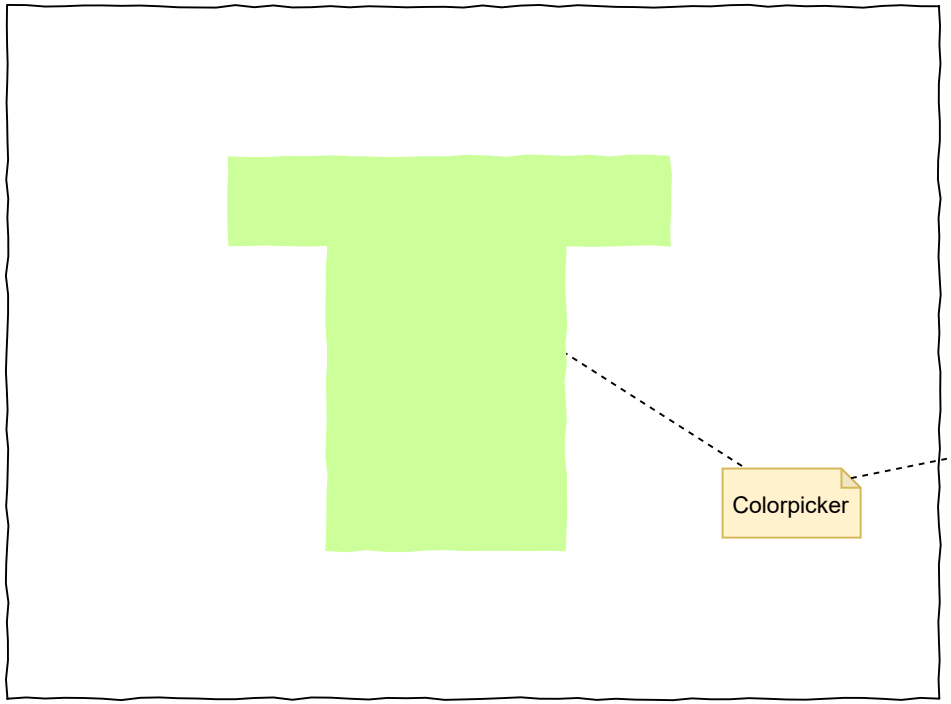
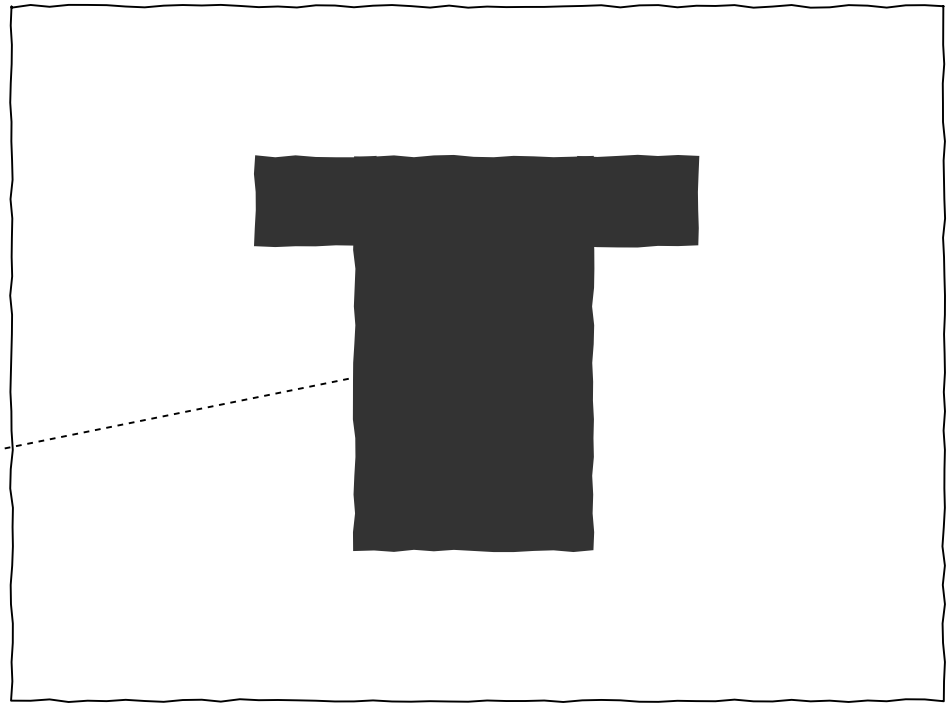
Wireframe

Team Bunt



Team Schwarz-Weiß



Weiter

Fifa auf Wish bestellt

Löschen des Spielers
Click

Pausieren um Spieler auszuwechseln
Click

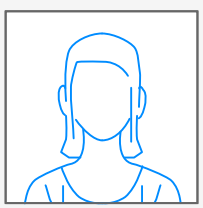
Time

Spieler 1
Spieler 2
Spieler 3
Spieler 5
Spieler 6
Spieler 7
Spieler 8
Spieler 9
Spieler 10
Spieler 11
Spieler 12
Spieler 13
Spieler 14
Spieler 15

Klick and Klick
Click

Spieler 4

Spieler 1 (x15)



Name:
Präzision:
Geschwindigkeit:

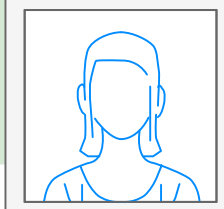
Ballbesitz

Start the Game

Start-Button

3:0

Spielstand



Name:
Präzision:
Geschwindigkeit:

Spielerinfo bei Hover

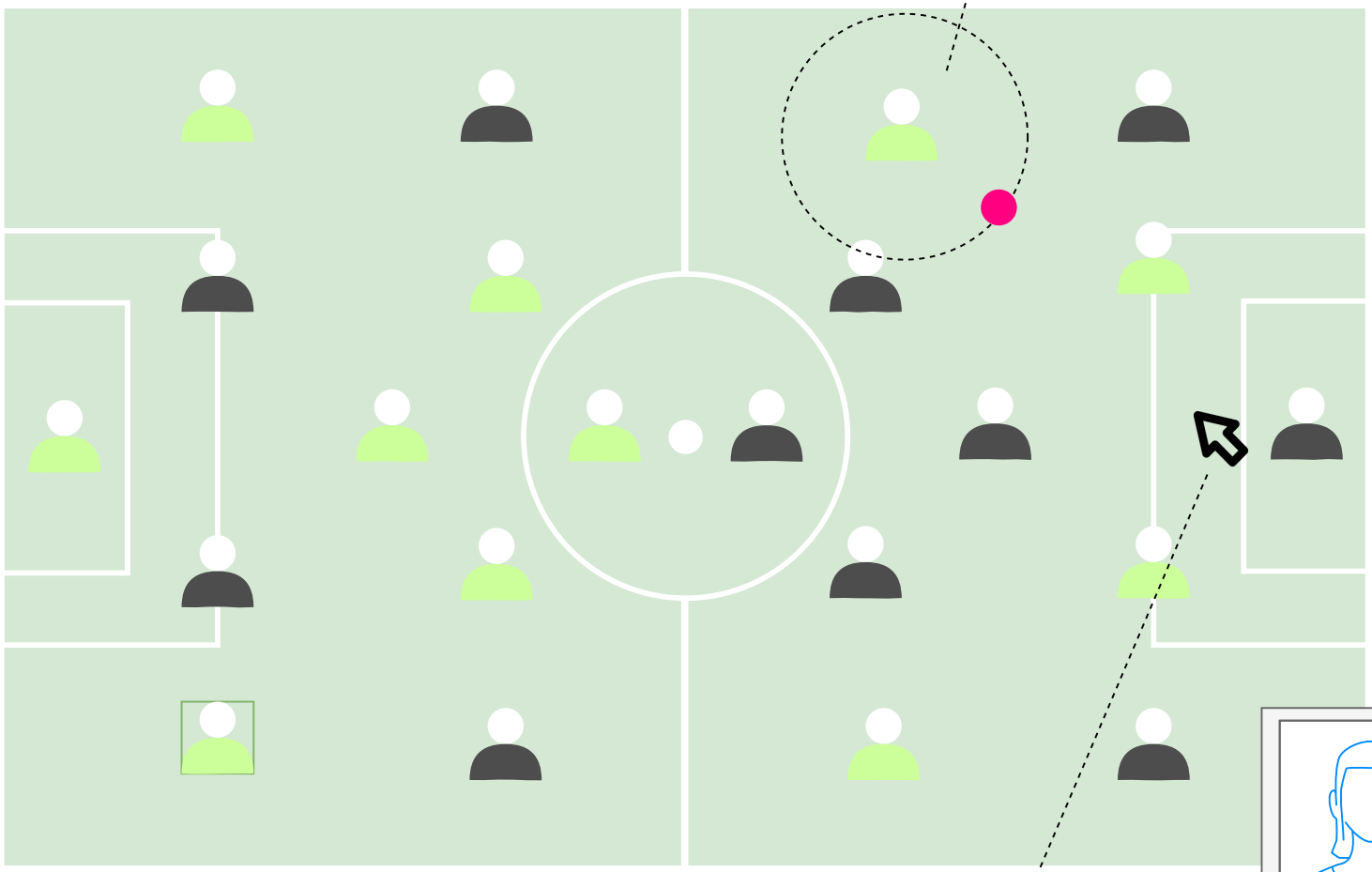
Fifa auf Wish bestellt

Pausieren und Weiter Button (Um Spieler auszuwechseln)

Time

Spieler bewegt sich zum Ball

Spieler 1
Spieler 2
Spieler 3
Spieler 4
Spieler 5
Spieler 6
Spieler 7
Spieler 8
Spieler 9
Spieler 10
Spieler 11
Spieler 12
Spieler 13
Spieler 14
Spieler 15



Spieler 1
Spieler 1
Spieler 1
Spieler 1
Spieler 1
Spieler 1
Spieler 1
Spieler 1
Spieler 1
Spieler 1
Spieler 1
Spieler 1
Spieler 1
Spieler 1

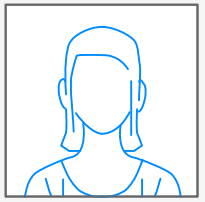
Name:
Präzision:
Geschwindigkeit:

