

Anleitung „Soccer Simulation“

Mona Kabelka, 266238, EIA2

SoSe2021

Um den „Soccer simulator“ spielen zu können, muss primär die Zip-Datei mit einem geeigneten Programm wie WinRar oder 7Zip entpackt werden. Dort sind mehrere Ordner enthalten. Mit Doppelklick auf die HTML-Datei „soccer.html“ wird die Anwendung im Browser geöffnet und es kann losgehen.

Das Fußballspiel startet direkt zu Beginn. Auf dem Spielfeld sind 14 Spieler jedes Teams gleichmäßig verteilt, am Rande befinden sich Auswechselspieler, Schiedsrichter und Linienrichter.

Im Formular links „Player adjustment“ können die Werte aller Spieler angepasst werden, bzw. der Bereich, in welchem zufällige Werte für die Spieler ausgegeben werden. So kann während des Spiels bestimmt werden, wie „schnell“ oder „präzise“ die Teams spielen. Außerdem können die Trikotfarben angepasst werden.

Im rechten Formular können die einzelnen Werte jedes Spielers ausgegeben werden, wobei man mit den Pfeiltasten durch die einzelnen Spieler navigieren kann. In diesem Formular können auch die Spieler mit den Auswechselspielern des Teams getauscht werden. Ausgetauschte Spieler tauschen während der Laufzeit (wenn der Ball rollt) an ihre neue Position.

Wenn ein Spieler den Ball erreicht, wird die Zeit angehalten und Du kannst auf einen beliebigen Punkt auf dem Spielfeld klicken, um den Ball zu schießen. Spieler in der Nähe bewegen sich zum Ball und die Zeit bleibt erneut stehen, wenn ein Spieler den Ball erreicht hat. Ist der Ball außerhalb ihres Spielradius, laufen die Spieler wieder an ihre Ursprungsposition zurück. Wird ein Tor erzielt, signalisiert ein WindowAlert den Erfolg. Mit klick auf „Ok“ wird das Spiel erneut angepiffen.

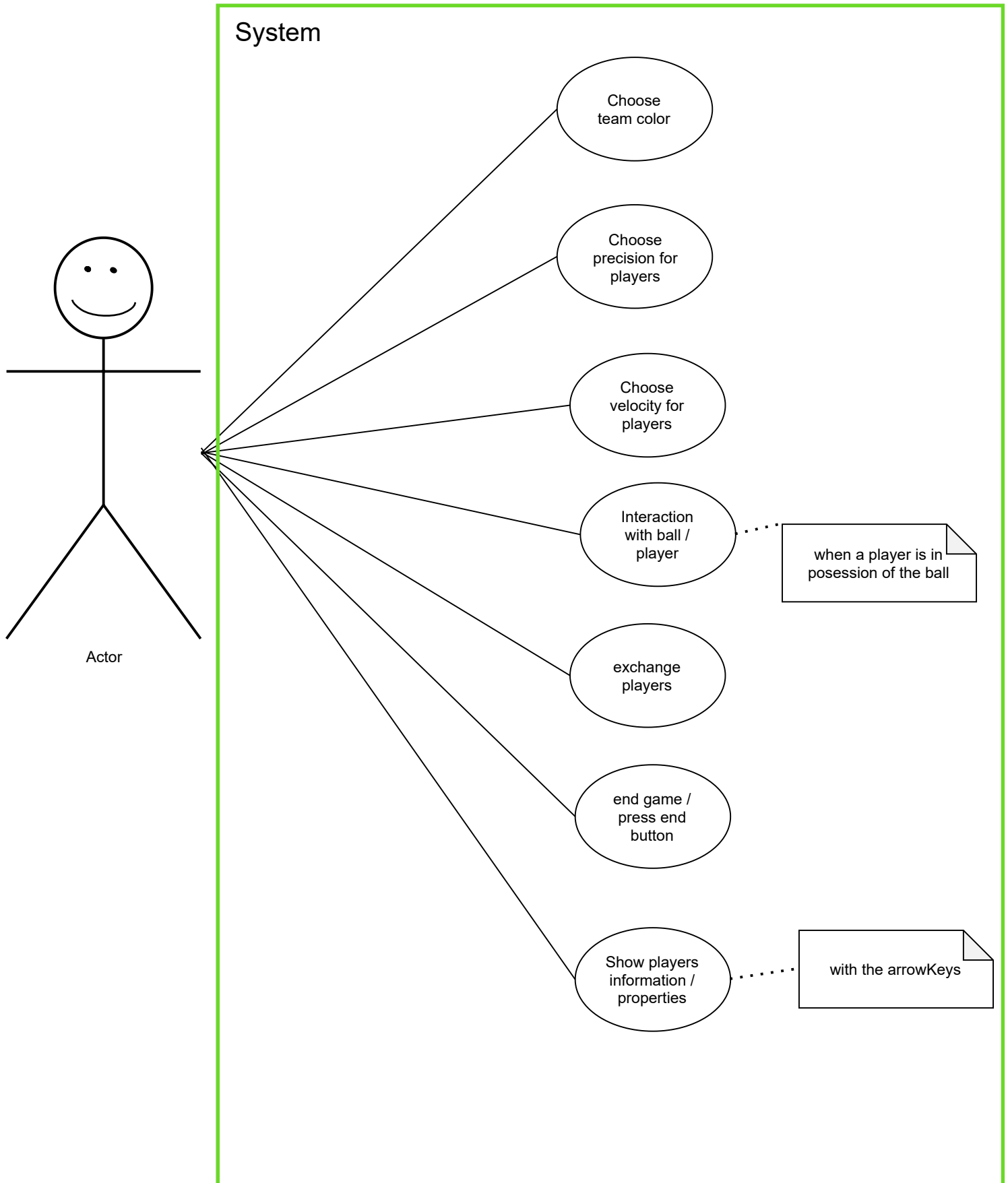
Der Spielstand und der Spieler im Ballbesitz werden im Display oberhalb des Spielfeldes angezeigt.

Um das Spiel neu zu starten einfach auf den „Restart“ Button unterhalb des Spielfeldes klicken.

Das Konzept und der Code entstand in Zusammenarbeit mit:

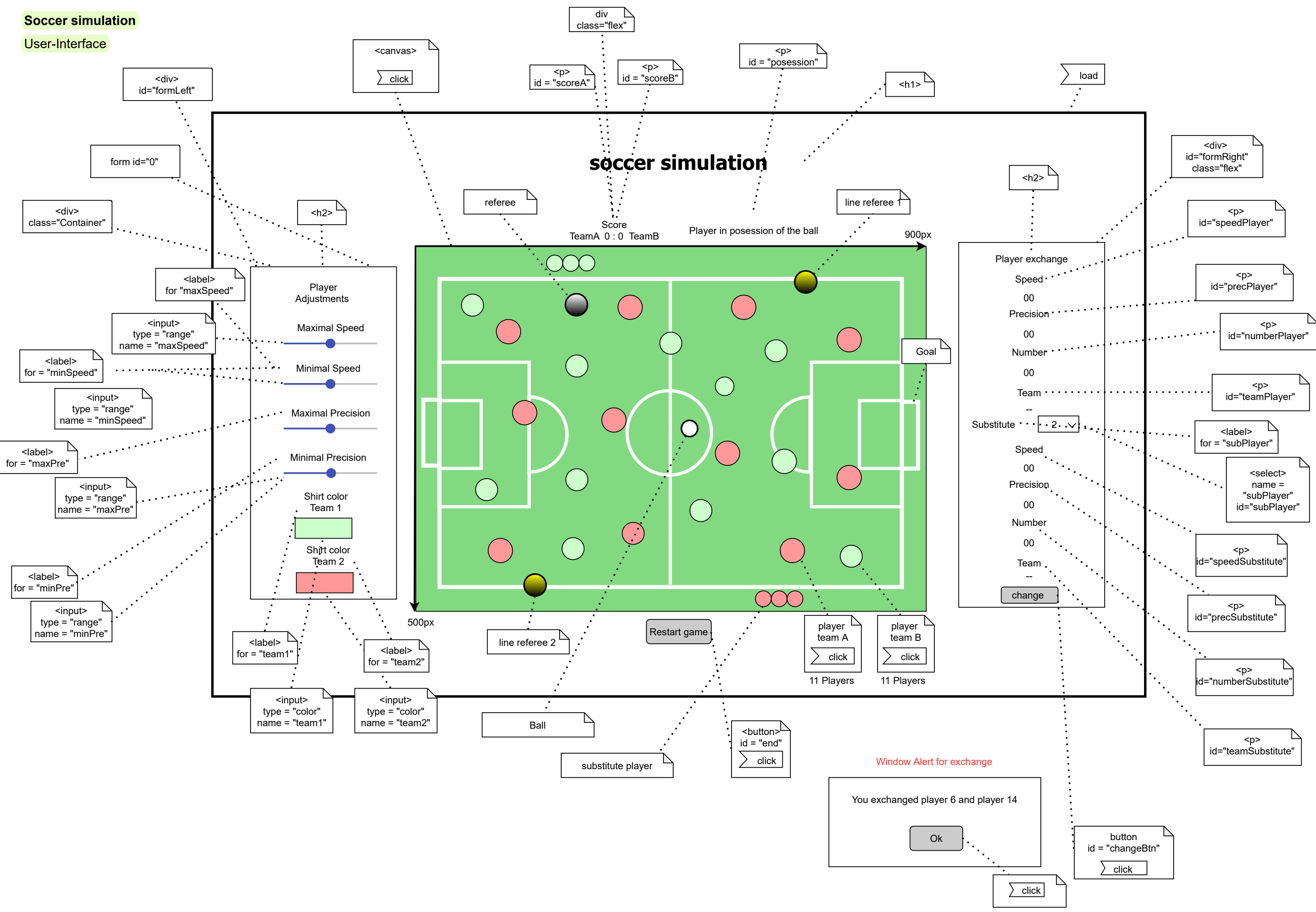
Huu Thien Phan Ngoc, Mariia Kolkutova, Christina Däschner und Timur Yildirim

