Jogis Soccer - Game Endabgabe EIA 2

Abgabe: 19.07.2021

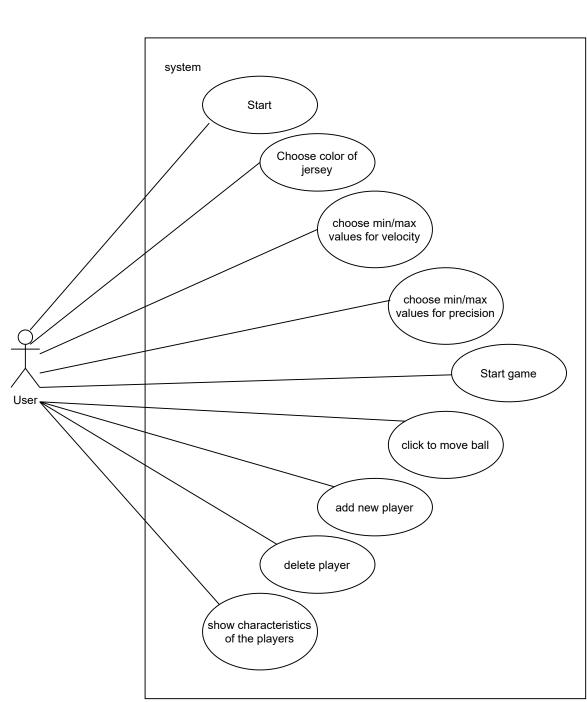
Dozent: Prof: Jirka Dell Oro-Friedl

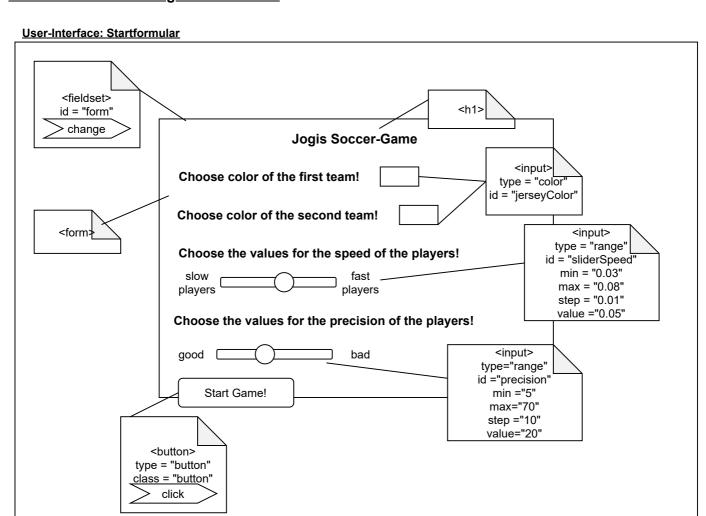
von

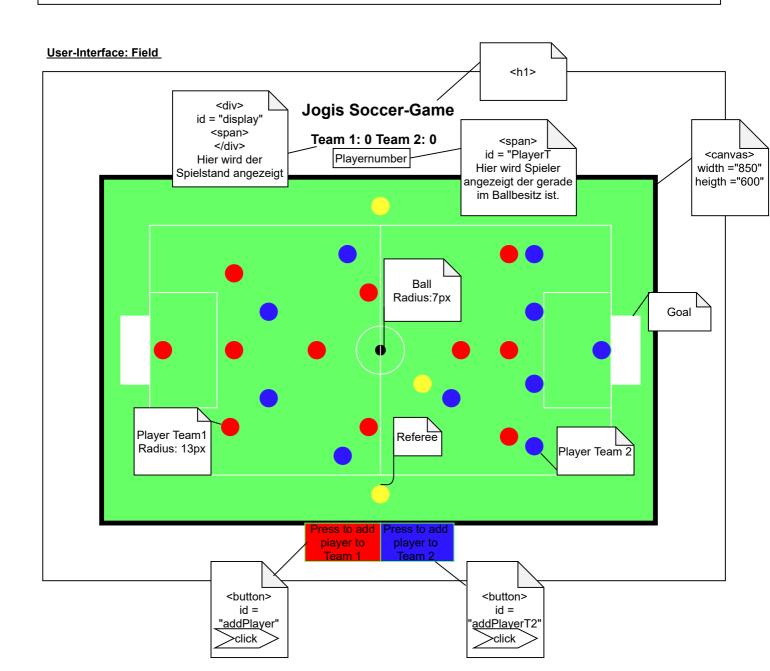
Name: Lukas Muschal Martrl.Nummer: 262264 Medienkonzeption

