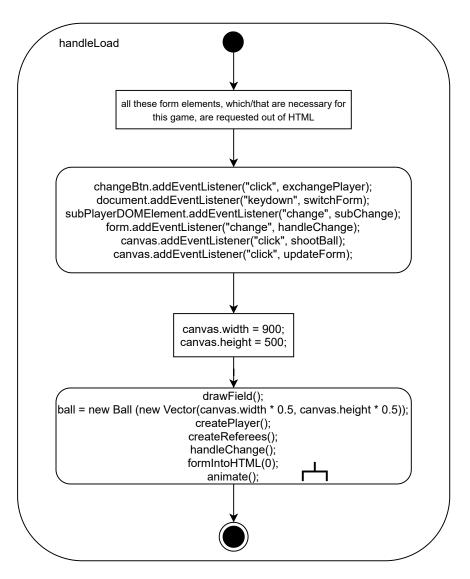
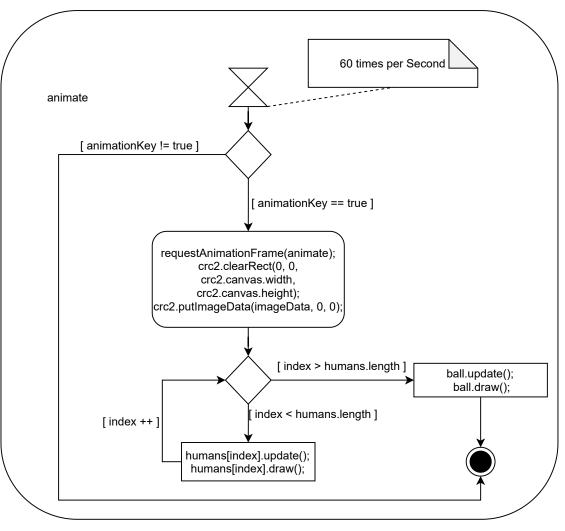
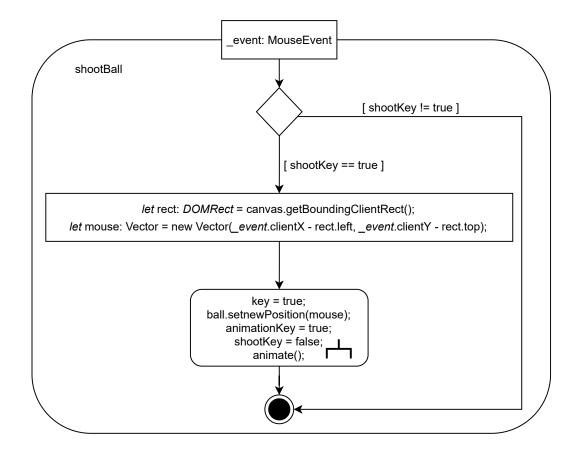
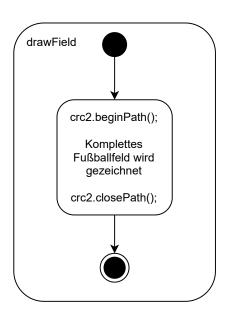
Activity-Diagram: Main let canvas: HTMLCanvasElement = <HTMLCanvasElement> document.querySelector("canvas"); let crc2 CanvasRenderingContext2D = <CanvasRenderingContext2D> canvas.getContext("2d"); export enum Task{}; let imageData: ImageData; change keyDown load let ball: Ball; let key: boolean; let animationKey: boolean = true; switchForm + handleLoad H subChange H let shootKey: boolean = false; let humans: Human[] = []; let scoreA: number = 0; let scoreB: number = 0; click change let form: HTMLDivElement; let playerNumberDOMElement: HTMLParagraphElement; shootBall handleChange H let teamDOMElement: HTMLParagraphElement; let teamADOMElement: HTMLButtonElement; let teamBDOMElement: HTMLButtonElement; let speedPlayer: HTMLParagraphElement; let precPlayer: HTMLParagraphElement; let numberPlayer: HTMLParagraphElement; let teamPlayer: HTMLParagraphElement; let speedSub: HTMLParagraphElement; let precSub: HTMLParagraphElement; let numberSub: HTMLParagraphElement; let teamSub: HTMLParagraphElement; let subPlayerDOMElement: HTMLSelectElement; let scoreADOMElement: HTMLElement; let scoreBDOMElement: HTMLElement; let changeBtn: HTMLButtonElement; let scoreA: number = 0; let scoreB: number = 0; let posession: HTMLParagraphElement; createReferees players.push(new Referee(new Vector(450, 150), "white")); players.push(new Linereferee(new Vector(680, 15), "pink")); players.push(new Linereferee(new Vector(230, 485), "pink"));