Soccer Simulation: Use-Case-Diagram

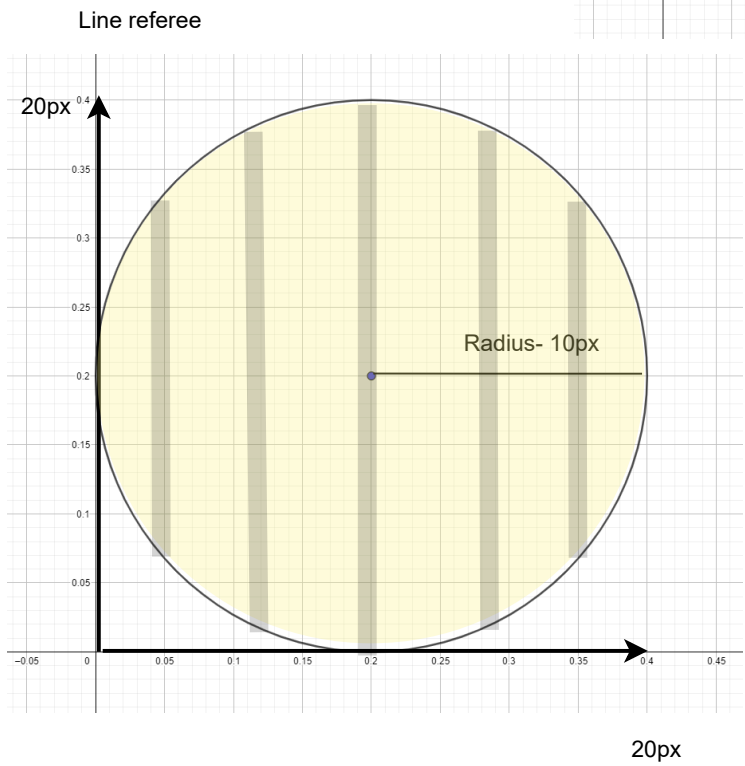
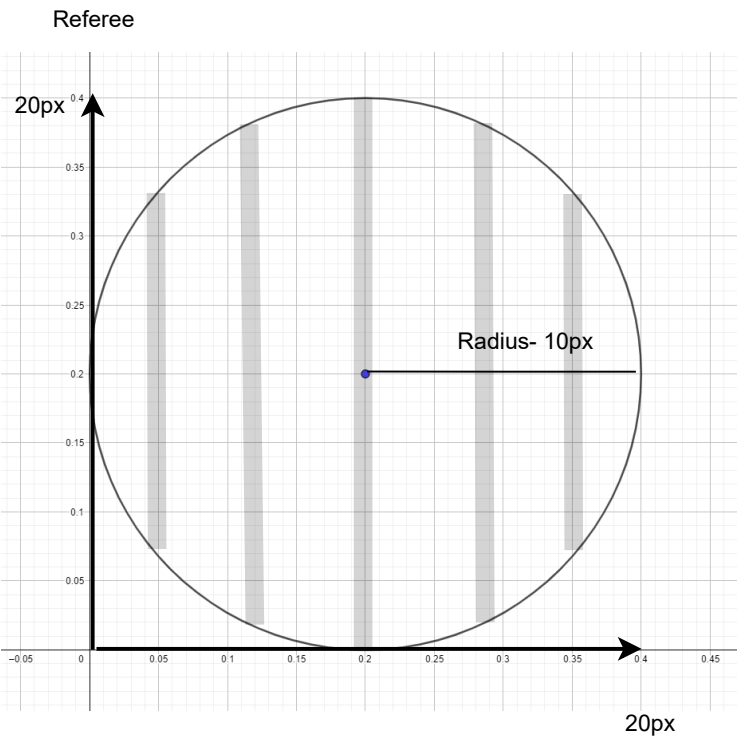
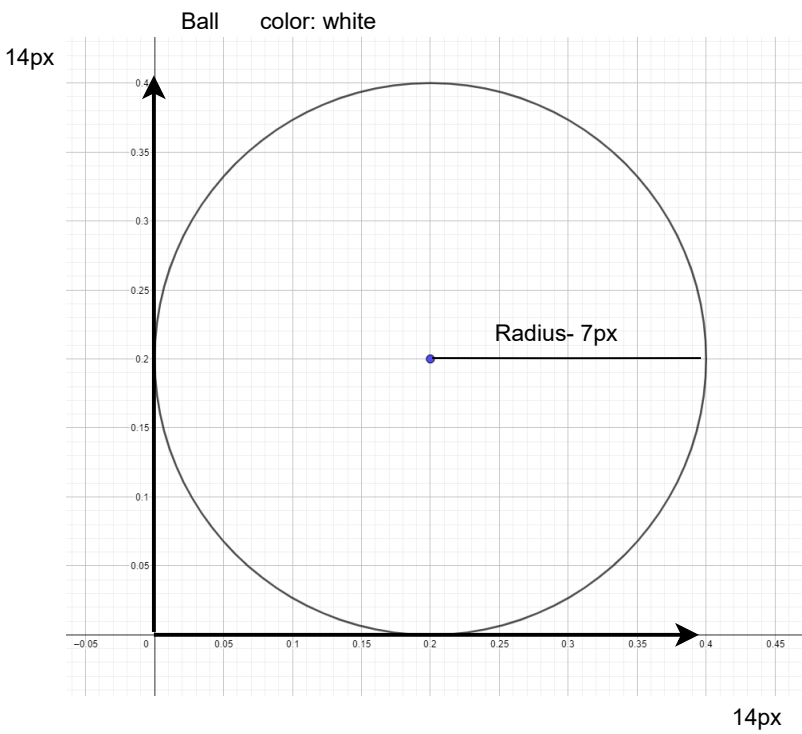