Ballbesitz

Spielstand

3:0

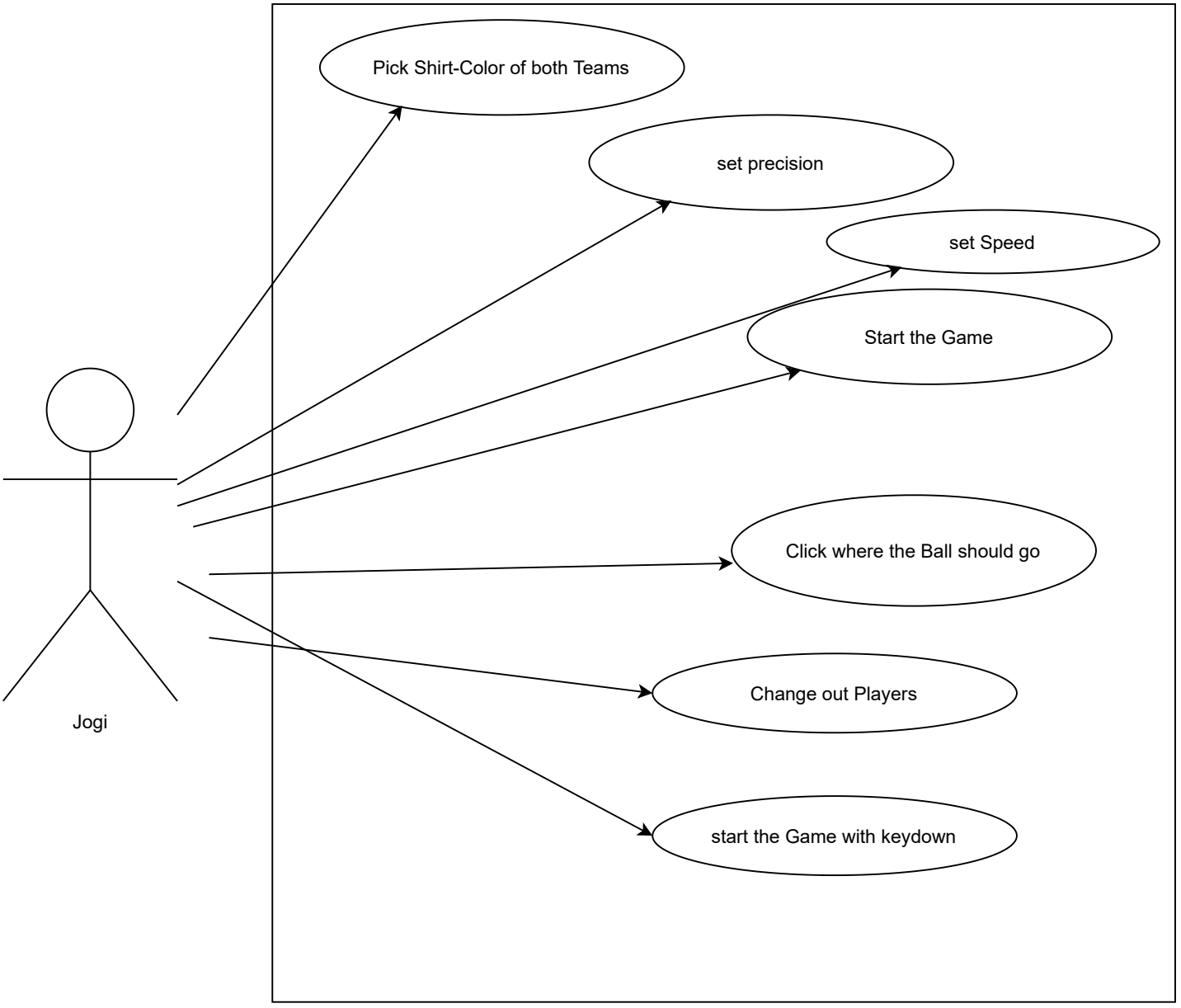
Spieler schießt, wo User hinklickt

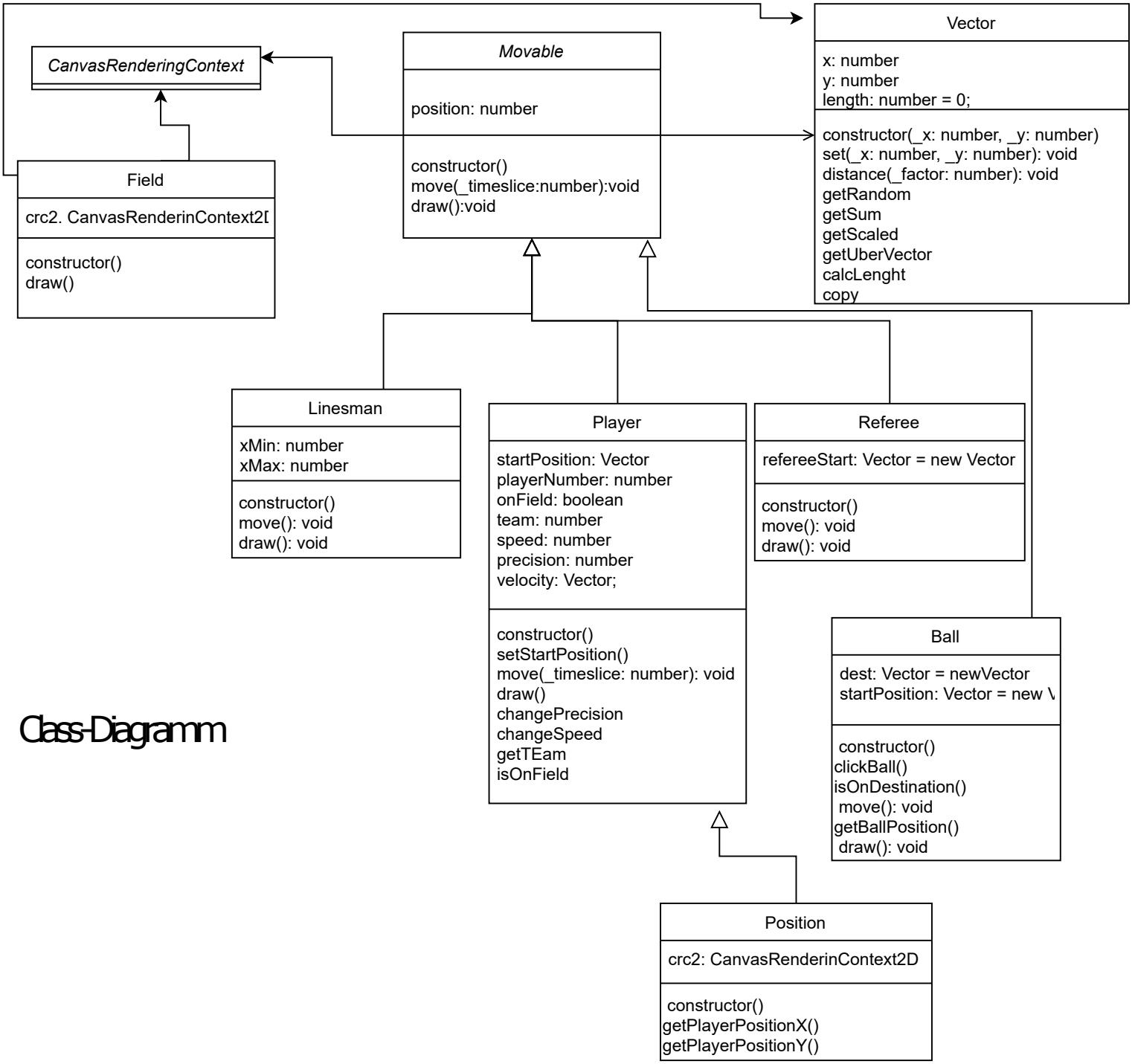


Name:
Präzision:
Geschwindigkeit:

Spielerinfo bei Hover

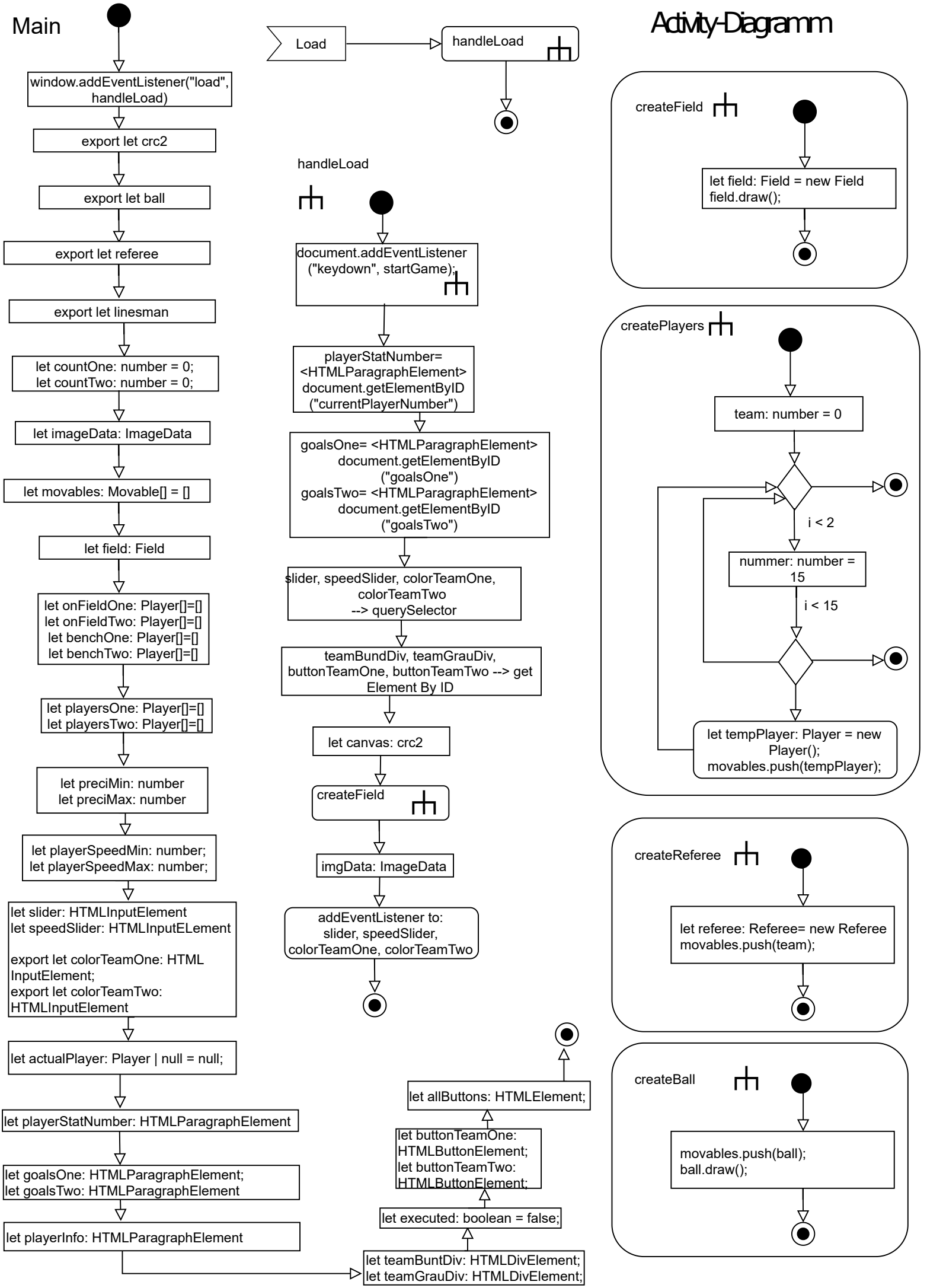
Use-Case_Diagramm



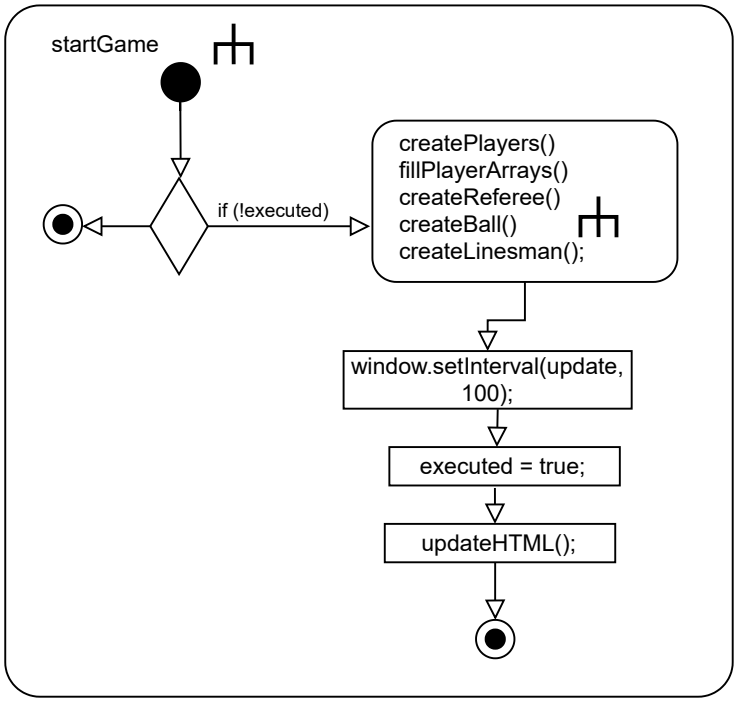


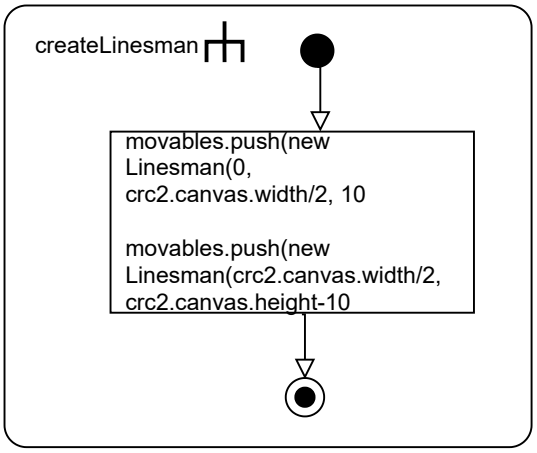
Class-Diagramm

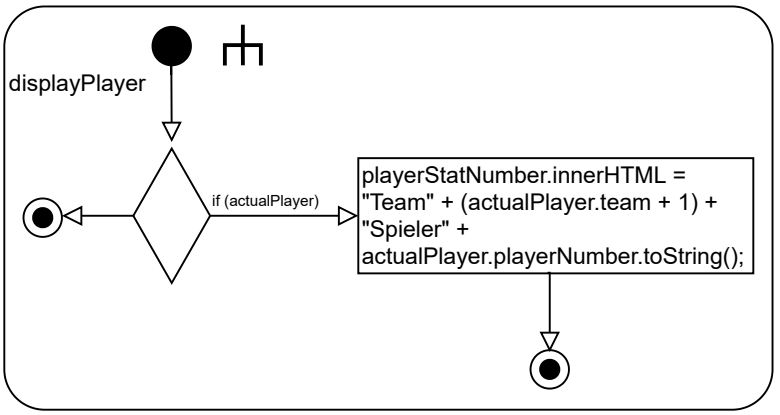
Activity-Diagramm

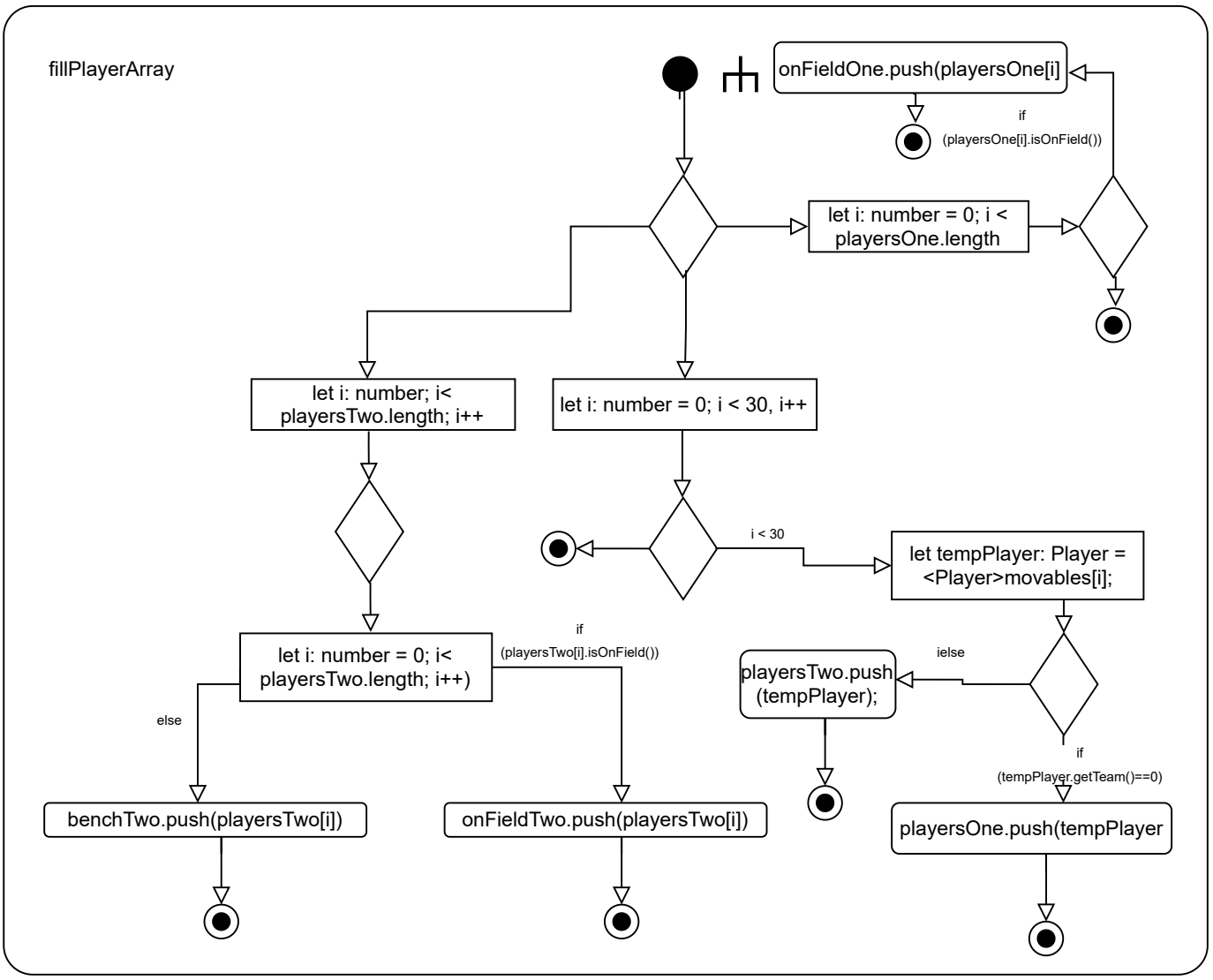


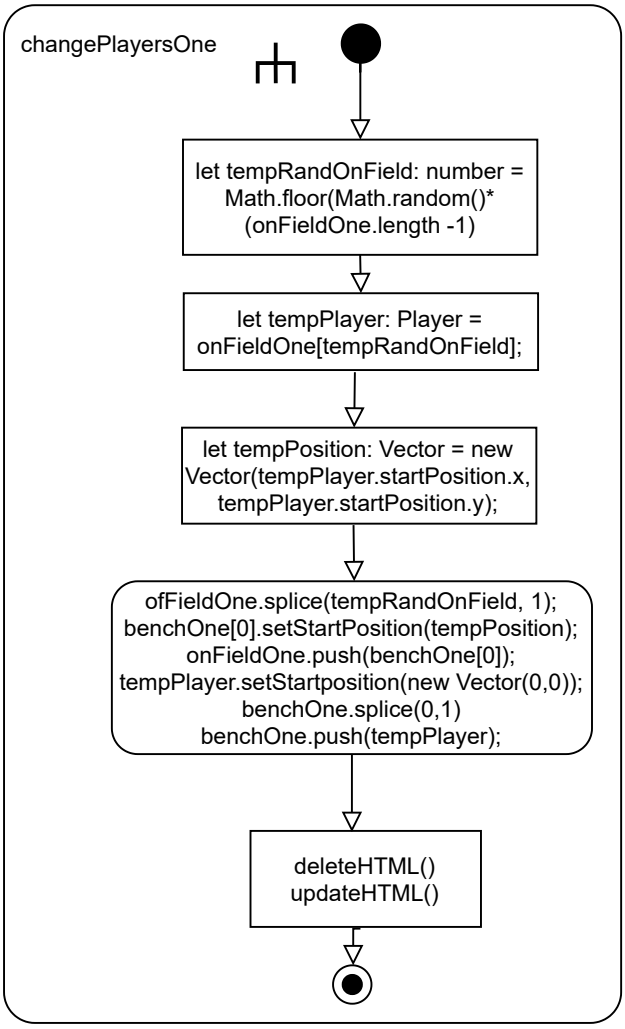
Main 2

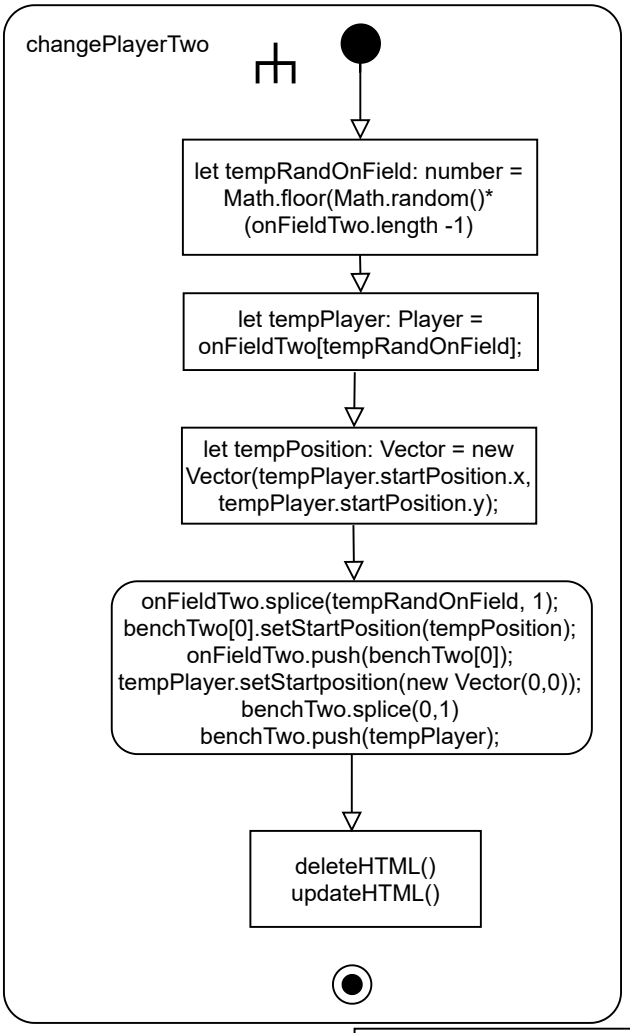

startGame
if (!executed)
createPlayers()
fillPlayerArrays()
createReferee()
createBall()
createLinesman();
window.setInterval(update, 100);
executed = true;
updateHTML();

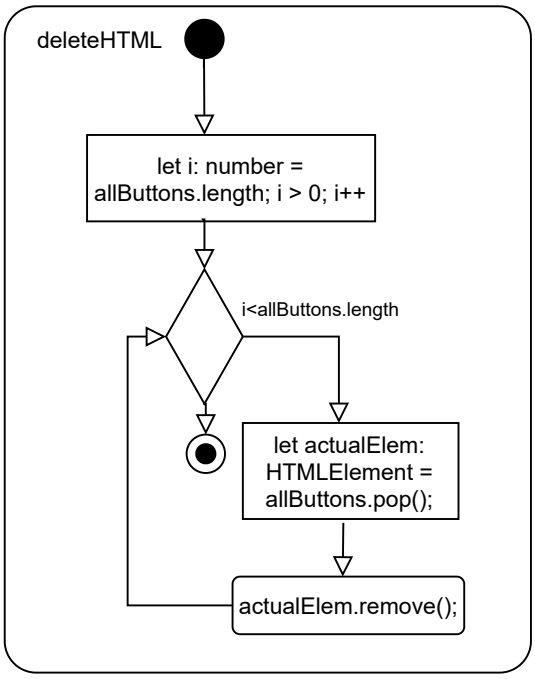

createLinesman
movables.push(new Linesman(0, crc2.canvas.width/2, 10
movables.push(new Linesman(crc2.canvas.width/2, crc2.canvas.height-10

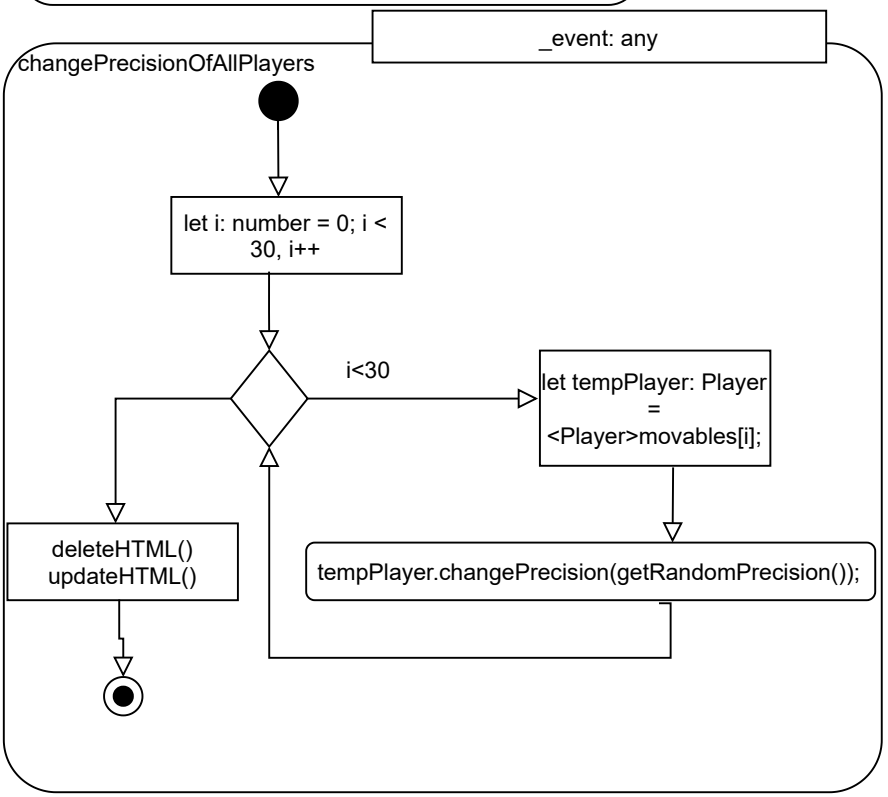

displayPlayer
if (actualPlayer)
playerStatNumber.innerHTML = "Team" + (actualPlayer.team + 1) + "Spieler" + actualPlayer.playerNumber.toString();

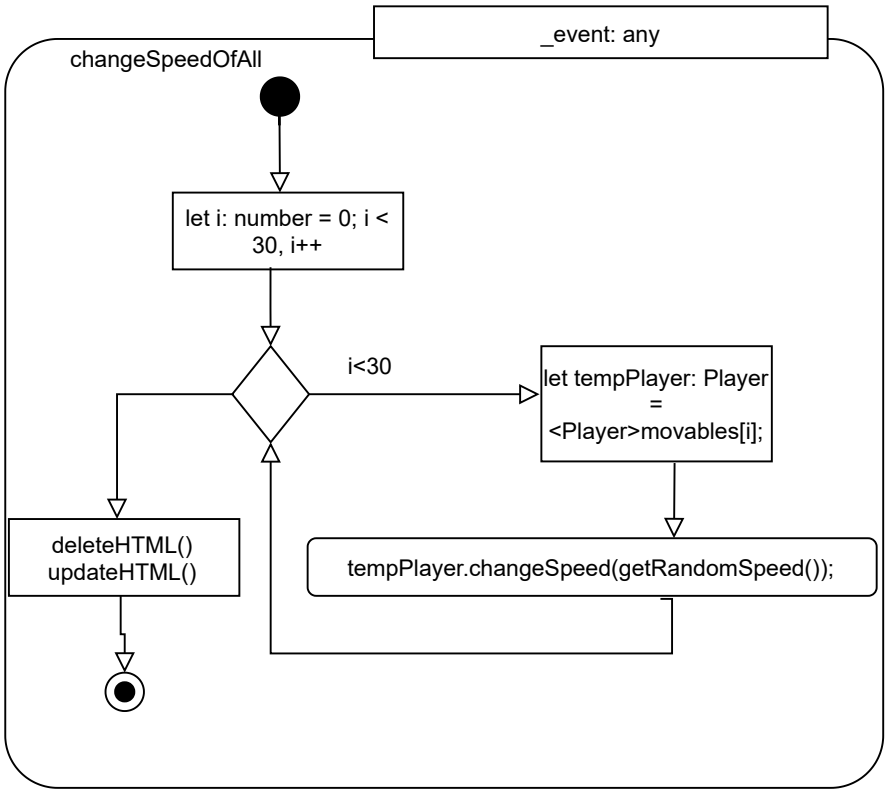

fillPlayerArray
