Football : Activity Diagramm

export let canvas: HTMLCanvasElement

export let crc2: CanvasRenderingContext2D

export let scale: number = window.devicePixelRatio

export let canvasBall: HTMLCanvasElement

export let canvasPlayers: HTMLCanvasElement

export let crc2Players: CanvasRenderingContext2D

export let canvasReferee: HTMLCanvasElement

export let crc2Referee: CanvasRenderingContext2D

let colorOne: HTMLInputElement

let colorTwo: HTMLInputElement

let minSpeedInput: HTMLInputElement

let maxSpeedInput: HTMLInputElement

let minPrecisionInput: HTMLInputElement

let maxPrecisionInput: HTMLInputElement

let form: HTMLDivElement

export let minSpeed: number = 1

export let maxSpeed: number = 5

let minPrecision: number = 1

let maxPrecision: number = 5

let scoreTeamOne: HTMLElement

let scoreTeamTwo: HTMLElement

let currentPlayer: HTMLElement

let playerNumber: HTMLElement

let playerSpeed: HTMLElement

let playerPrecision: HTMLElement

let x: number[] = [10, 150, 150, 150, 150, 425, 425, 425, 725, 750, 725]

let y: number[] = [350, 125, 275, 425, 575, 175, 350, 525, 125, 350, 575]

let a: number[] = [990, 850, 850, 850, 850, 575, 575, 575, 275, 250, 275]

let b: number[] = [350, 575, 425, 275, 125, 525, 350, 175, 575, 350, 125]

export let people: Player[] = []

export let colors: string[]

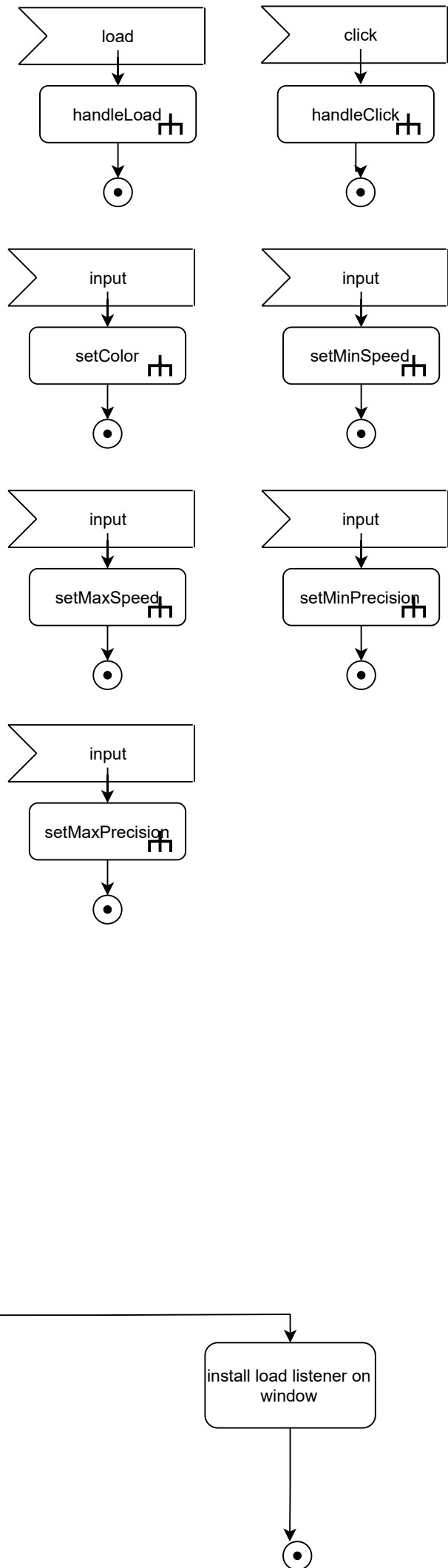export let referees:Referee[] = []

export let clickX: number

export let clickY: number

export let ball: Ball

export let positionBall: Vector

export let playerPosition: Vector

export let i: number = 0

export let j: number = 6

let scores: number[] = [0, 0]