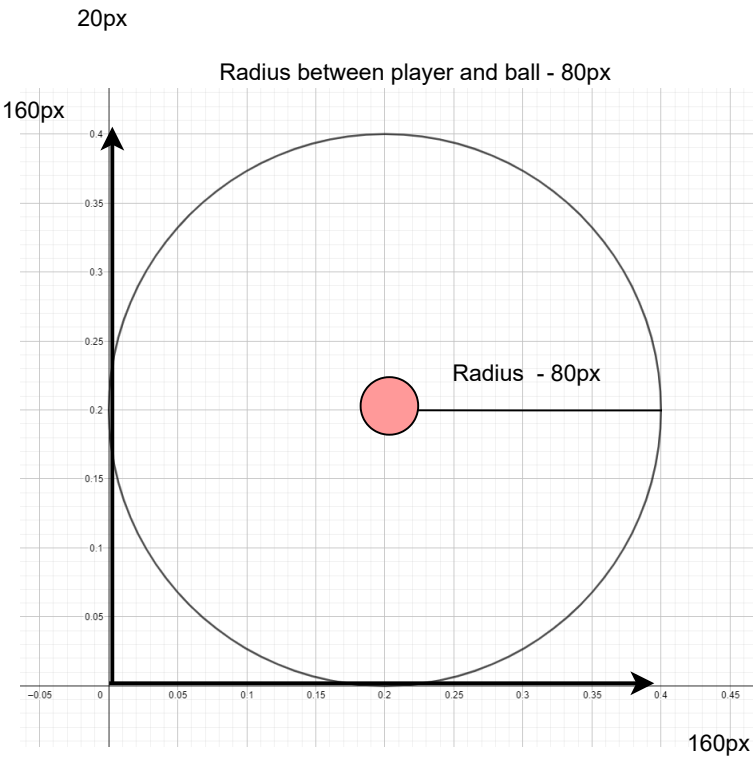
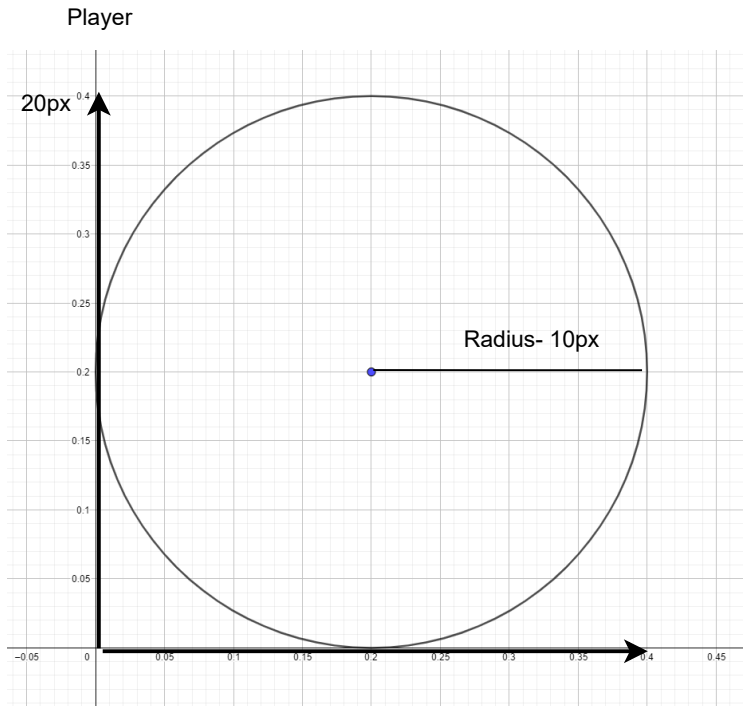
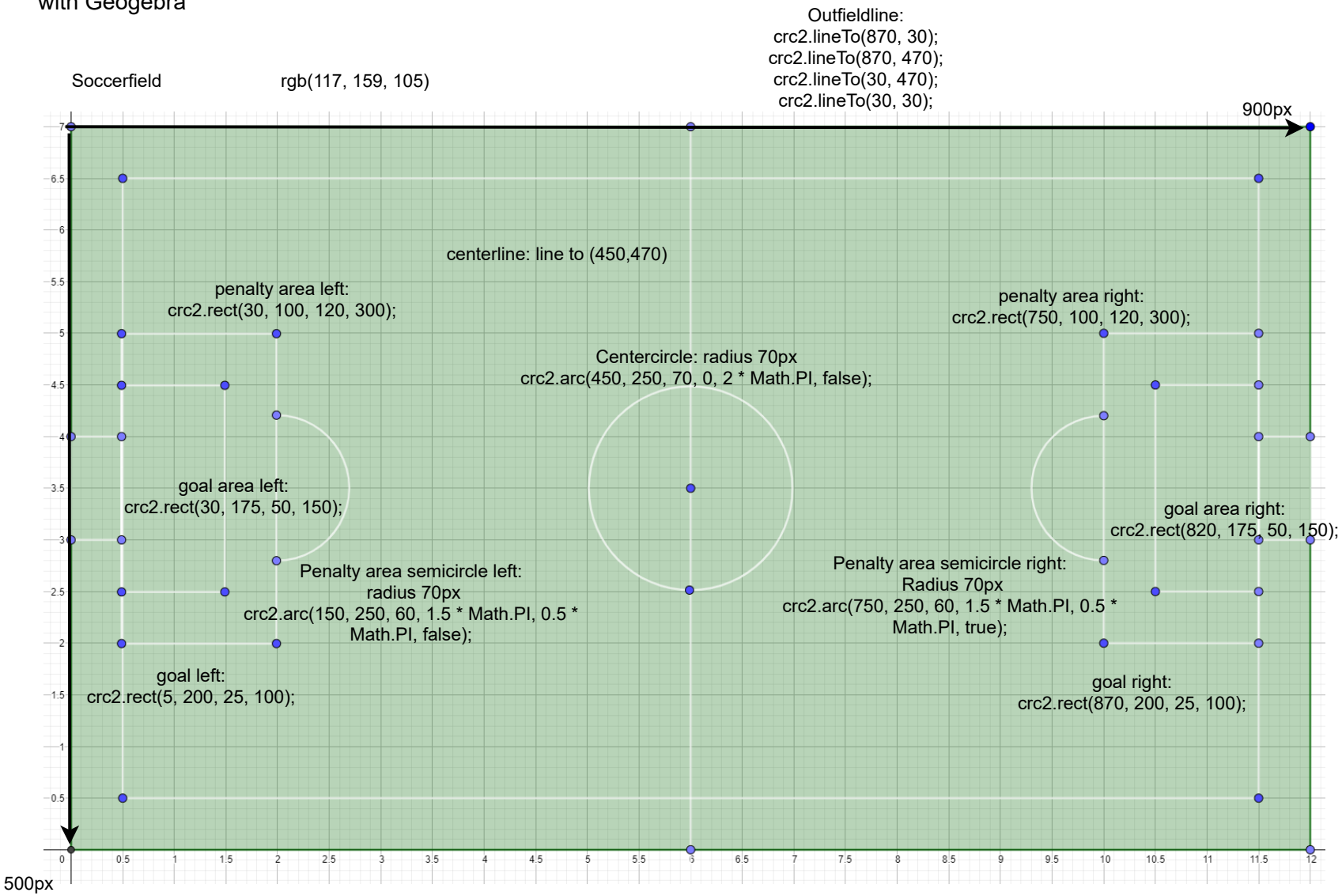