onFieldOne.push(playersOne[i]
if (playersOne[i].isOnField())
let i: number = 0; i < playersOne.length
let i: number; i< playersTwo.length; i++
let i: number = 0; i < 30, i++
i < 30
let tempPlayer: Player = <Player>movables[i];
let i: number = 0; i< playersTwo.length; i++)
if (playersTwo[i].isOnField())
else
ielse
playersTwo.push (tempPlayer);
if (tempPlayer.getTeam()==0)
benchTwo.push(playersTwo[i])
onFieldTwo.push(playersTwo[i])
playersOne.push(tempPlayer

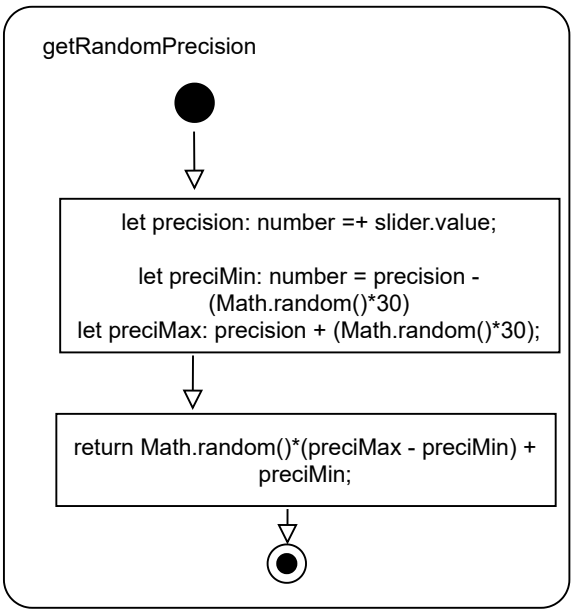

changePlayersOne
let tempRandOnField: number = Math.floor(Math.random()* (onFieldOne.length -1)
let tempPlayer: Player = onFieldOne[tempRandOnField];
let tempPosition: Vector = new Vector(tempPlayer.startPosition.x, tempPlayer.startPosition.y);
ofFieldOne.splice(tempRandOnField, 1); benchOne[0].setStartPosition(tempPosition); onFieldOne.push(benchOne[0]); tempPlayer.setStartposition(new Vector(0,0)); benchOne.splice(0,1) benchOne.push(tempPlayer);
deleteHTML() updateHTML()