let ballMovement: number

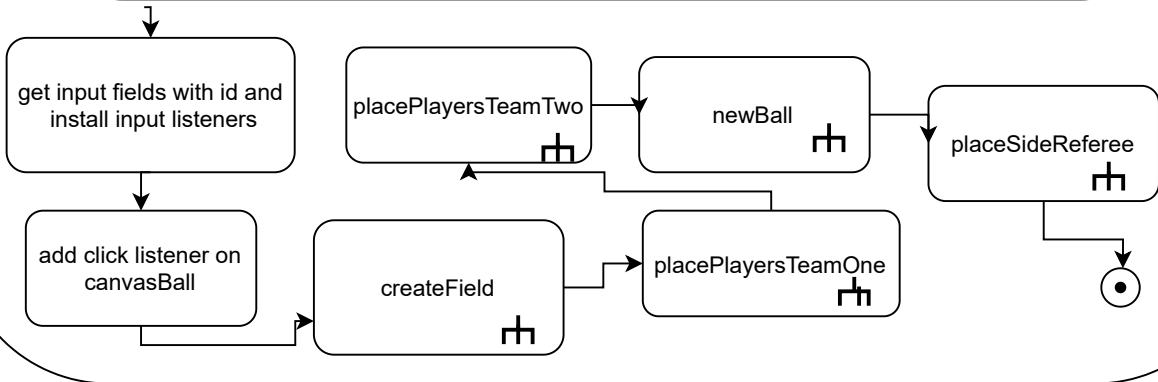let playerMovement: number

let clicked: boolean

handleLoad

```
canvasBall = <HTMLCanvasElement>document.getElementById("ball");
crc2Ball = <CanvasRenderingContext2D>canvasBall.getContext("2d");
canvasPlayers = <HTMLCanvasElement>document.getElementById("players");
crc2Players = <CanvasRenderingContext2D>canvasPlayers.getContext("2d");

canvas = <HTMLCanvasElement>document.getElementById("field");
crc2 = <CanvasRenderingContext2D>canvas.getContext("2d");
canvasReferee = <HTMLCanvasElement>document.getElementById("referee");
crc2Referee = <CanvasRenderingContext2D>canvas.getContext("2d");
canvas.width = 1000 * scale;
canvas.height = 700 * scale;
canvasBall.width = 1000 * scale;
canvasBall.height = 700 * scale;
canvasPlayers.width = 1000 * scale;
canvasPlayers.height = 700 * scale;
canvasReferee.width = 1000 * scale;
canvasReferee.height = 800 * scale;
colors = ["black", "red"]
clicked = false
```

get input fields with id and install input listeners

placePlayersTeamTwo

newBall

placeSideReferee

add click listener on canvasBall

createField

placePlayersTeamOne

createPlayersTeamOne

```
for (let i: number = 0; i < 11; i++) {
playerPosition = new Vector(x[i] * scale, y[i] * scale);
let player: Player = new Player(playerPosition, colors[0], i);
if (player.speed == undefined) {
  player.speed = Math.floor(Math.random() * (maxSpeed - minSpeed + 1) + minSpeed) / 200;
}
if (player.precision == undefined) {
  player.precision = Math.floor(Math.random() * (maxPrecision - minPrecision + 1) + minPrecision);
}
player.draw();
people.splice(i, 1, player);

}
```

_event: Event

setMaxPrecision

```
let amount: string = (<HTMLInputElement>_event.target).value;
console.log(amount);
maxPrecision = parseFloat(amount);

for (let player of people) {
   player.changePrecision(minPrecision, maxPrecision);
}
console.log(people);
```

createField

create lines of football pitch (s. scribble)

setColor

_event: Event

```
let color: string = (<HTMLInputElement>_event.target).value;
/*  console.log(_event.target);
 console.log(colorOne); */
if (_event.target == colorOne) {
   for (let i: number = 0; i < 11; i++) {
      colors[0] = color;
      let player: Player = people[i];
      player.changeColor(color);
      console.log(player.color);
      placePlayersTeamOne();
   }
}
else {
   colors[1] = color;
   for (let i: number = 11; i < 22; i++) {
      colors[1] = color;
      let player: Player = people[i];
      player.changeColor(color);
      placePlayersTeamTwo();
   }

}
```

setMinSpeed

_event: Event

```
let amount: string =
(<HTMLInputElement>_event.target).
value;
 console.log(amount);
 minSpeed = parseFloat(amount);

 for (let player of people) {
      player.changeSpeed
(minSpeed, maxSpeed);
    }
```

setMaxSpeed

_event: Event

```
let amount: string = (<HTMLInputElement>_event.target).value;
    console.log(amount);
    maxSpeed = parseFloat(amount);

    for (let player of people) {
       player.changeSpeed(minSpeed, maxSpeed);
    }
```