Soccer simulation

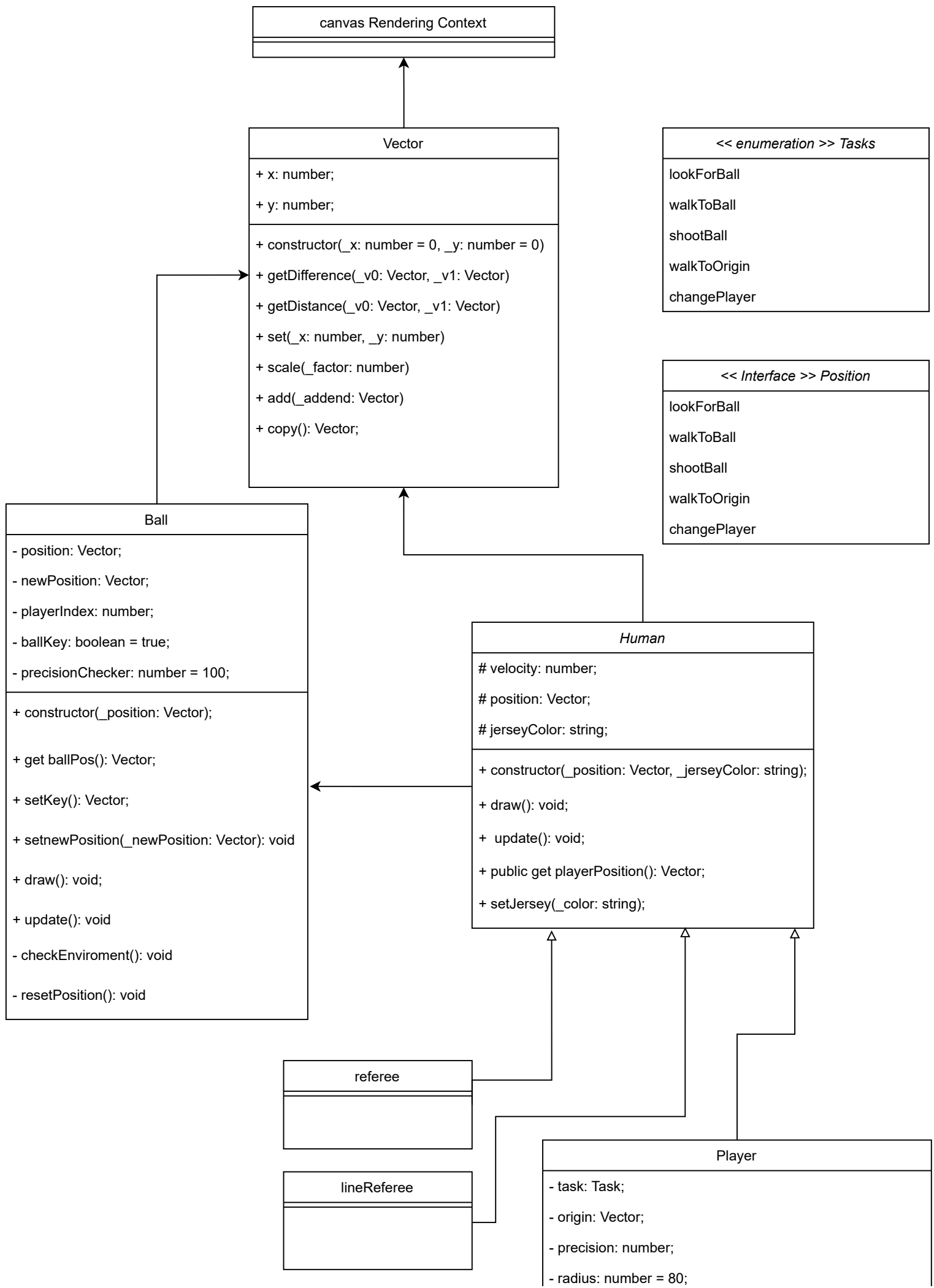
User-Interface



Measurements
with Geogebra



Classdiagram



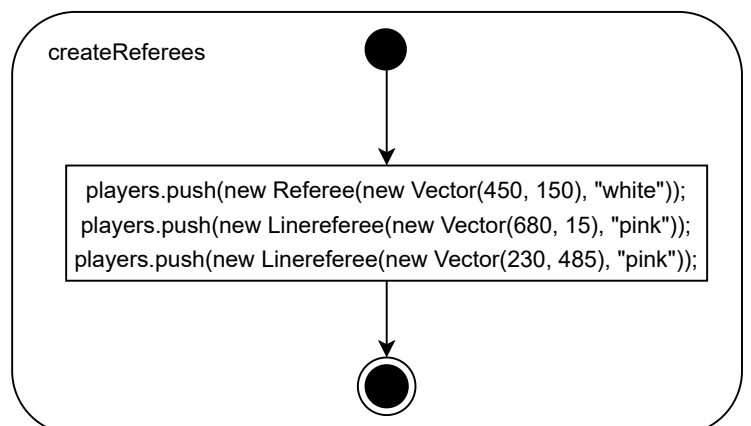
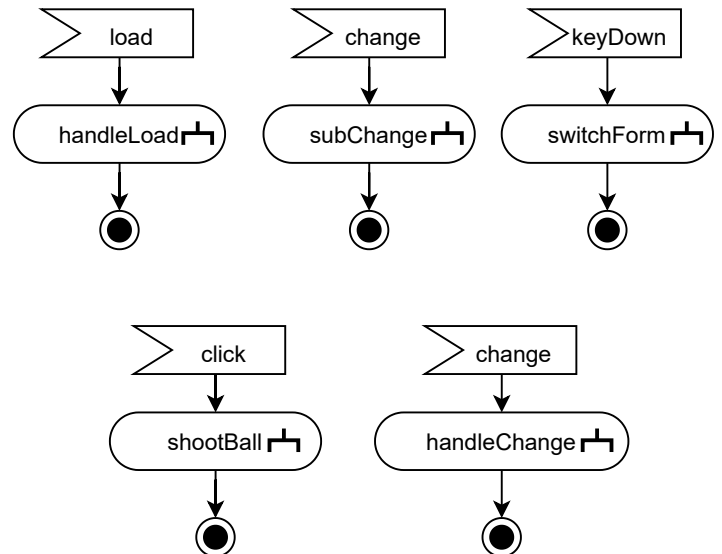
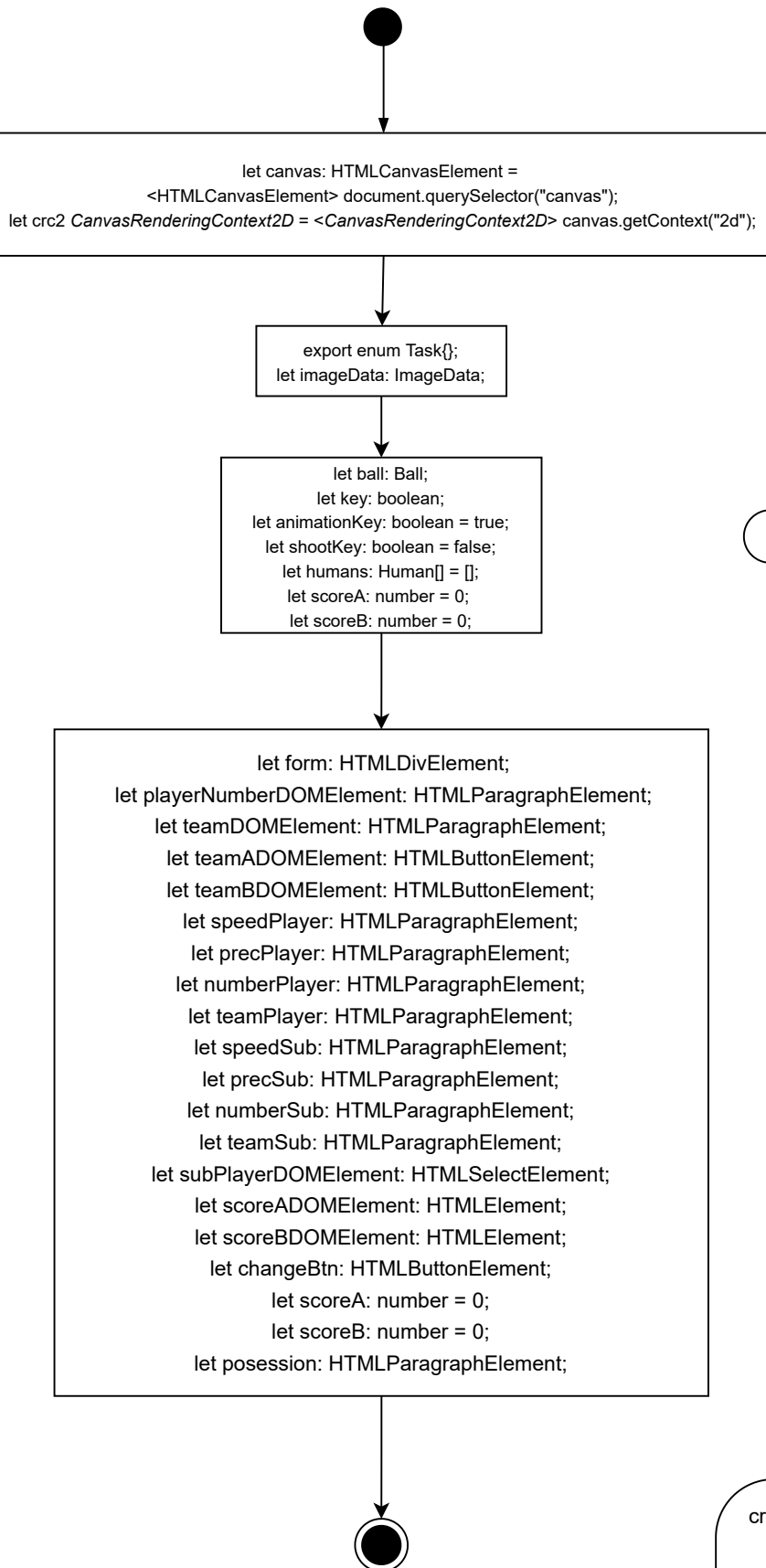
- jerseyNumber: number;
- distancePlayerBall: number;
- onField: boolean;