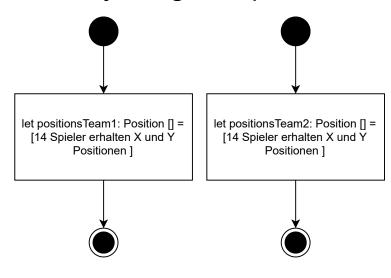




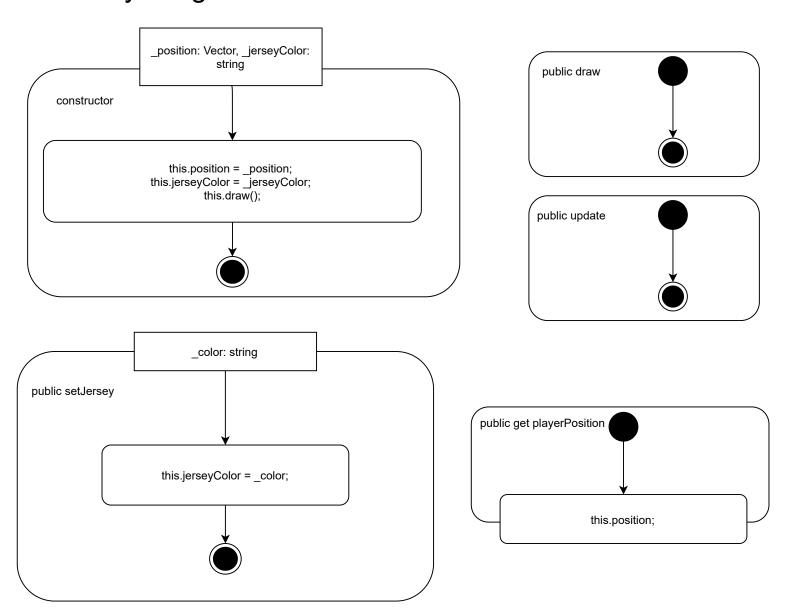
Activity-Diagram: footballField



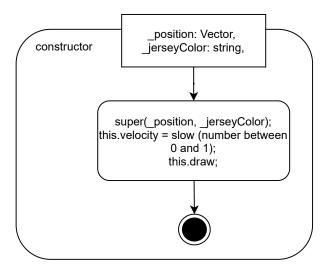
Activity-Diagram: position

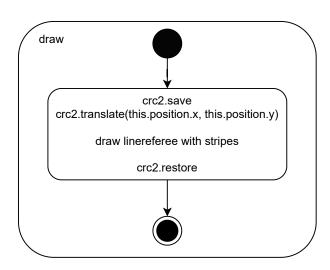


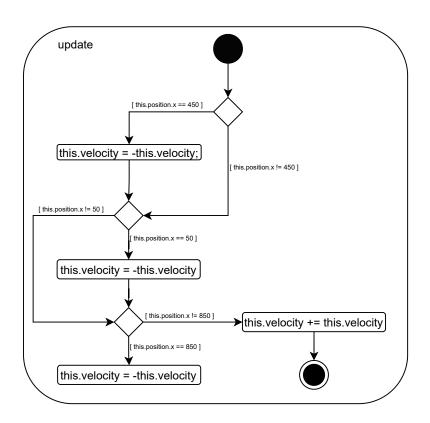
Activity-Diagram: human



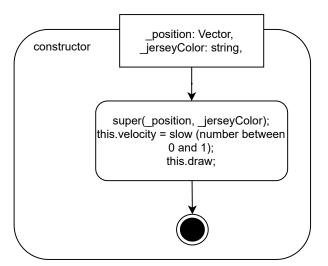
Activity-Diagram: lineReferee

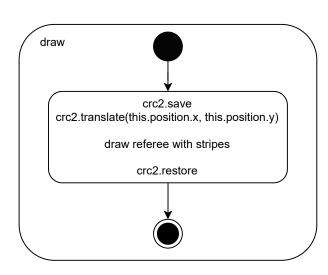


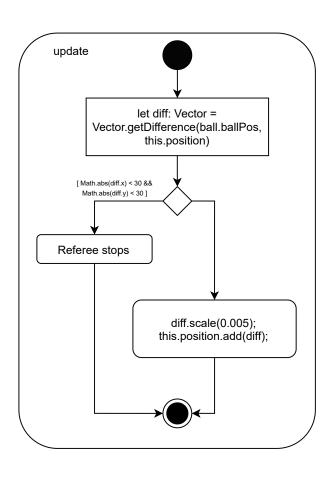




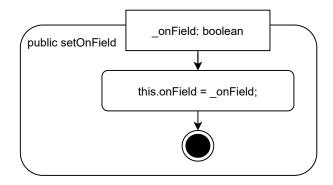
Activity-Diagram: Referee

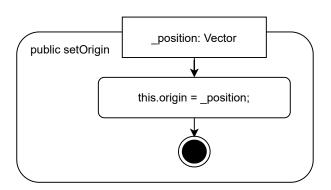


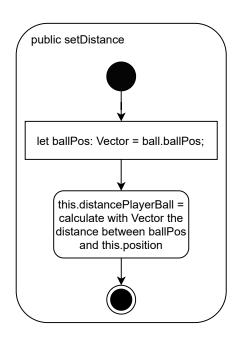


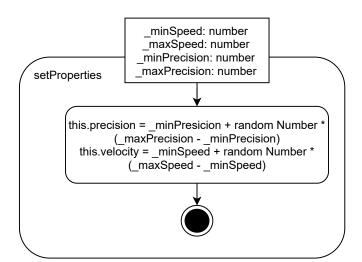


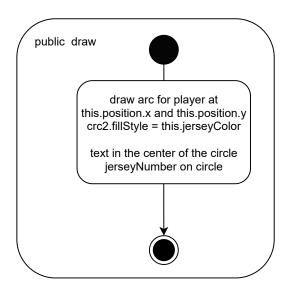
Activity-Diagram: Player position: Vector, _jerseyColor: string, _onField: boolean, jerseyNumber: number, constructor _team: string, task: Task = Task.lookForBall super(_position, _jerseyColor); origin: Vector; this.onField = _onField; this.velocity = 0.5; precision: number; this.jerseyNumber = _jerseyNumber; this.origin = this.position (origin); radius: number = 80; jerseyNumber: number; distancePlayerBall: number; this.team = _team; onField: boolean; team: string; newPosition: Vector; public get jerseyNumberPlayer public get distance public get playerSpeed this.jerseyNumber; this.velocity; this.distancePlayerBall; public get playerPrecision public get playerOnField public get playerTeam return this.precision; return this.onField; return this.team; public get playerOrigin _position: Vector private movePlayer let playerDifference: Vector = return this.origin; Vector.getDifference(_position, this.position); let playerDifference: Vector = Vector.getDifference(_position, this.position); let ratio: number = this.velocity / playerDistance playerDifference.scale(ratio); this.position.add(playerDifference); this.draw;

