Klassendiagramm- Ball

canvas Rendering Context

Vector

+ x: number;

+ y: number;

+ constructor(_x: number = 0, _y: number = 0)

+ getDifference(_v0: Vector, _v1: Vector)

+ getDistance(_v0: Vector, _v1: Vector)

+ set(_x: number, _y: number)

+ scale(_factor: number)

+ add(_addend: Vector)

+ copy(): Vector;

<< enumeration >> Tasks

lookForBall

walkToBall

shootBall

walkToOrigin

changePlayer

<< Interface >> Position

lookForBall

walkToBall

shootBall

walkToOrigin

changePlayer

Ball

- position: Vector;

- newPosition: Vector;

- playerIndex: number;

- ballKey: boolean = true;

- precisionChecker: number = 100;

+ constructor(_position: Vector);

+ get ballPos(): Vector;

+ setKey(): Vector;

+ setnewPosition(_newPosition: Vector): void

+ draw(): void;

+ update(): void

- checkEnviroment(): void

- resetPosition(): void

Human

velocity: number;

position: Vector;

jerseyColor: string;

+ constructor(_position: Vector, _jerseyColor: string);

+ draw(): void;

+ update(): void;

+ public get playerPosition(): Vector;

+ setJersey(_color: string);

referee

lineReferee

Player

- task: Task;

- origin: Vector;

- precision: number;

- radius: number = 80;
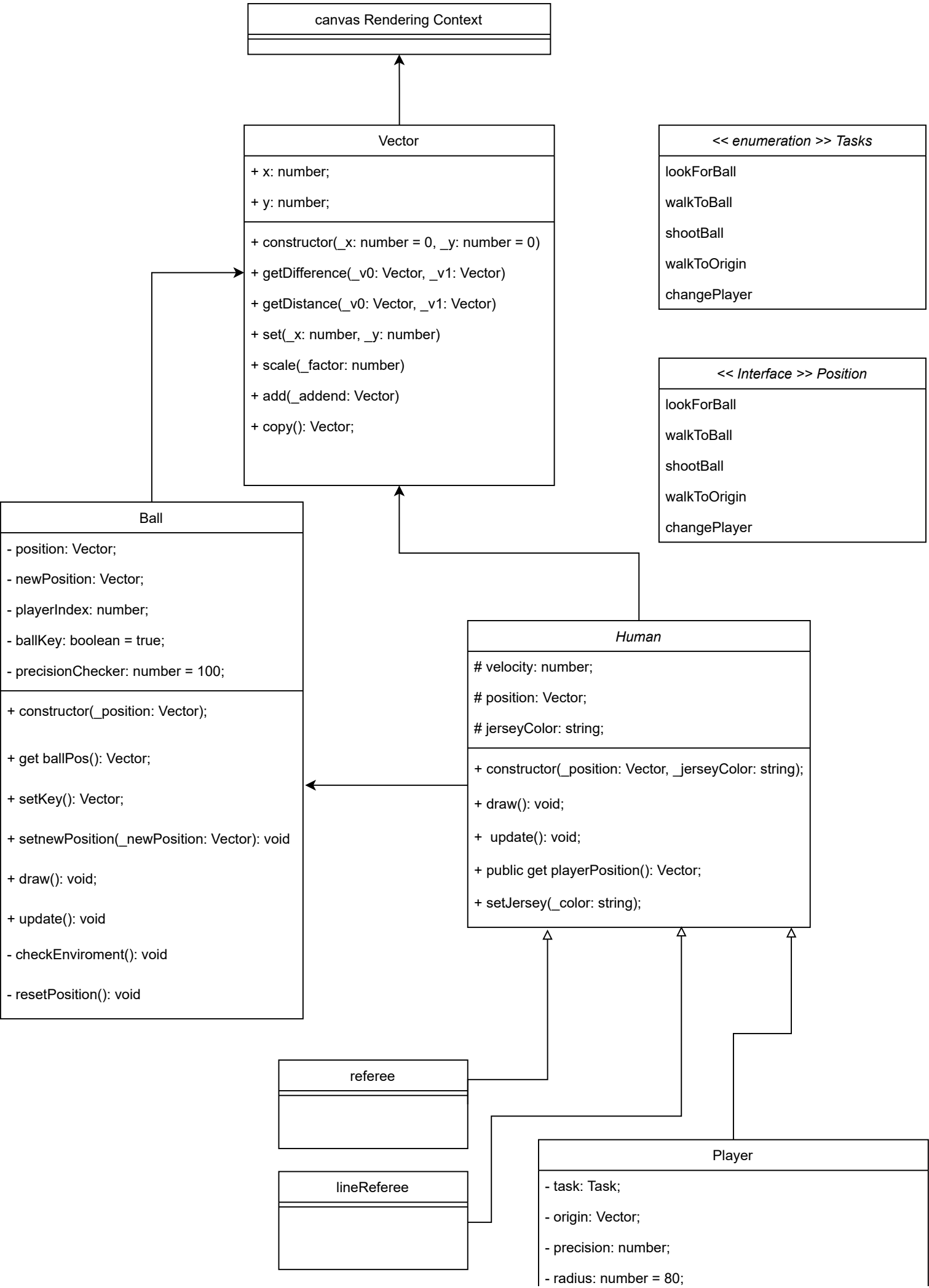
- jerseyNumber: number;

- distancePlayerBall: number;

- onField: boolean;

+ constructor(_position: Vector, _onField: boolean,
_jerseyColor: string, _team: string)

+ get jerseyNumberPlayer(): number

+ get playerSpeed(): number

+ get distance(): number

+ get playerPrecision(): number

+ get playerOnField(): number

+ get playerTeam(): number

+ setOnField(_onField): void

+ setProperties(_minSpeed: number, _maxSpeed: number,
_minPrecision: number, _maxPrecision: number): void

+ setDistance(): void

+ drawRadius(): void

- movePlayer(_position: Vector): void