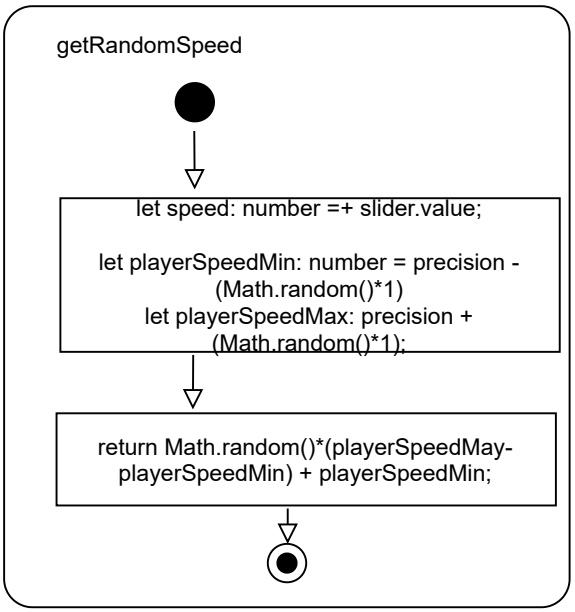

changePlayerTwo
let tempRandOnField: number = Math.floor(Math.random()* (onFieldTwo.length -1)
let tempPlayer: Player = onFieldTwo[tempRandOnField];
let tempPosition: Vector = new Vector(tempPlayer.startPosition.x, tempPlayer.startPosition.y);
onFieldTwo.splice(tempRandOnField, 1); benchTwo[0].setStartPosition(tempPosition); onFieldTwo.push(benchTwo[0]); tempPlayer.setStartposition(new Vector(0,0)); benchTwo.splice(0,1) benchTwo.push(tempPlayer);
deleteHTML() updateHTML()

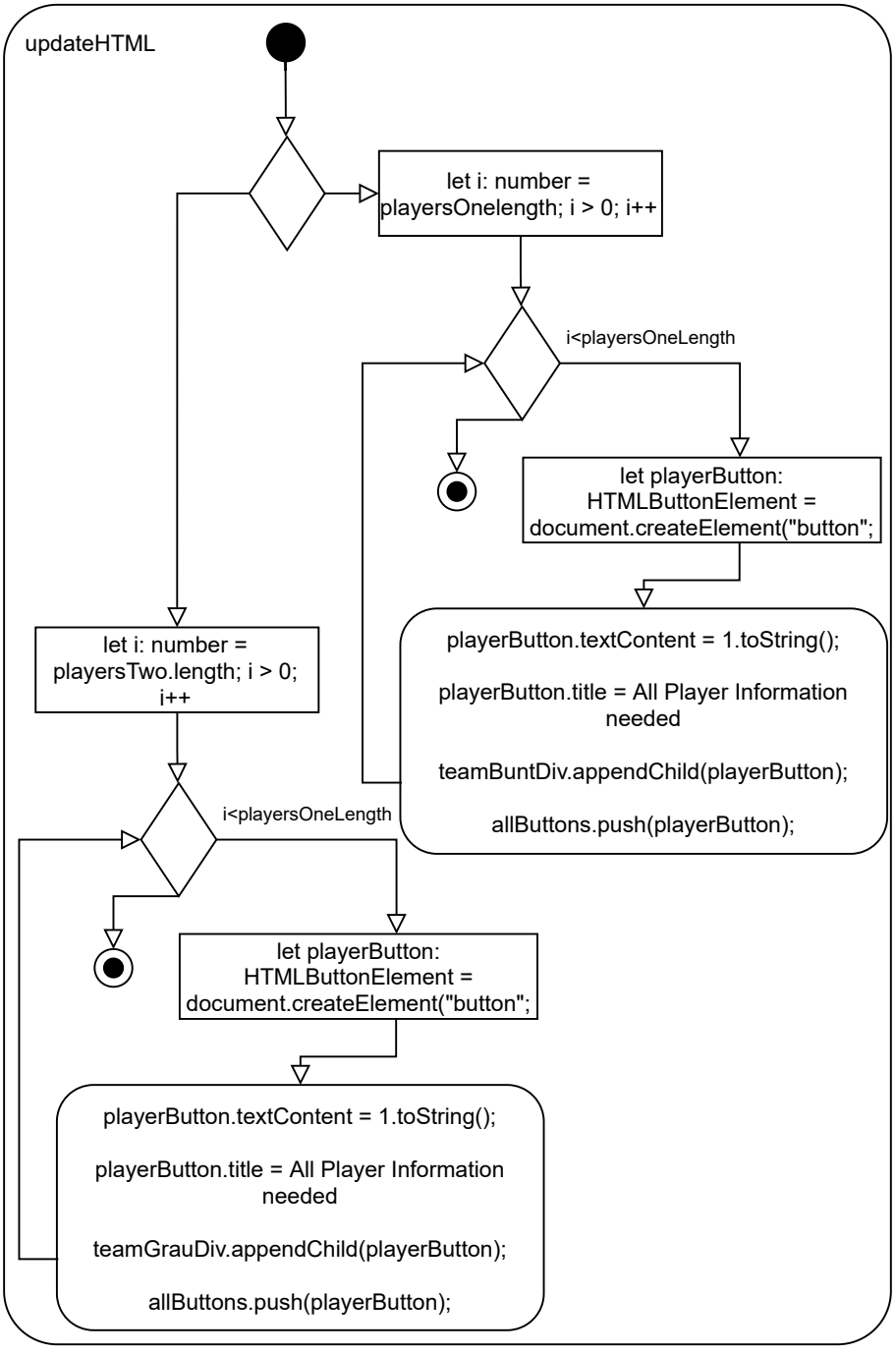

deleteHTML
let i: number = allButtons.length; i > 0; i++
i<allButtons.length
let actualElem: HTMLElement = allButtons.pop();
actualElem.remove();

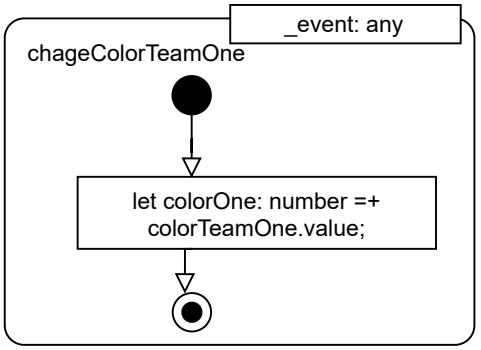

_event: any
changePrecisionOfAllPlayers
let i: number = 0; i < 30, i++
i<30
let tempPlayer: Player = <Player>movables[i];
tempPlayer.changePrecision(getRandomPrecision());
deleteHTML() updateHTML()

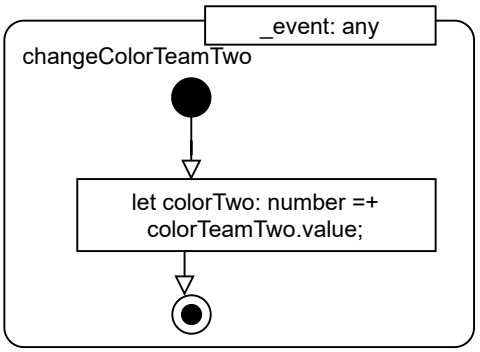

_event: any
changeSpeedOfAll
let i: number = 0; i < 30, i++
i<30
let tempPlayer: Player = <Player>movables[i];
deleteHTML()
updateHTML()
tempPlayer.changeSpeed(getRandomSpeed());

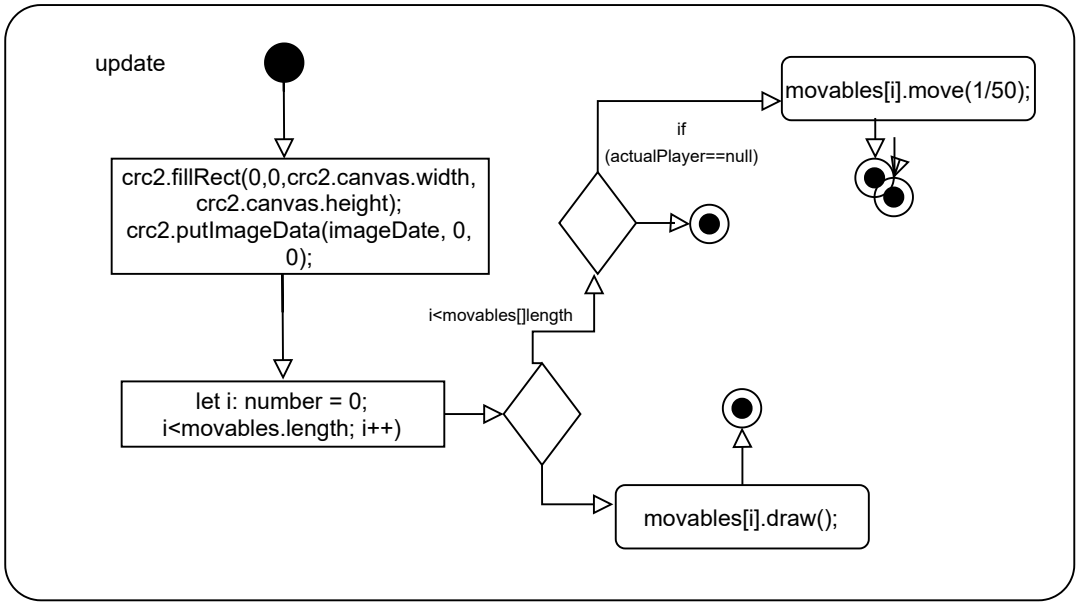

getRandomPrecision
let precision: number =+ slider.value;
let preciMin: number = precision - (Math.random()*30)
let preciMax: precision + (Math.random()*30);
return Math.random()*(preciMax - preciMin) + preciMin;


getRandomSpeed
let speed: number =+ slider.value;
let playerSpeedMin: number = precision - (Math.random()*1)
let playerSpeedMax: precision + (Math.random()*1);
return Math.random()*(playerSpeedMay-playerSpeedMin) + playerSpeedMin;