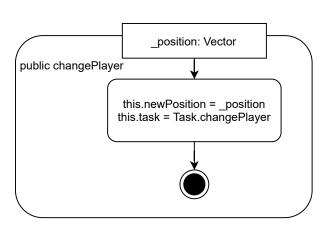


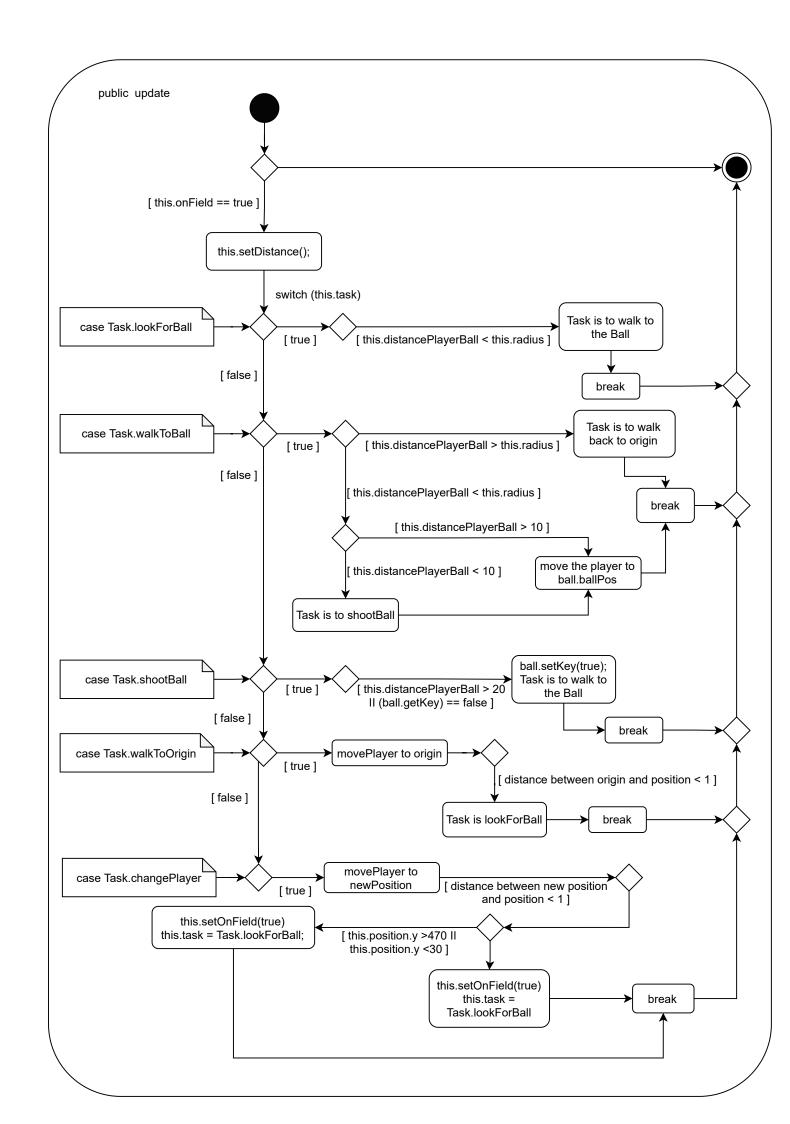




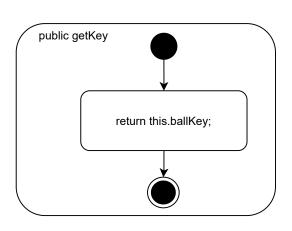


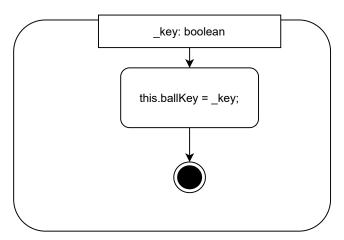


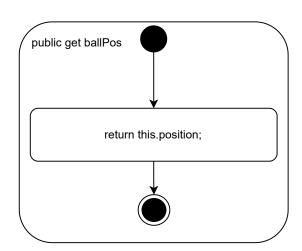


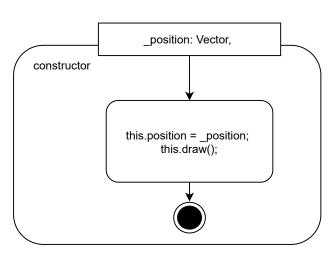


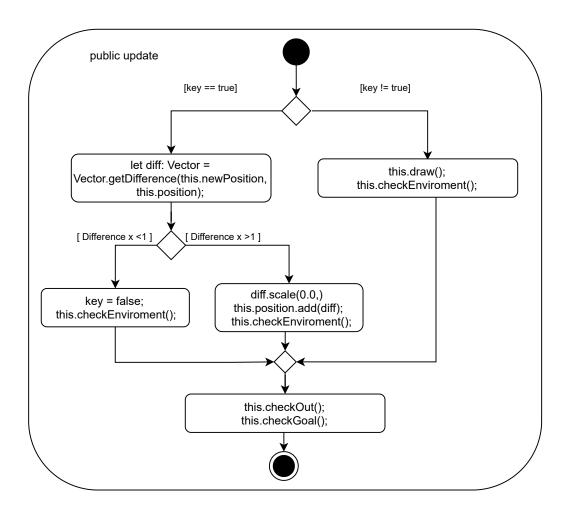
Aktivitätsdiagramm- Ball

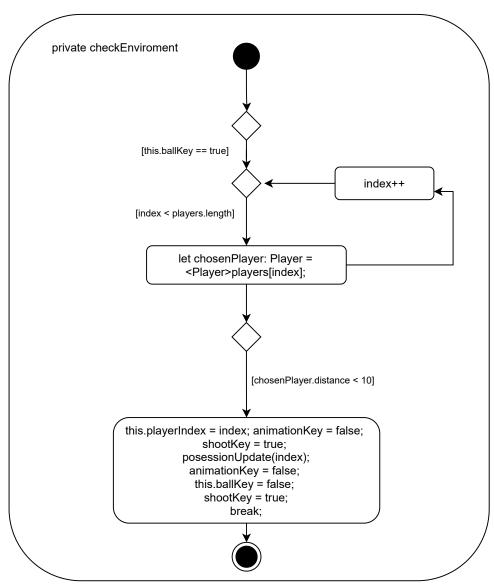


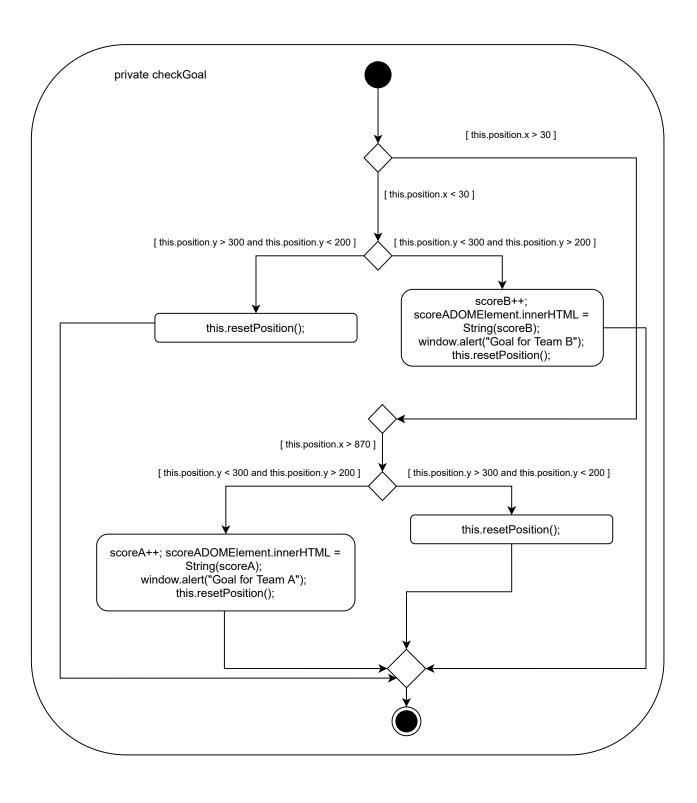