setMinPrecision

_event: Event

```
let amount: string = (<HTMLInputElement>_event.target).value;
                  console.log(amount);
           minPrecision = parseFloat(amount);

               for (let player of people) {
         player.changePrecision(minPrecision, maxPrecision);
                  }
```
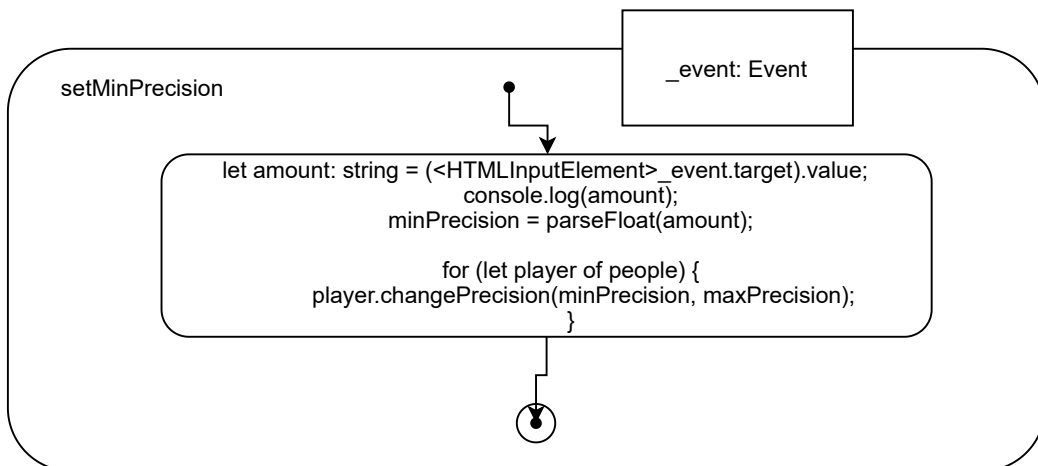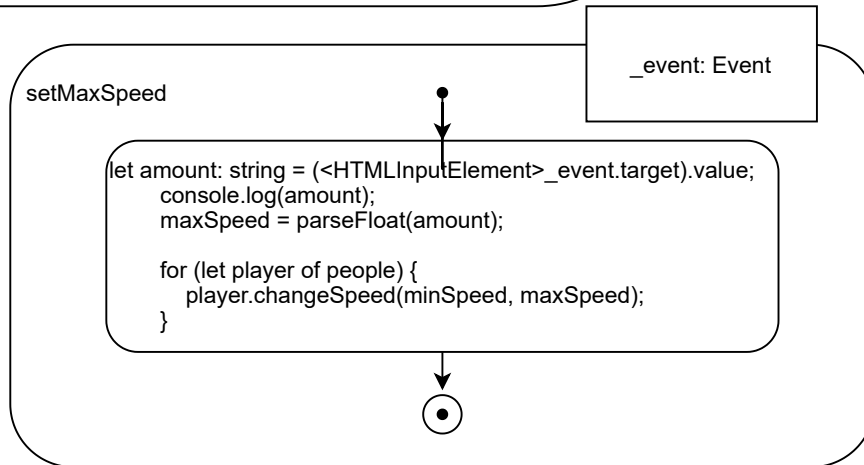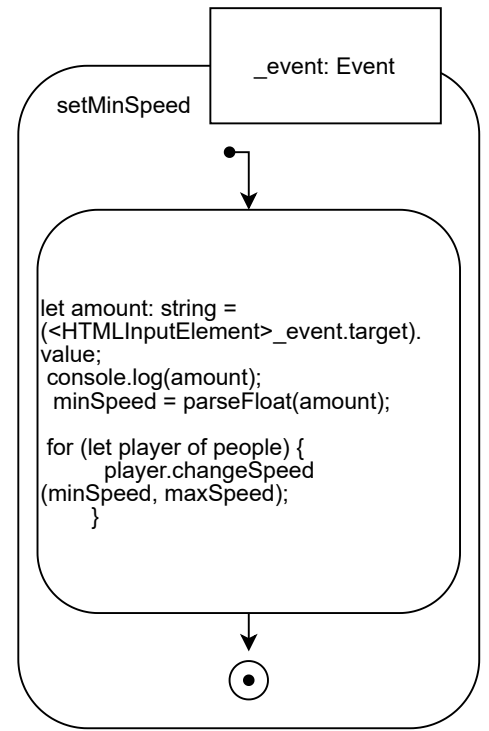