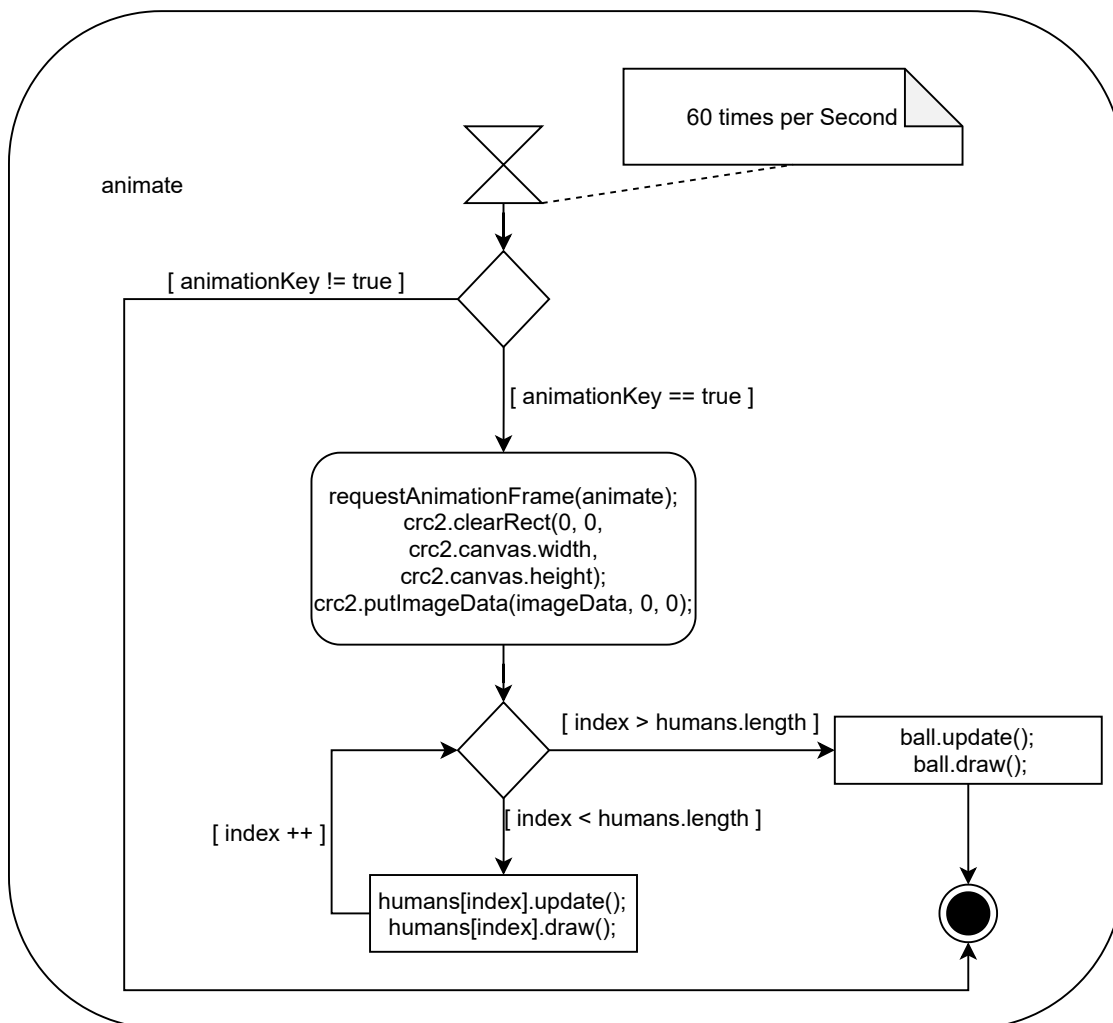
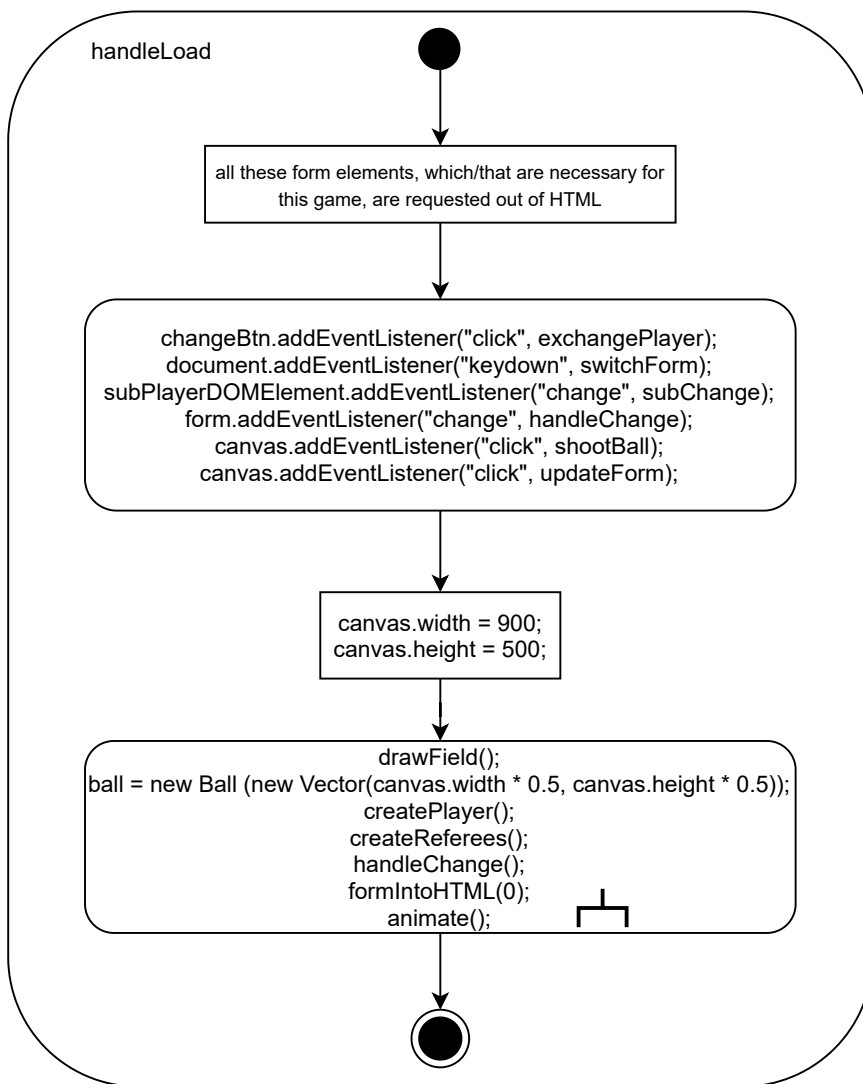
+ constructor(_position: Vector, _onField: boolean, _jerseyColor: string, _team: string)
+ get jerseyNumberPlayer(): number
+ get playerSpeed(): number
+ get distance(): number
+ get playerPrecision(): number
+ get playerOnField(): number
+ get playerTeam(): number
+ setOnField(_onField): void
+ setProperties(_minSpeed: number, _maxSpeed: number, _minPrecision: number, _maxPrecision: number): void
+ setDistance(): void
+ drawRadius(): void
- movePlayer(_position: Vector): void