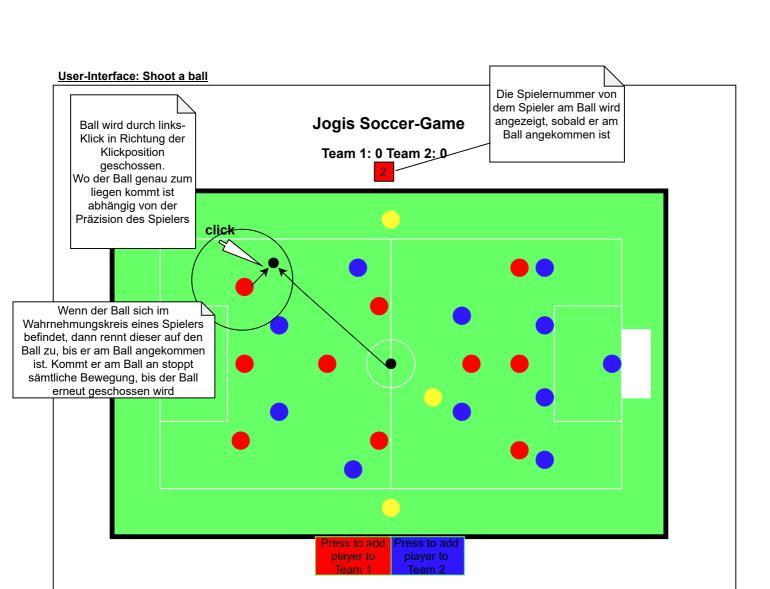
Inhalt
Use-Case-Diagramm
User-Interfaces
Klassendiagramme
Aktivitätsdiagramme

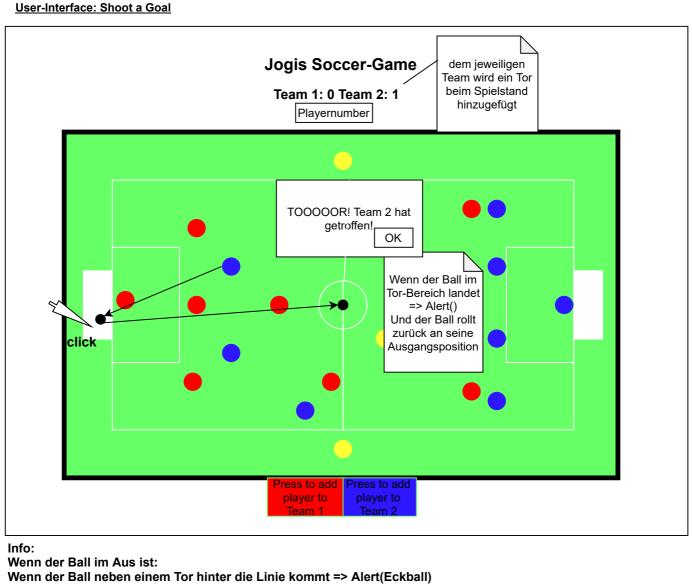
<u>Use-Case-Diagramm: Endabgabe EIA2 SS21</u>