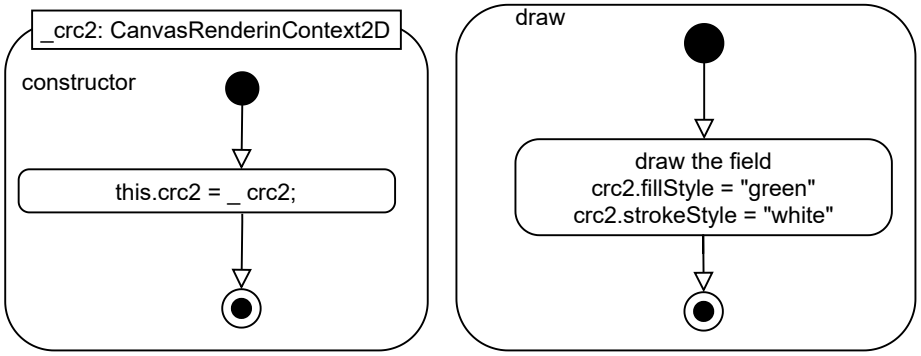

updateHTML
let i: number = playersOnelength; i > 0; i++
i<playersOneLength
let playerButton: HTMLButtonElement = document.createElement("button";
playerButton.textContent = 1.toString();
playerButton.title = All Player Information needed
teamBuntDiv.appendChild(playerButton);
allButtons.push(playerButton);
let i: number = playersTwo.length; i > 0; i++
i<playersOneLength
let playerButton: HTMLButtonElement = document.createElement("button";
playerButton.textContent = 1.toString();
playerButton.title = All Player Information needed
teamGrauDiv.appendChild(playerButton);
allButtons.push(playerButton);


_event: any
chageColorTeamOne
let colorOne: number =+ colorTeamOne.value;

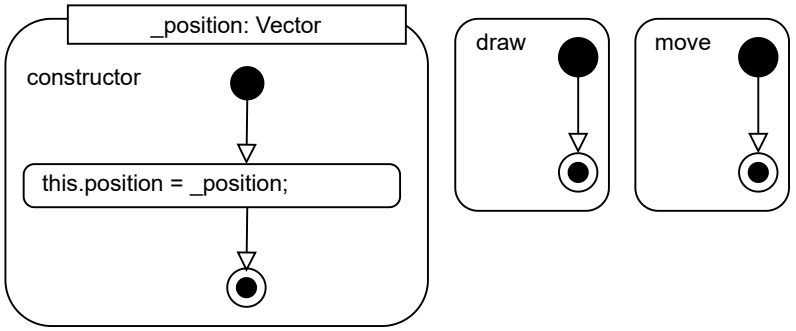

_event: any
changeColorTeamTwo
let colorTwo: number =+ colorTeamTwo.value;