Activitydiagram: Main

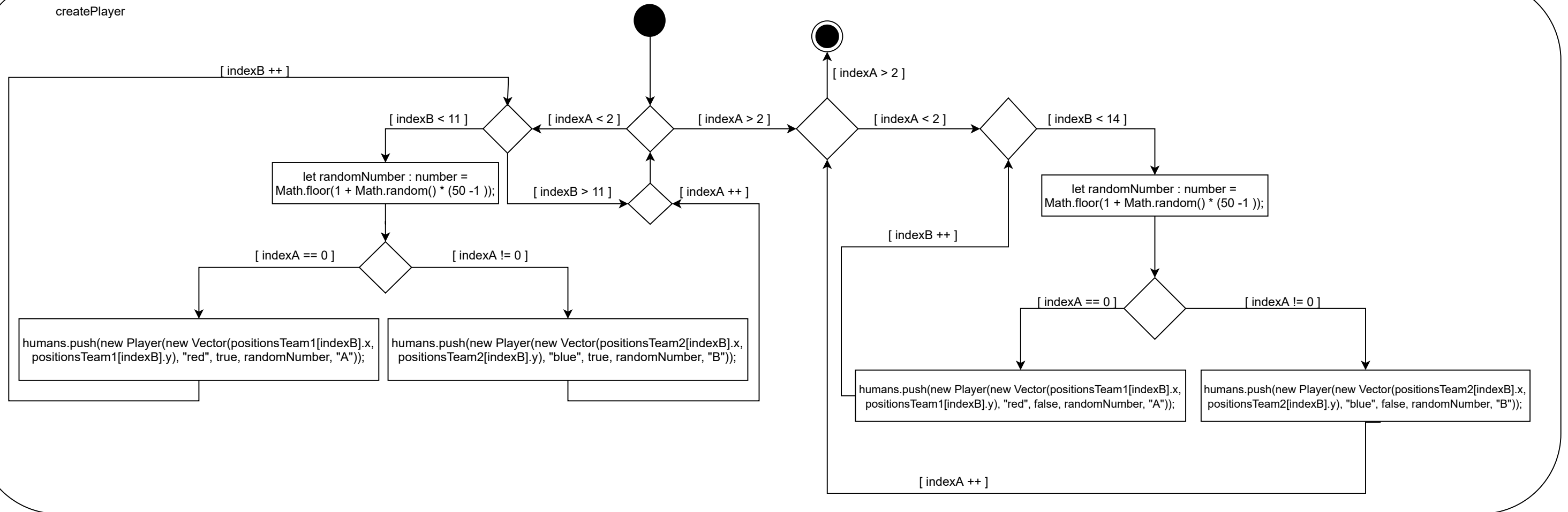
Mona Kabelka

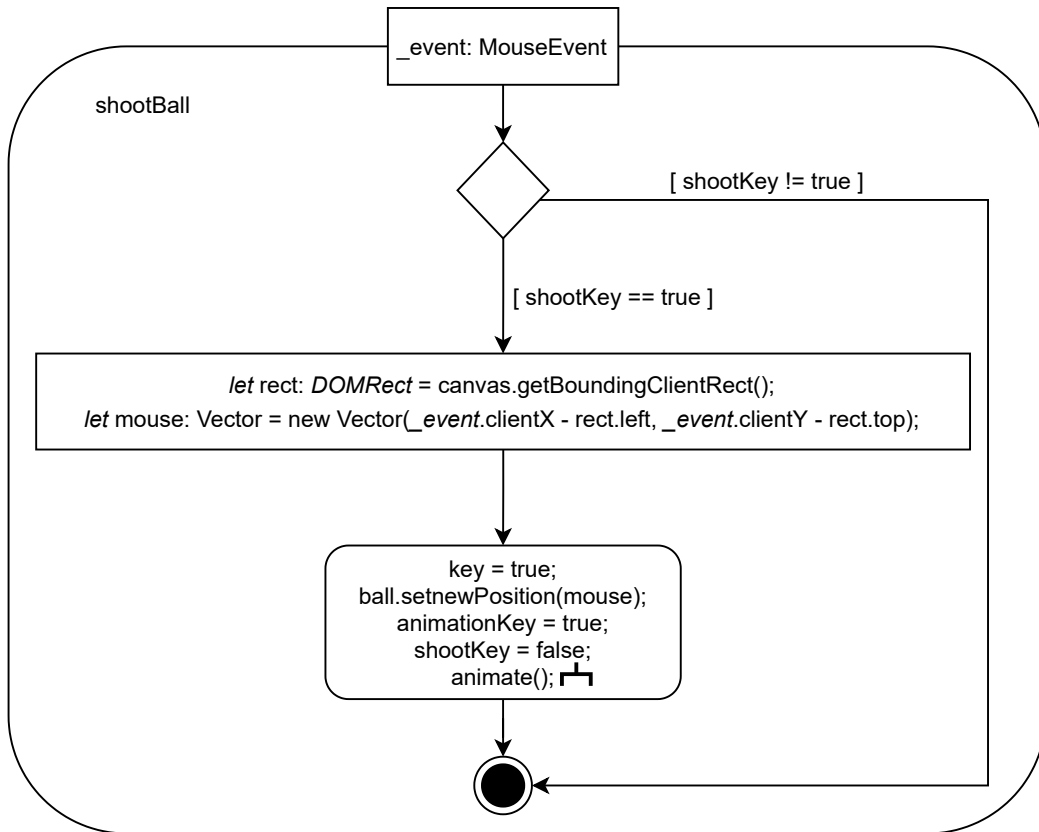
This concept was developed in cooperation with Huu Thien Phan Ngoc, Mariia Kolkutova, Christina Däschner and Timur Yildirim