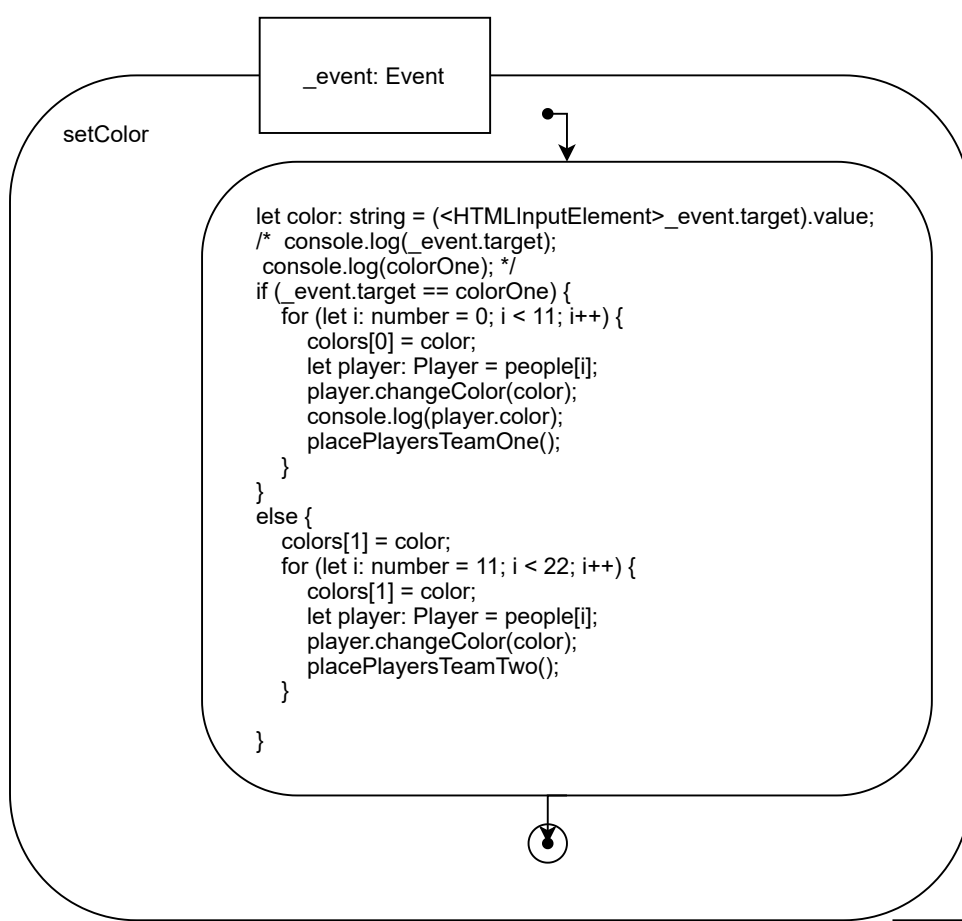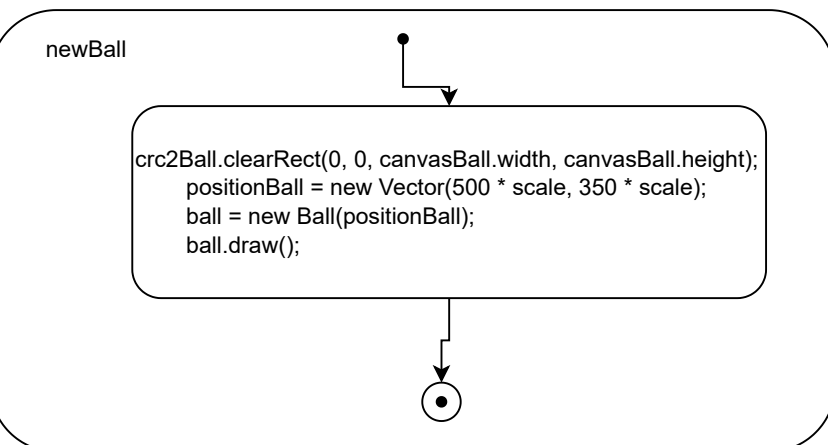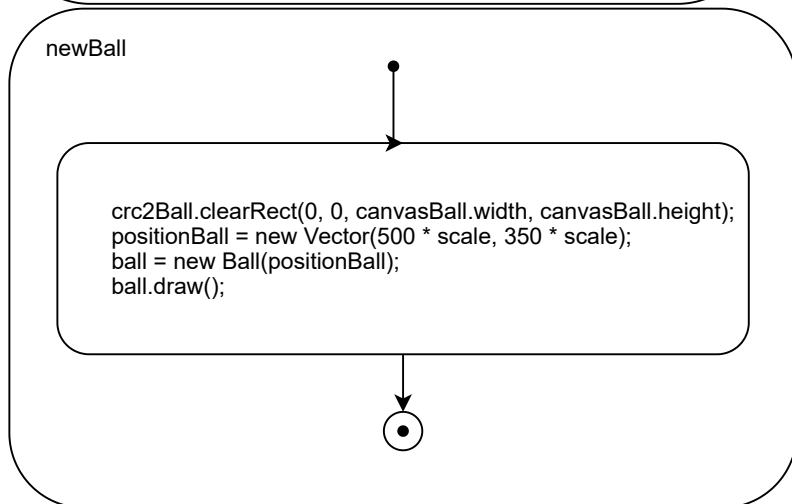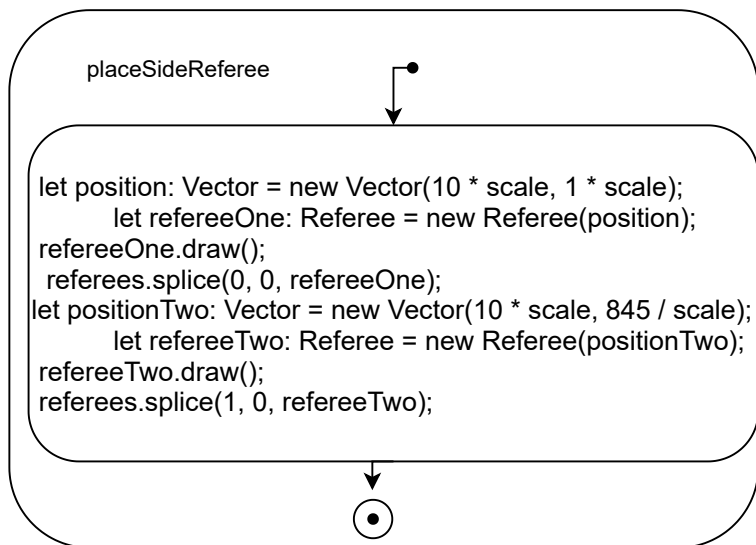
setScore

```
if (ball.goal == false) {
let scoreOne: number = scores[0];
 let scoreTwo: number = scores[1];
 if (994 * scale <= positionBall.x && positionBall.x
<= 1000 * scale && 315 * scale <= positionBall.y &&
positionBall.y <= 385 * scale) {
scoreOne += 1;
scores.splice(0, 1, scoreOne);
scoreTeamOne.innerHTML = scoreOne.toString() + ":";
ball.goal = true;
clearInterval(ballMovement);

newBall();
 }
 else if (0 <= positionBall.x && positionBall.x <= 6 *
scale && 315 * scale << positionBall.y &&
positionBall.y <= 385 * scale) {
scoreTwo += 1;
scores.splice(1, 1, scoreTwo);
scoreTeamTwo.innerHTML = scoreTwo.toString();
ball.goal = true;
clearInterval(ballMovement);

newBall();
```