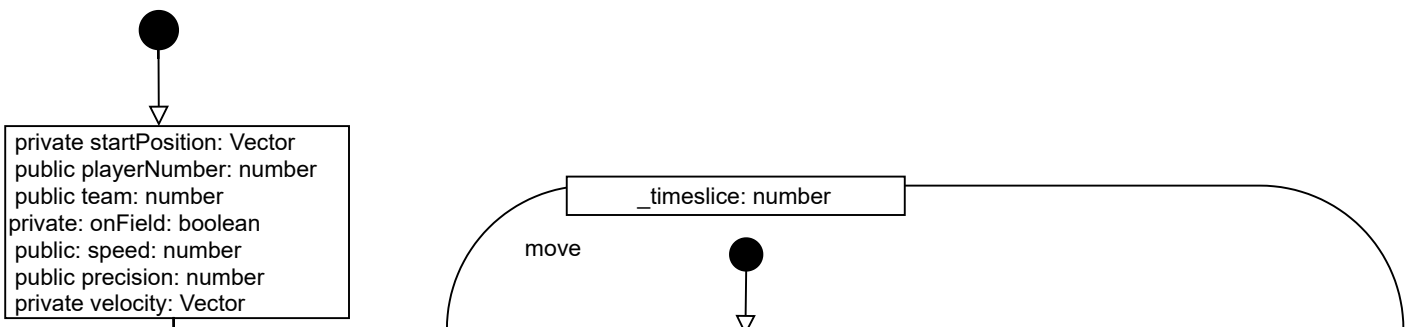
Field

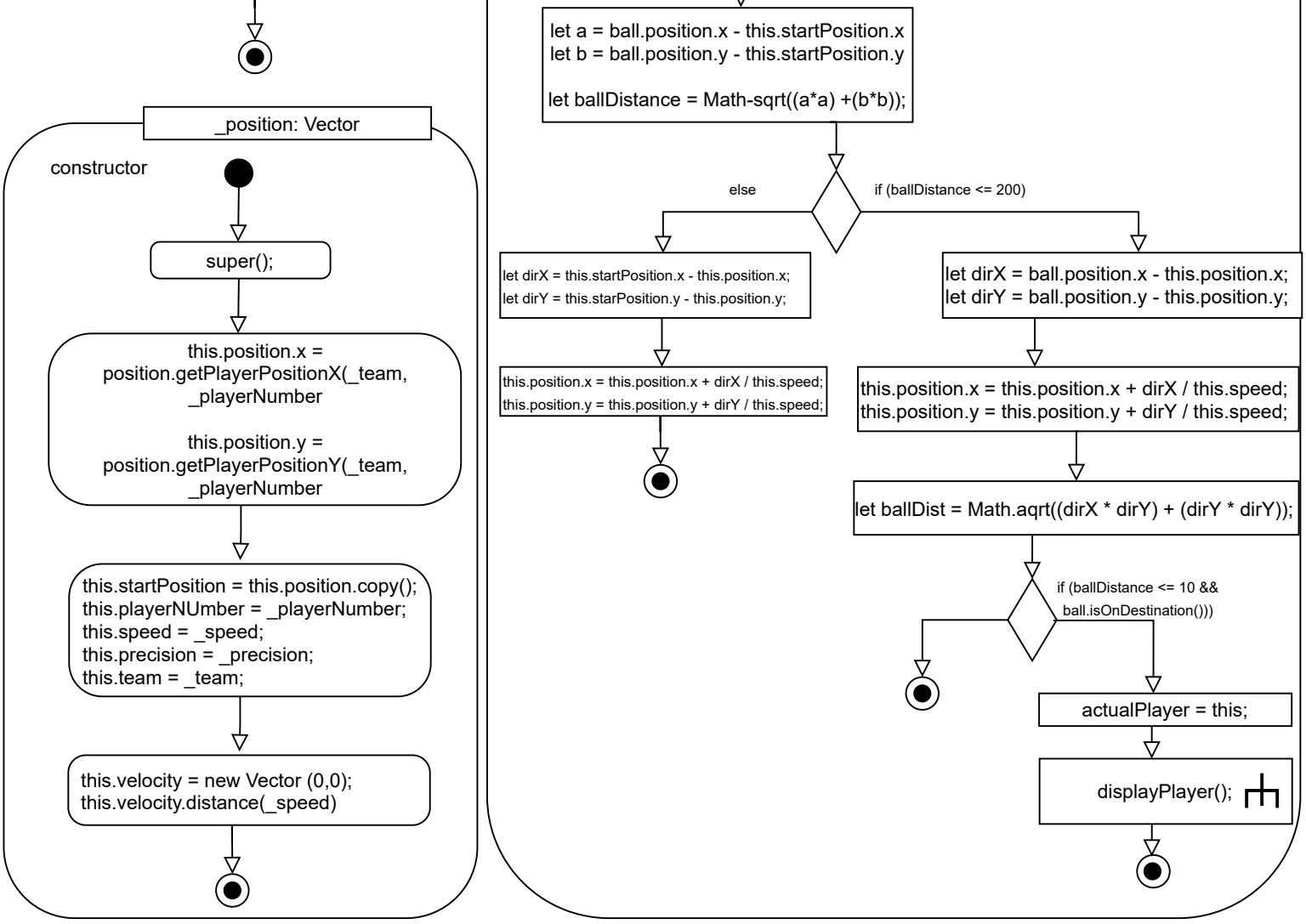


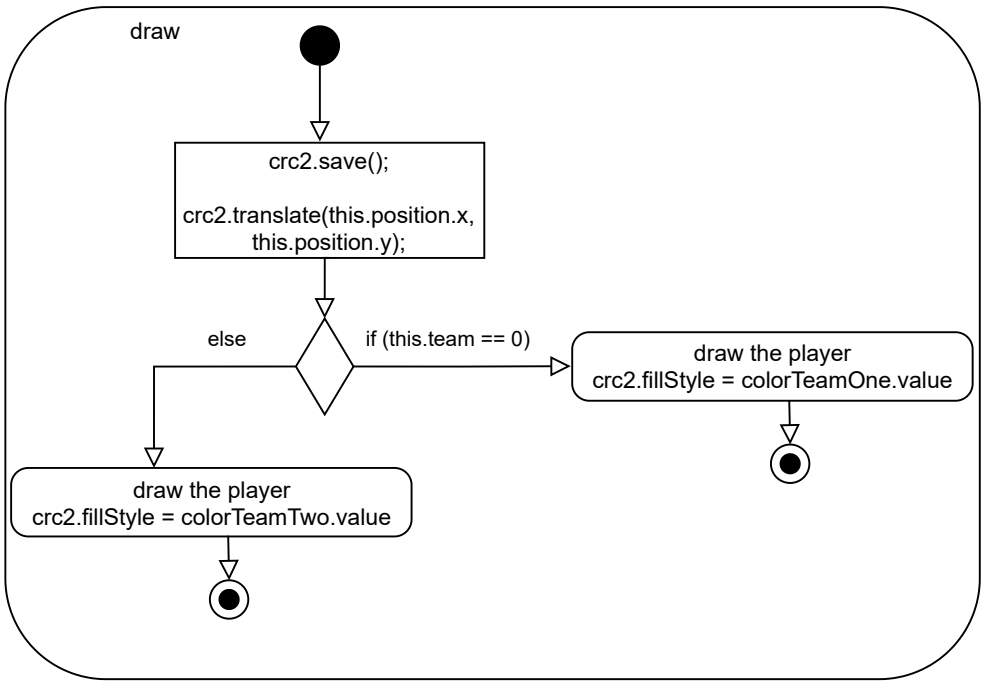
Movable

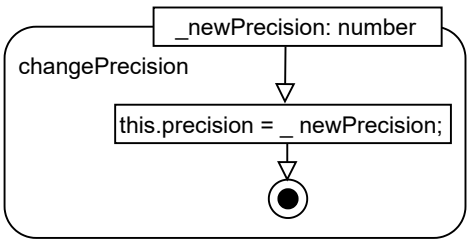


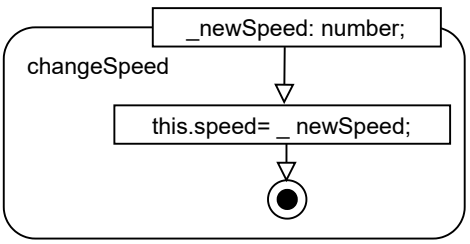
Player

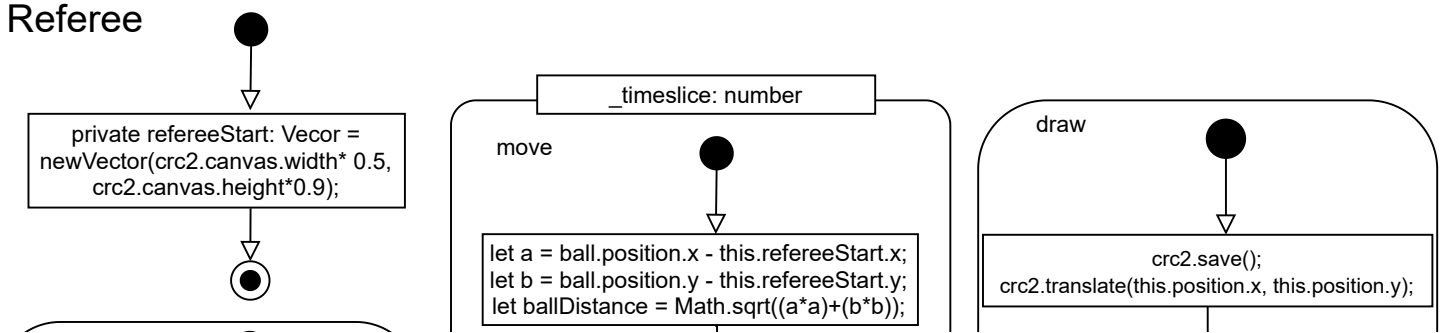


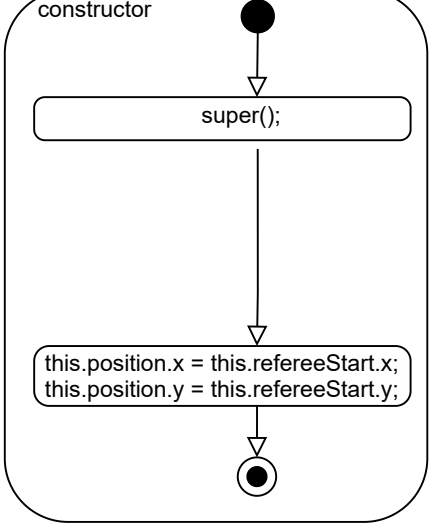

_position: Vector
constructor
super();
this.position.x =
position.getPlayerPositionX(_team,
_playerNumber
this.position.y =
position.getPlayerPositionY(_team,
_playerNumber
this.startPosition = this.position.copy();
this.playerNUmber = _playerNumber;
this.speed = _speed;
this.precision = _precision;
this.team = _team;
this.velocity = new Vector (0,0);
this.velocity.distance(_speed)
let a = ball.position.x - this.startPosition.x
let b = ball.position.y - this.startPosition.y
let ballDistance = Math-sqrt((a*a) +(b*b));
else
if (ballDistance <= 200)
let dirX = this.startPosition.x - this.position.x;
let dirY = this.starPosition.y - this.position.y;
this.position.x = this.position.x + dirX / this.speed;
this.position.y = this.position.y + dirY / this.speed;
let dirX = ball.position.x - this.position.x;
let dirY = ball.position.y - this.position.y;
this.position.x = this.position.x + dirX / this.speed;
this.position.y = this.position.y + dirY / this.speed;
let ballDist = Math.aqrt((dirX * dirY) + (dirY * dirY));
if (ballDistance <= 10 &&
ball.isOnDestination()))
actualPlayer = this;
displayPlayer();

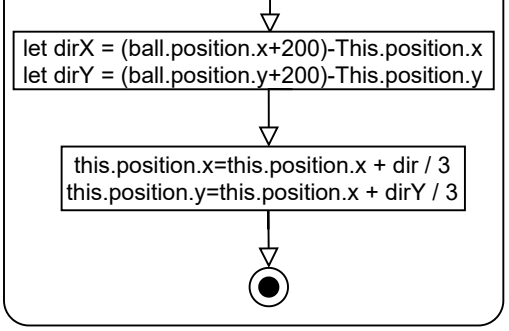

draw
crc2.save();
crc2.translate(this.position.x,
this.position.y);
else
if (this.team == 0)
draw the player
crc2.fillStyle = colorTeamOne.value
draw the player
crc2.fillStyle = colorTeamTwo.value

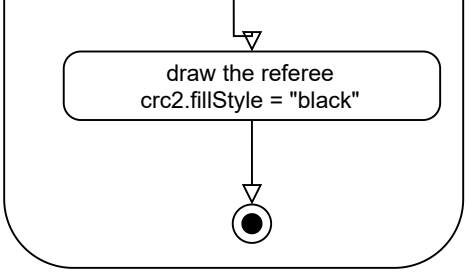

_newPrecision: number
changePrecision
this.precision = _ newPrecision;


_newSpeed: number;
changeSpeed
this.speed= _ newSpeed;

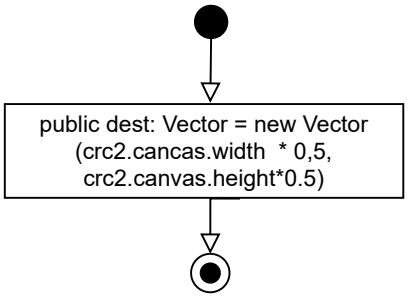

Referee
private refereeStart: Vecor =
newVector(crc2.canvas.width* 0.5,
crc2.canvas.height*0.9);
_timeslice: number
move
let a = ball.position.x - this.refereeStart.x;
let b = ball.position.y - this.refereeStart.y;
let ballDistance = Math.sqrt((a*a)+(b*b));
draw
crc2.save();
crc2.translate(this.position.x, this.position.y);

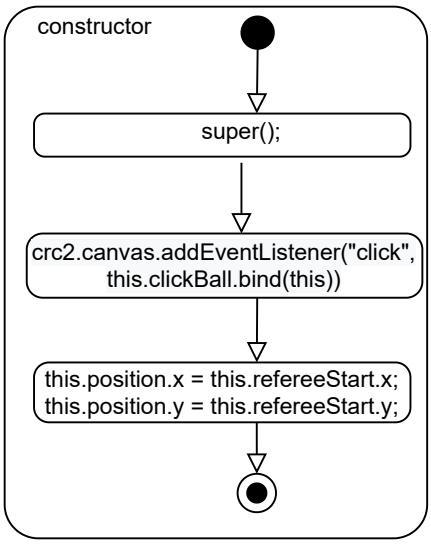

constructor
super();
this.position.x = this.refereeStart.x;
this.position.y = this.refereeStart.y;

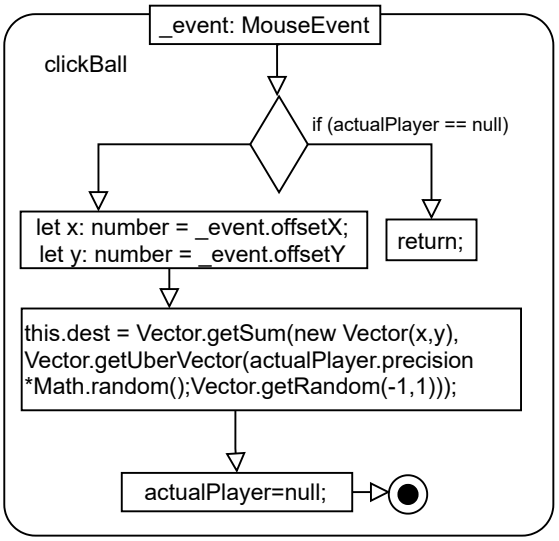

let dirX = (ball.position.x+200)-This.position.x
let dirY = (ball.position.y+200)-This.position.y
this.position.x=this.position.x + dir / 3
this.position.y=this.position.x + dirY / 3

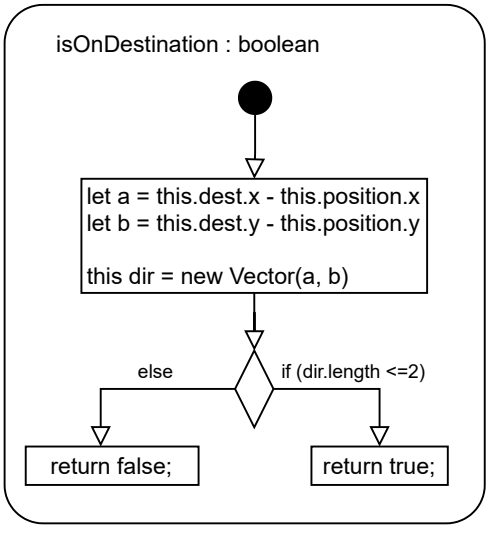

draw the referee
crc2.fillStyle = "black"

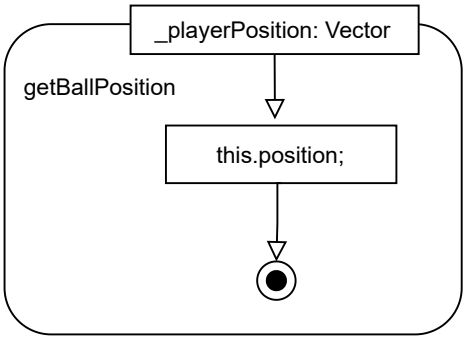
Ball

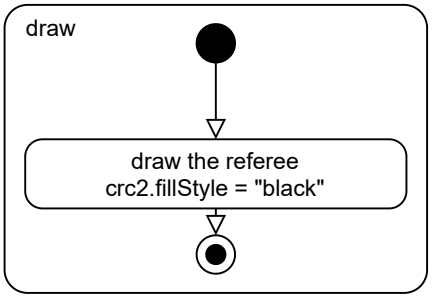

public dest: Vector = new Vector
(crc2.cancas.width * 0,5,
crc2.canvas.height*0.5)

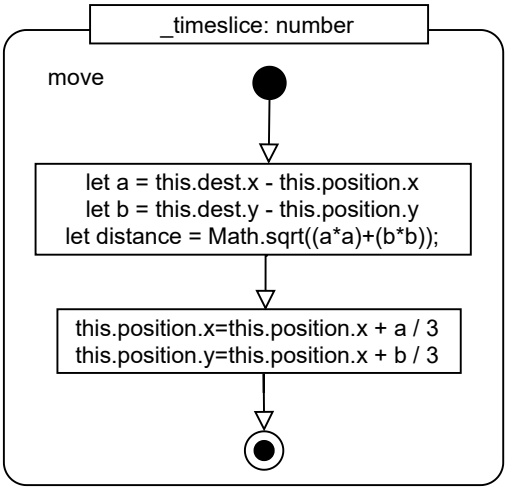

constructor
super();
crc2.canvas.addEventListener("click",
this.clickBall.bind(this))
this.position.x = this.refereeStart.x;
this.position.y = this.refereeStart.y;