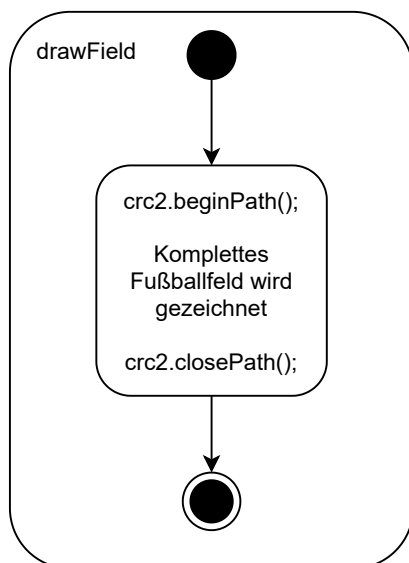


createPlayer

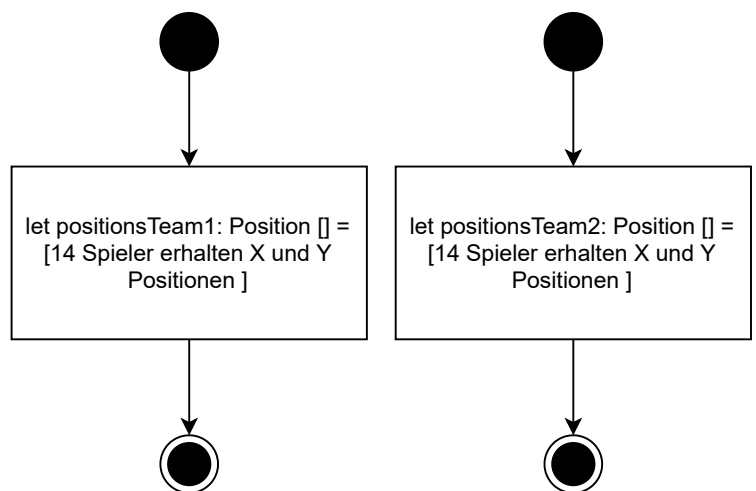




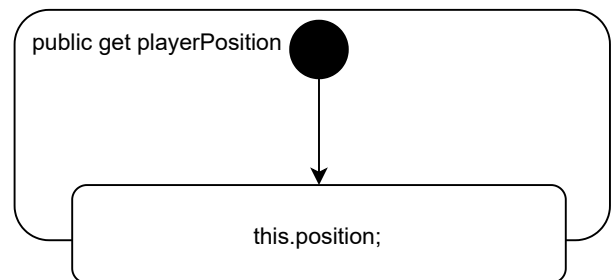
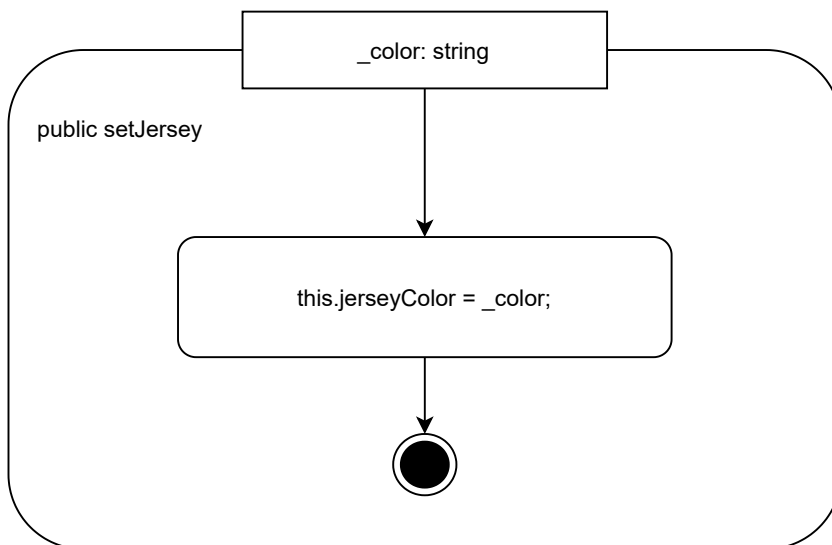
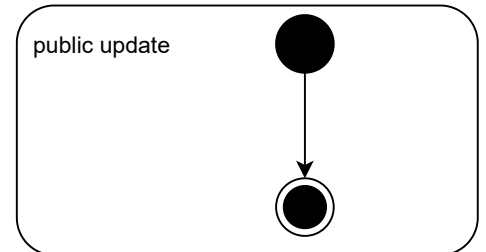
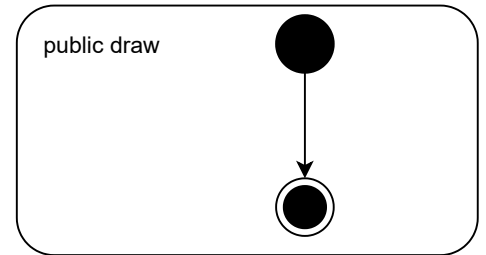
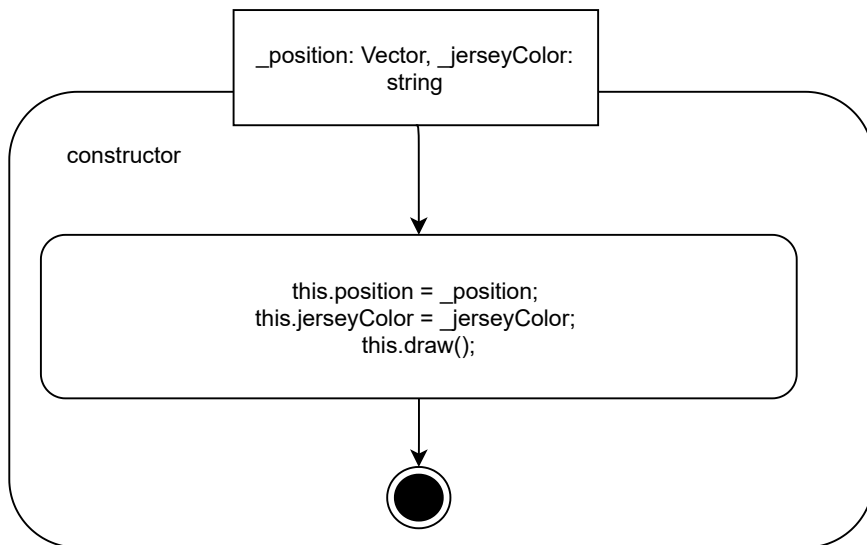
Activitydiagram: footballField



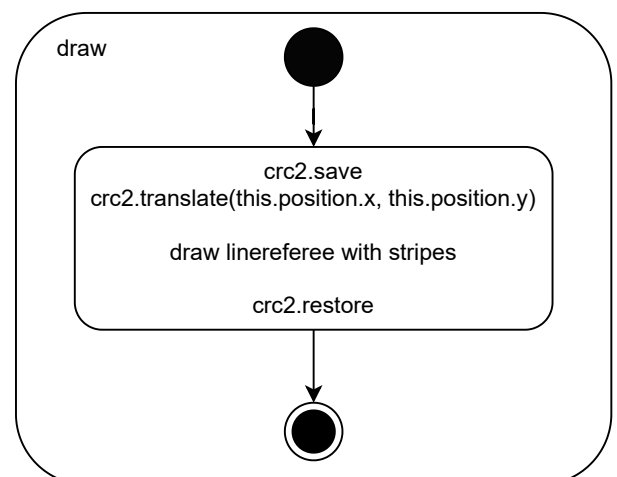
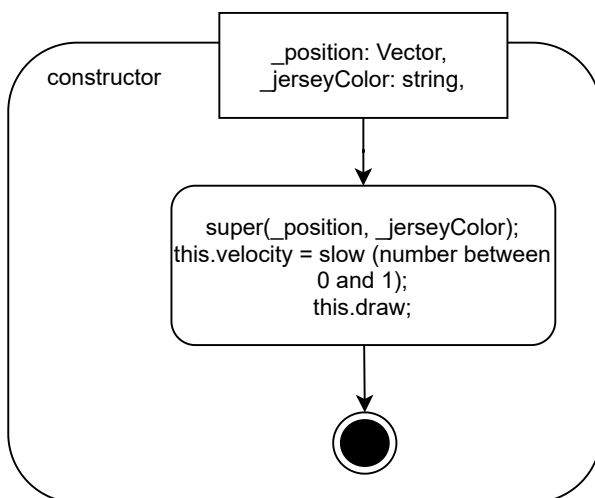
Activitydiagram: position