moveBall

```
let minX: number = clickX - 1;
let maxX: number = clickX + 1;
let minY: number = clickY - 1;
let maxY: number = clickY + 1;
crc2Ball.clearRect(0, 0, canvasBall.width, canvasBall.height);

ball.move(1 / 50);
ball.draw();
/* console.log(positionBall, clickX, clickY); */
if (minX <= positionBall.x && positionBall.x <= maxX
&& minY <= positionBall.y && positionBall.y <= maxY) {
   clearInterval(ballMovement);
   setScore();
```

createPlayersTeamTwo

```
for (let i: number = 11; i < 22; i++) {
  playerPosition = new Vector(a[i - 11] * scale, b[i - 11] * scale);
  let player: Player = new Player(playerPosition, colors[1], i);
  if (player.speed == undefined) {
    player.speed = Math.floor(Math.random() * (maxSpeed - minSpeed + 1) + minSpeed) / 200;
  }
  if (player.precision == undefined) {
    player.precision = Math.floor(Math.random() * (maxPrecision - minPrecision + 1) + minPrecision);
  }
  player.draw();
  people.splice(i, 1, player);

}
```

placeSideReferee

```
let position: Vector = new Vector(10 * scale, 1 * scale);
        let refereeOne: Referee = new Referee(position);
refereeOne.draw();
 referees.splice(0, 0, refereeOne);
let positionTwo: Vector = new Vector(10 * scale, 845 / scale);
        let refereeTwo: Referee = new Referee(positionTwo);
refereeTwo.draw();
referees.splice(1, 0, refereeTwo);
```

newBall

```
crc2Ball.clearRect(0, 0, canvasBall.width, canvasBall.height);
positionBall = new Vector(500 * scale, 350 * scale);
ball = new Ball(positionBall);
ball.draw();
```

newBall

```
crc2Ball.clearRect(0, 0, canvasBall.width, canvasBall.height);
        positionBall = new Vector(500 * scale, 350 * scale);
        ball = new Ball(positionBall);
        ball.draw();
```

```
handleClick    _event: MouseEvent

let rectangleB: DOMRect = canvasBall.getBoundingClientRect();
        clickX = Math.floor(_event.clientX - rectangleB.left);
        clickY = Math.floor(_event.clientY - rectangleB.top);
        console.log(clickX, clickY);
        if (clicked == true) {
            clicked = false
        for (let player of people) {
            player.checkClick(clickX, clickY);
            if (player.clicked == true) {
                playerNumber.innerHTML = "Player " + player.playerNumber;
                playerSpeed.innerHTML = "Speed " + player.speed;
                playerPrecision.innerHTML = "Precision " + player.precision;
                player.clicked = false;
                clicked = true;
        if (clicked == false) {
            playerNumber.innerHTML = "";
            playerSpeed.innerHTML = "";
            playerPrecision.innerHTML = "";
            ballMovement = setInterval(moveBall, 20);
            playerMovement = setInterval(movePlayer, 20);
}
```
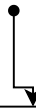
movePlayer

```
crc2Players.clearRect(0, 0, canvasPlayers.width, canvasPlayers.height);
for (let player of people) {
    /* console.log(player.speed); */
    player.checkPosition();
    if (player.near == true) {
        player.move();
        player.draw();
    player.checkCollision();
    if (player.atBall == true) {
        clearInterval(playerMovement);
        clearInterval(ballMovement);
        currentPlayer.innerHTML = "Player" + player.playerNumber;
        playerMovement = setInterval(movePlayerBack, 20);
```

movePlayerBack

```
crc2Players.clearRect(0, 0, canvasPlayers.width, canvasPlayers.height);
for (let player of people) {
    /* console.log(player.speed); */
    if (player.atBall == false) {
        console.log("false");
        player.moveToStart();
        player.draw();
    else if (player.atBall == true) {
        player.draw();
    }
    let currentPosition: Vector = new Vector(Math.floor(player.position.x), Math.floor(player.position.y));

if (player.startPosition.x - 10 <= currentPosition.x && currentPosition.x <= player.startPosition.x + 10 && player.startPosition.y - 10
<= currentPosition.y && currentPosition.y <= player.startPosition.y + 10) {
        player.atStartposition = true;
    }
    let k: number = 0;
    for (let player of people) {
        if (player.atStartposition == true) {
            k += 1;
        }
    }
    if (k == 21) {
        clearInterval(playerMovement);
        player.atBall = false;
        player.near = false;
```

- `<canvas>` id=field 1
- `<canvas>` id=referee 2
- `<canvas>` id=players 3
- `<canvas>` id=ball 4

- `<p>` id=scoreOne
- `<h1>`
- `<p>` id=scoreTwo

Player with ball

Score
0 : 0

Select what you want!

color team one
color team two

Minimal speed 10 — 4
Maximum speed 50 — 10

Minimal precision 10 — 4
Maximum precision 50 — 10

- `<div>` id=scoreDiv
- `<h1>`
- `<input>` id=colorOne
- `<input>` id=colorTwo
- `<div>` id=form
- `<input>` id=minspeed
- `<input>` id=maxspeed
- `<input>` id=minprecision
- `<input>` id=maxprecision