_event: MouseEvent
clickBall
if (actualPlayer == null)
let x: number = _event.offsetX;
let y: number = _event.offsetY
return;
this.dest = Vector.getSum(new Vector(x,y),
Vector.getUberVector(actualPlayer.precision
*Math.random();Vector.getRandom(-1,1)));
actualPlayer=null;

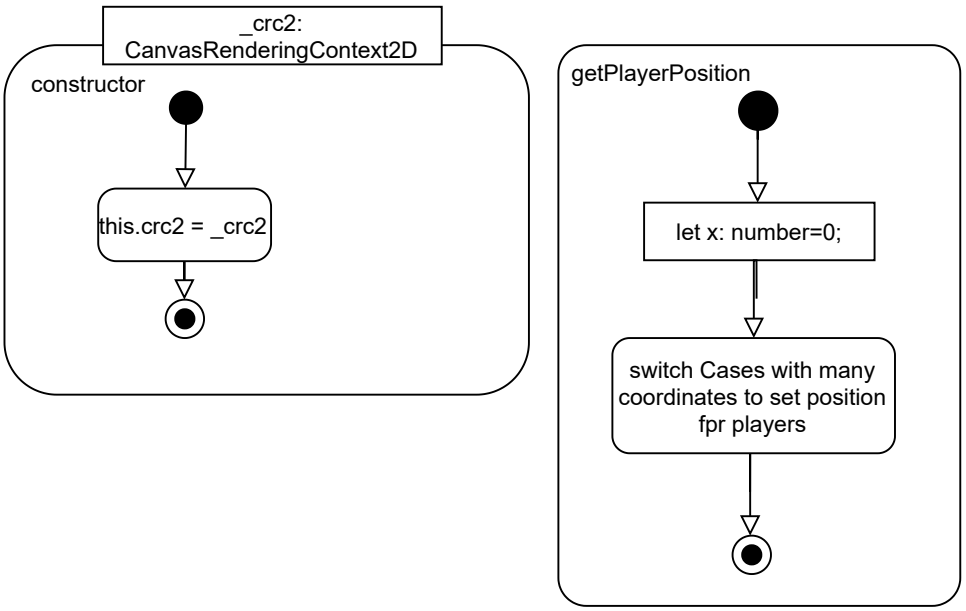

isOnDestination : boolean
let a = this.dest.x - this.position.x
let b = this.dest.y - this.position.y
this dir = new Vector(a, b)
else
if (dir.length <=2)
return false;
return true;


_playerPosition: Vector
getBallPosition
this.position;

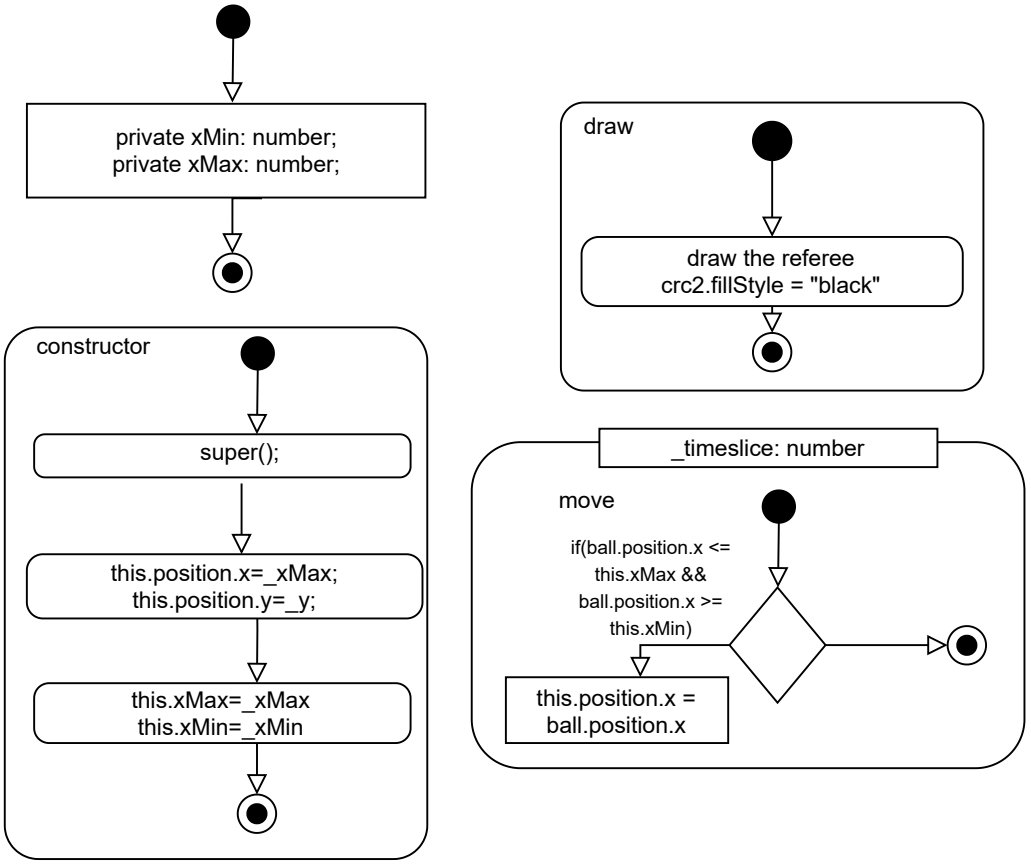

draw
draw the referee
crc2.fillStyle = "black"


_timeslice: number
move
let a = this.dest.x - this.position.x
let b = this.dest.y - this.position.y
let distance = Math.sqrt((a*a)+(b*b));
this.position.x=this.position.x + a / 3
this.position.y=this.position.x + b / 3

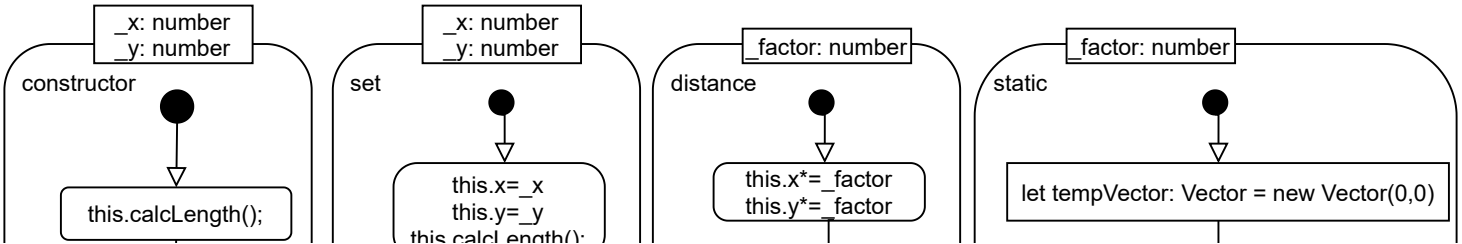
Position

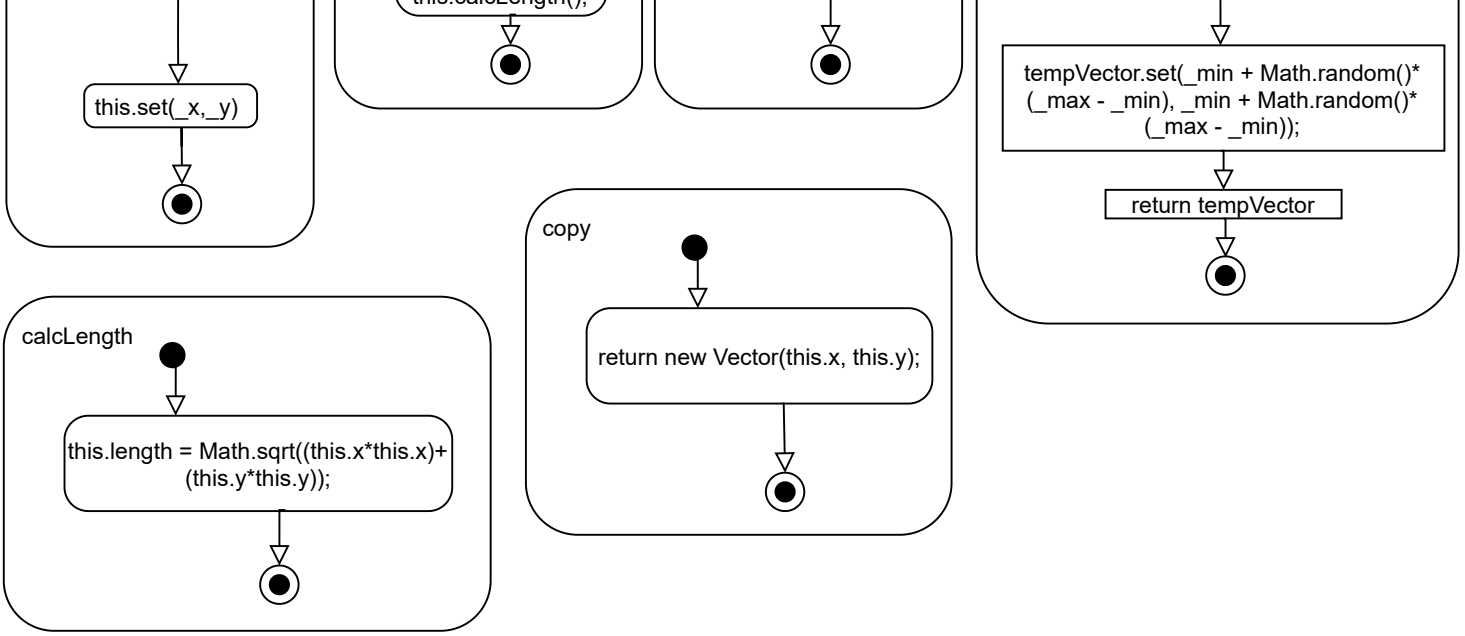


Linesman



Vector




this.set(_x,_y)
tempVector.set(_min + Math.random()*
(_max - _min), _min + Math.random()*
(_max - _min));
return tempVector

copy
return new Vector(this.x, this.y);

calcLength
this.length = Math.sqrt((this.x*this.x)+
(this.y*this.y));