boolean
+ hasUnits(): boolean
+ displayUnits(): void
boardGame
+ img: PImage
+ menu: String
+ regions: Region[]
+ buttons: Button[]
+ selectNumber: Spinner
+ selectPlayers: Spinner
+ players: ArrayList<Player>
+ shape: Pshape
+ shape: boolean
+ started: boolean
+ selectActions: boolean
+ selectError: boolean
+ move: boolean
+ train: boolean
+ build: boolean
+ selectingRegion: boolean
+ selectingRegion2: boolean
+ choosingNumber: boolean
+ selectStructure: boolean
+ turn: int
+ numberLimit: int
+ diceSides: int
+ hitLimit: int
+ selectRegion: Region
+ selectRegion2: Region
+ setup()
+ draw()
+ createRegions(): void
+ startMenu(): void
+ rules(): void
+ displayBoard(): void
+ selectedRegions(): Region
+ advanceTurn(): void
+ game(): void
+ playerInhabits(Region): Player
+ move(): void
+ conflict(): void
+ train(): void
+ build(): void
+ mouseReleased()
+ attackHits(int): int
Structure
+ c: color
+ location: Region
+ quantity: int
+ display(): void
+ collectRecources(): String
+ getQuantity(): int
+ setQuantity(int): void
+ addStructure(int): void
+ getLocation(): Region
Troop
+ c: color
+ location Region
+ quantity: int
+ Troop(color, Region, int)
+ getLocation(): Region
+ setLocation(Region): void
+ getQuantity(): int
+ setQuantity(int): void
+ addTroops(int): void
+ display(): void
Castle
+ Castle(color, Region, int)
+ display(): void
Village
+ Village(color, Region, int)
+ display(): void
Spinner
+ x: int
+ y: int
+ number: int
+ upDepressed: boolean
+ downDepressed: boolean
+ Spinner(int, int)
+ display(): void
+ upDepressed(): boolean
+ downDepressed(): boolean
+ addNumber(int): void
+ getNumber(): int
+ setNumber(int): void
Button
+ x: int
+ y: int
+ w: int
+ h: int
+ r: int
+ textSize: int
+ c: color
+ pressedColor: color
+ hovorColor: color
+ textColor: color
+ depressed: boolean
+ released: boolean
+ text: String
+ Button(int, int, int, int, int,color, color, color)
+ setText(String,color,int): void
+ display(): void
+ depressed(): boolean
+ released(): boolean
+ setReleased(boolean): void
2...4
1
18
0...*
*
11
2