0/0

100

200

300

400

500

600

0/100

100

200

300

400

500

600

700

800

900

1000/0

1000/100

1000px ≙ 100 m
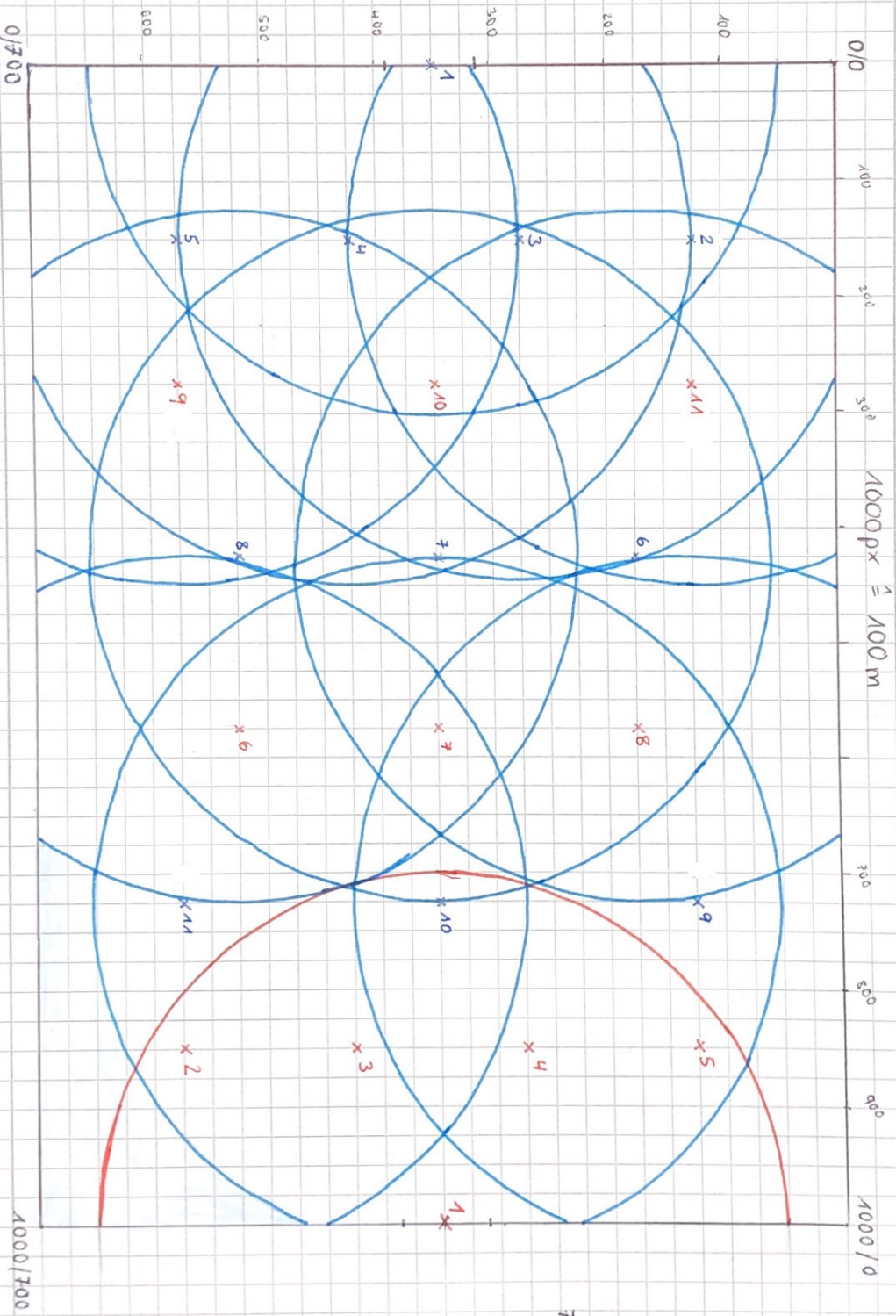
2 cm ≙ 10 m

700px ≙ 70 m

Angaben in px   2 cm ≙ 100 px ≙ 10 m

0/0   55/0   165/0

0/150

0/260

0/345

0/385   70 px

0/440

0/550

0/700

55/260

55/440

110/350

165/150

165/550

500/0

500/350
350

90

500/700

835/0   945/0   1000/0

835/150

835/550

945/260

500/350

545/440

70 px

1000/150

1000/260

1000/345

1000/440

1000/550

1000/700

User

Start

choose the jersey color

choose the velocity of the players

choose the precision of the players

switch players

show the number of the players

show the features of the players

click to move the ball