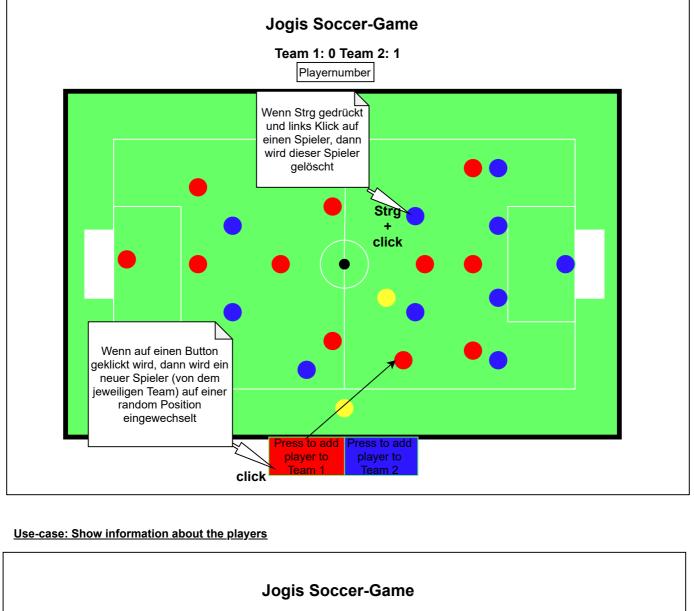


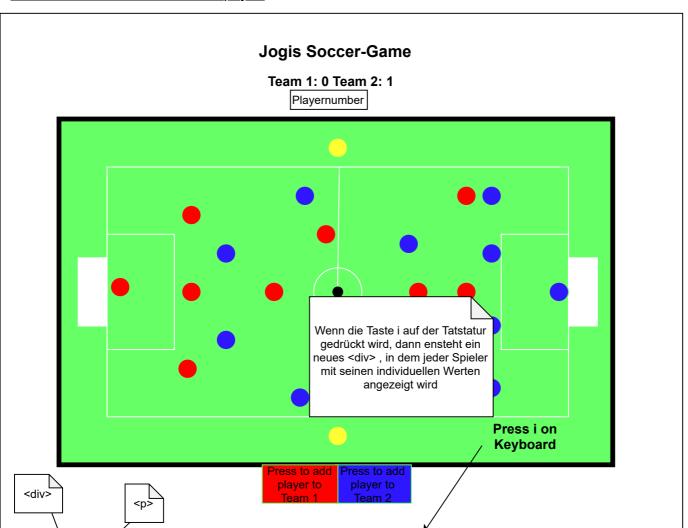


Der Ball geht dann an die jeweilige Ecke des Spielfeldes Wenn der Ball an den Seitenlinien ins Aus geht, => Alert(Ball im Aus!)

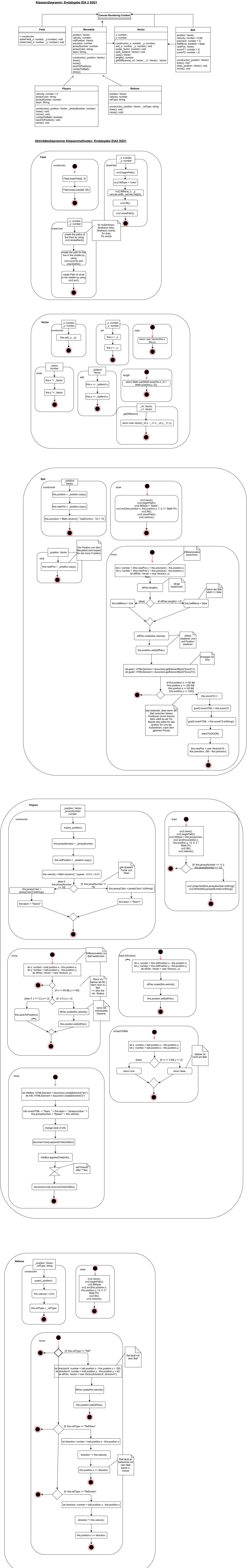
Der Ball geht dann an die Stelle zurück, an der er ins Aus gegangen ist.