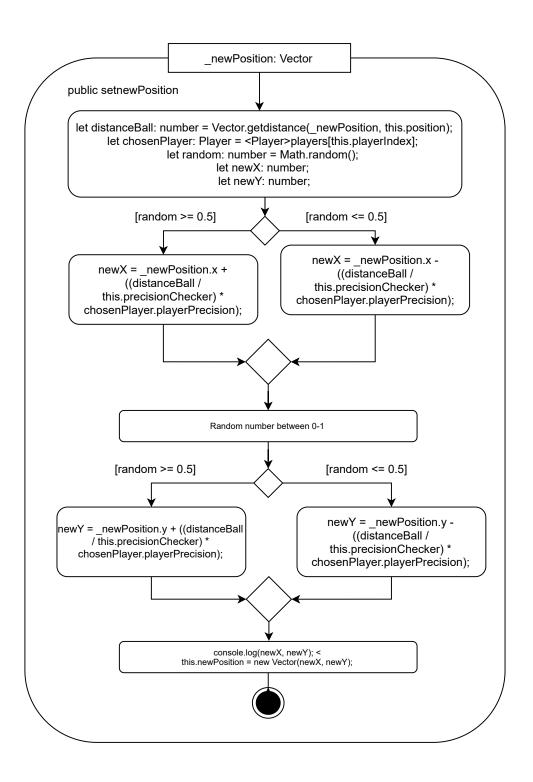




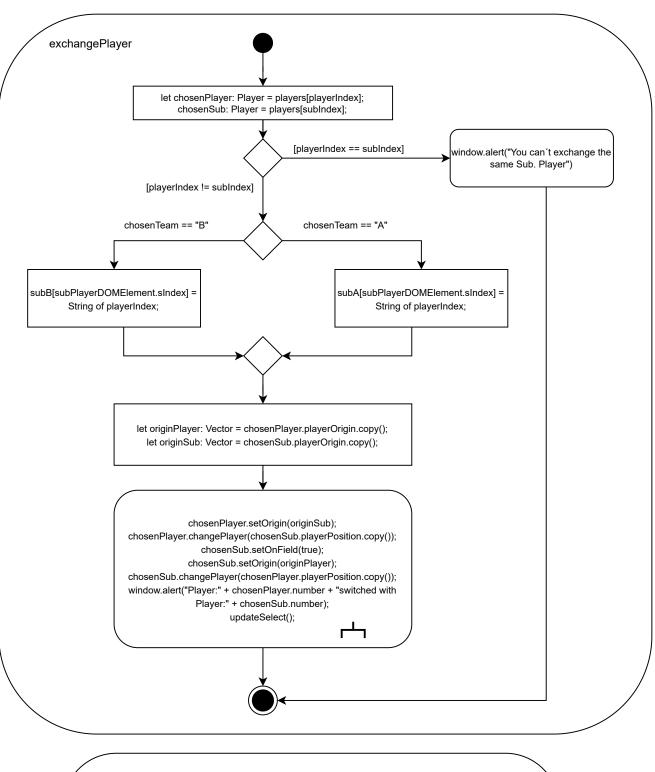


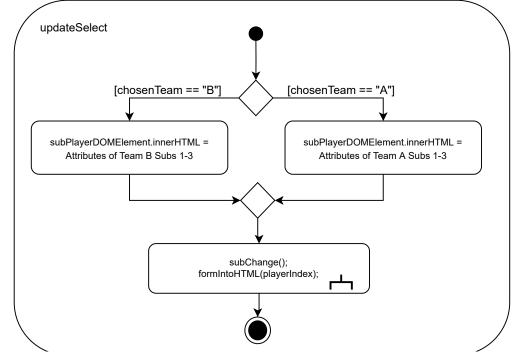


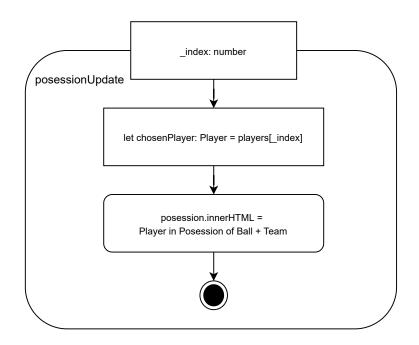


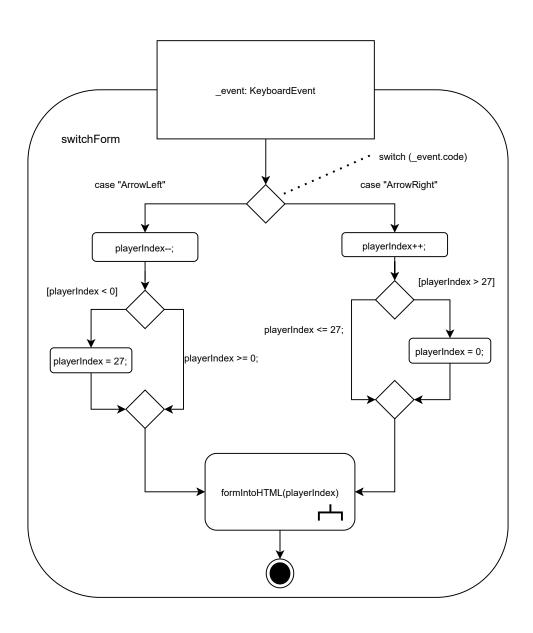


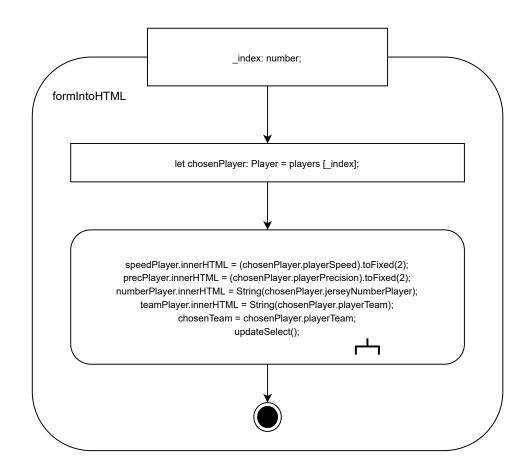
Aktivitätsdiagramm-Forms let valuesGlobal: string []; let playerIndex: number = 0; let subIndex: number; let chosenTeam: string; let subA: string: [] = [22, 23, 24]; let subB: string [] = [25, 26, 27]; handleChange let formData: FormData = new FormData (document.forms[0]); valuesGlobal = []; push Data in valuesGlobal [index = 0][formData++] (durchläuft alle FormDatas) [index++] [index >= 28] [index < 28] let chosenPlayer: Player = players[index]; setProperties of chosenPlayer with valuesGlobal [index >= 25] set Jersey of set Jersey of [index >= 22] [index < 22] [index < 11] players[index] with players[index] with valuesGlobal[5]; valuesGlobal[4]; players[index].draw(); players[index].draw(); [index < 25] index >= 11 set Jersey of set Jersey of players[index] players[index] with with valuesGlobal[5]; valuesGlobal[4]; players[index].draw(); players[index].draw();

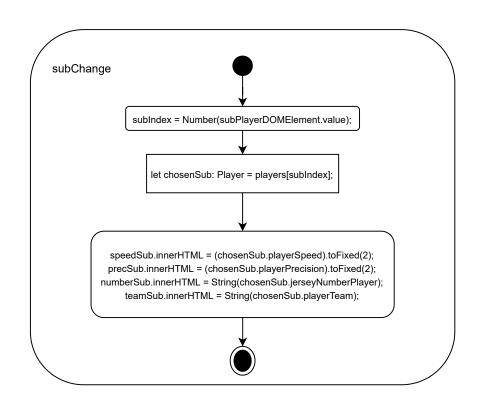












Activity Diagram Vector