User-Interface: Add or delete player





₹eam: x Team: x Team: x Team: x Team: x Jerseynumber: x Jerseynumber: x Jerseynumber: x Jerseynumber: x Jerseynumber: x Speed: x Speed: x Speed: x Speed: x Speed: x Precision: x Precision: x Precision: x Precision: x Precision: x Team: x Team: x Team: x Team: x Team: x Jerseynumber: x Jerseynumber: x Jerseynumber: x Jerseynumber: x Jerseynumber: x Speed: x Speed: x Speed: x Speed: x Speed: x Precision: x Precision: x Precision: x Precision: x Precision: x



Aktivitätsdiagramm: Endabgabe EIA 2 SS21 main install load listener on window load export let crc2: CanvasRenderingContext2D; export let canvas: HTMLCanvasElement; handleLoad export let imgData: ImageData; let form: HTMLElement; let button: HTMLElement; let button2: HTMLElement; let button3: HTMLElement; _event: Event export let jerseyColor1: String = ""; let jerseyColor2: String = ""; export let speed: number = 0.05; handleLoad export let ballControl: number = 20; export let moveables: Moveable [] = []; export let ball: Ball; canvas = document.getElemntsByTagName("canvas")[0]; export let ballArray: Ball [] = []; export let playerAtBall: PLayer; let positionTeams: number [] [] = [[x position of an player, y position of an player]]; let jerseyNumbers: number [] = [numbers of the players]; let positionRefs: number [] [] = [[pos x of an ref, pos y of an ref]]; let refTYp: string [] = ["Ref", "RefOben", "RefUnten"]; crc2=canvas.getContext("2d"); let feld: Field = new Field; get imagedata of canvas select form and add Eventlistener "change" select button and add Eventlistener "click" change click on button handleChange showField _event: Event handleChange let formData: HTMLInputElement = <HTMLInputElement>_event.target [if: formData.id == "jerseyColor1"] jerseyColor1 = formData.value [if: formData.id == "jerseyColor2"] jerseyColor2 = formData.value; [if: formData.id == "sliderSpeed"] speed = Number(formData.value); [if: formData.id == "precision"] ballControl = Number(formData.value) _event: Event click click on button2 showField form.classList.add("invisible"); button.classList.add("invisible"); clickBall newPLayer createPLayer(); click on button3 click createBall(); deletePlayer newPLayerT2 Add Eventlistener on canvas, button2, button3, document keypress window.setInterval(update, 50) showValues createPLayer createBall [for: i=0;i < 21 ball = newBall player = new Player ball.draw() i++ PLayer.draw(); ballArray.push(ball) moveables.push(player); [for: i = 0; $i \le 2$] reff = new Referee _event: MouseEvent das selbe für newPlayer die Funktion reff.draw(); j++ newPLayerT2() [for i = 0; $i \le 1$ moveables.push(reff) position New P Layer = randomposition j++ playernew = new PLayer() moveables.push(playernew); _event: MouseEvent deletePLayer let taste: boolean = _event.ctrlKey [if: taste == true] let rect: DOMRect = can vas. get Bounding Client Rect();let x = _event.clientX - rect.left let y = _vent.clientY - rect.top [for: [i, moveable] of moveables.entries()] let difVec = new Vector(moveable.position.x - x, moveable.position.y - y); let length = diffVec.length(); moveables.splice(i, 1); playeronBall for:player of [if: player.contactToball() == true] [if: playerAtBall.team == "Team1" moveables playerAtBall = player [else] playerNumber = document.getElementById("PlayerT); das selbe wie für Team1 playerNumber.innerHTML = player.jerseyNumber; return false return true playerNumber.style.backgroundColor = player.jerseyColor event: MouseEvent clickBall [if: taste == true let taste:. boolean = _event.ctrlKey; return; let rect: DOMRect = canvas.getBoundingClientRect(); let x = _event.client.X - rect.left; let y = _event.client.Y - rect.top; ball.click(newVector(x, y); _event: KeyboardEvent showValues let infoKey: string = _event.key; [if: infoKey == "i" [for: player of moveables] player.show() update crc2.clearRect(0, 0, crc2.canvas.width, crc2.canvas.height); crc2.putImageData(imgData, 0, 0); [if: ball.ballMove == false && [else] playeronBall() == true] [for player of moveables] [for let player of moveables] player.draw(); player.move(); player.draw(); [for ball of ballArray] ball.move(); ball.draw();