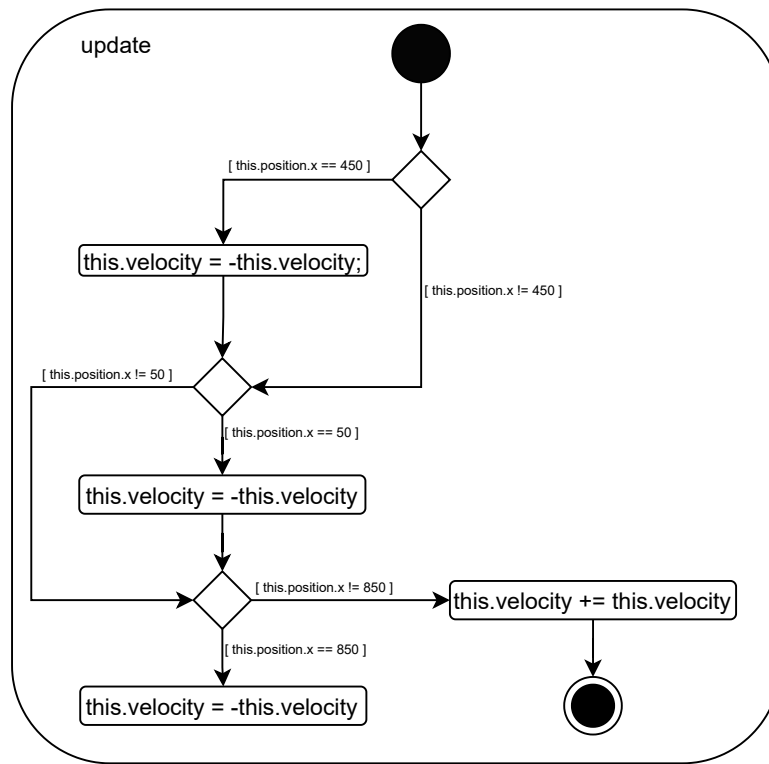


Activitydiagram: human

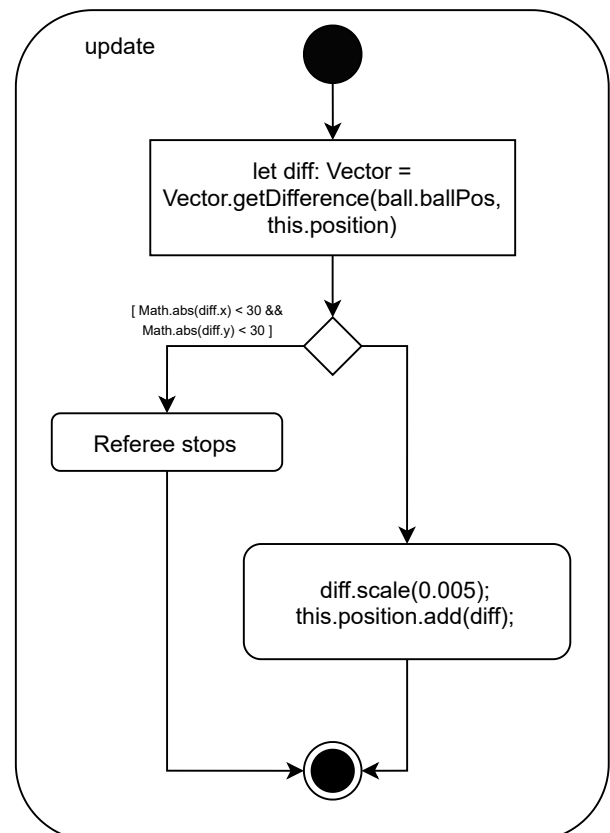
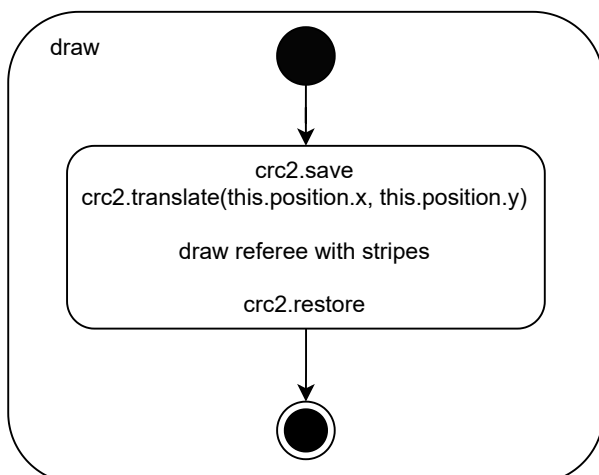
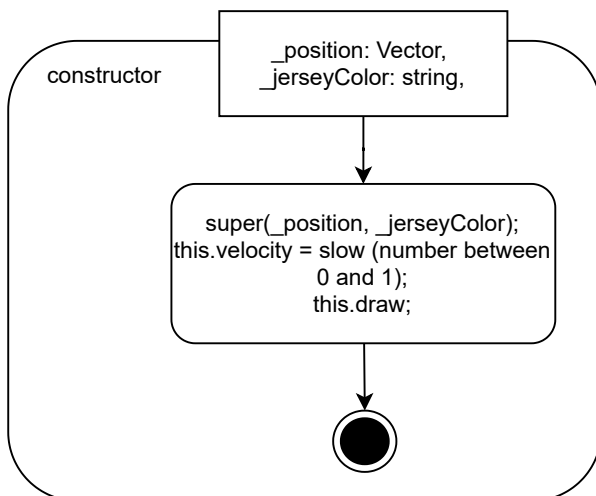


Activitydiagram: lineReferee

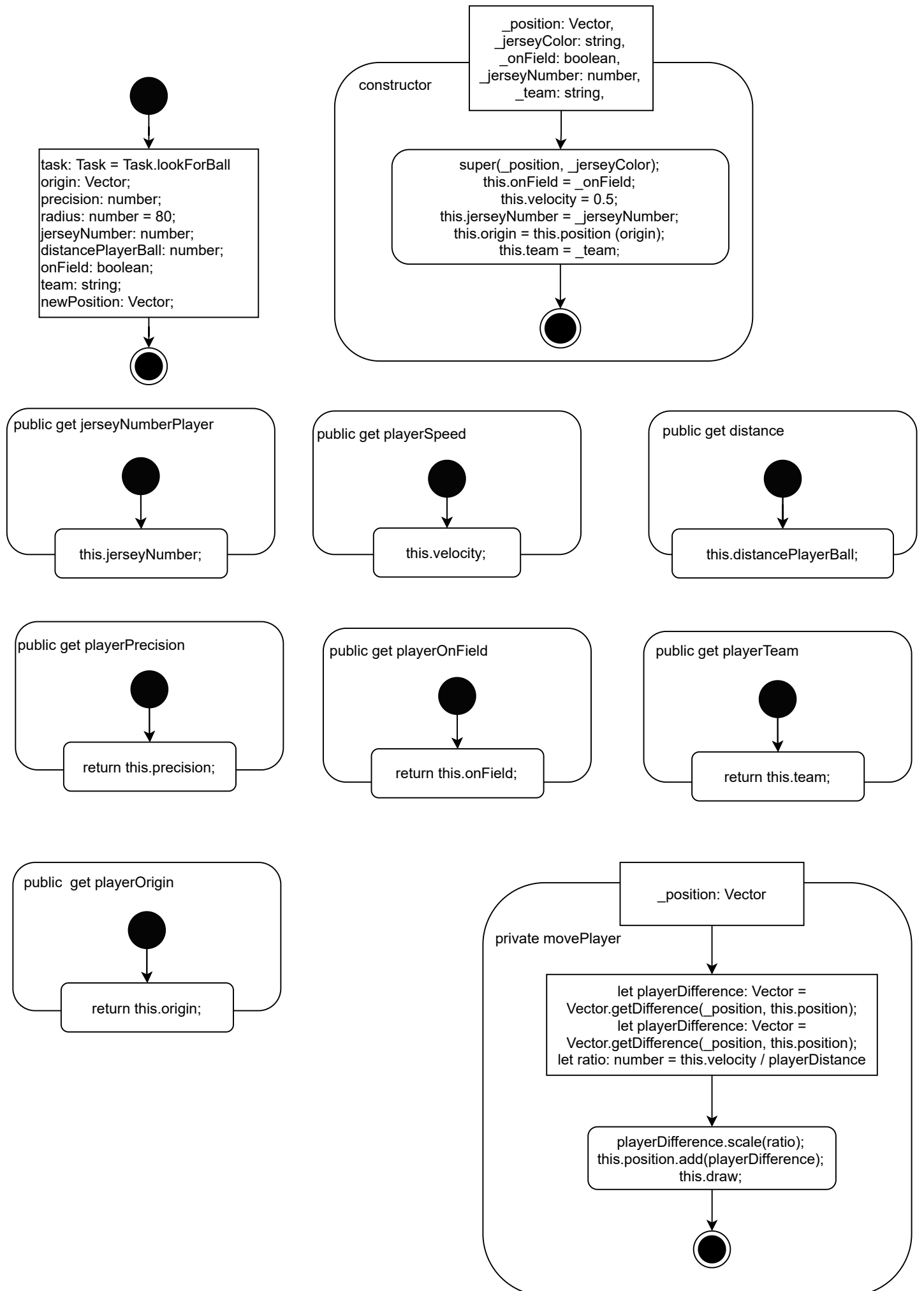


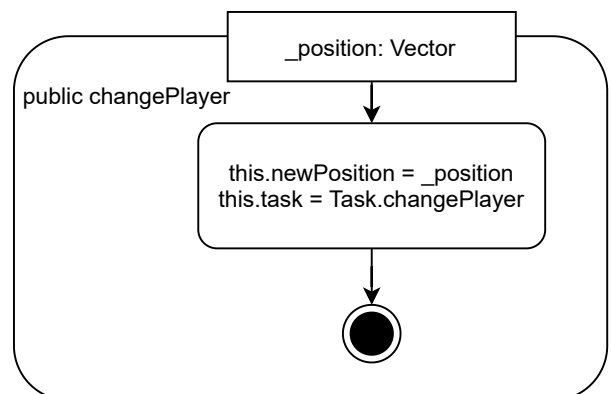
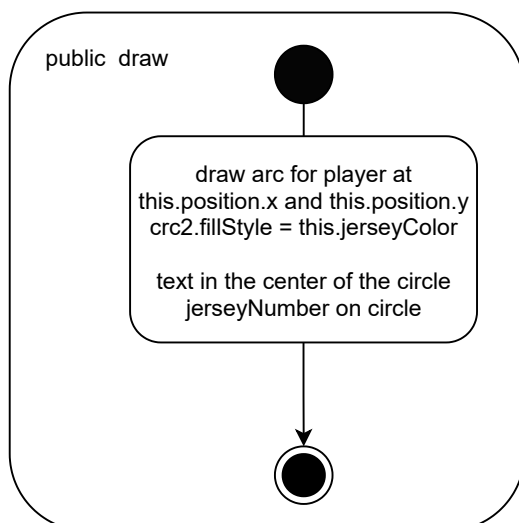
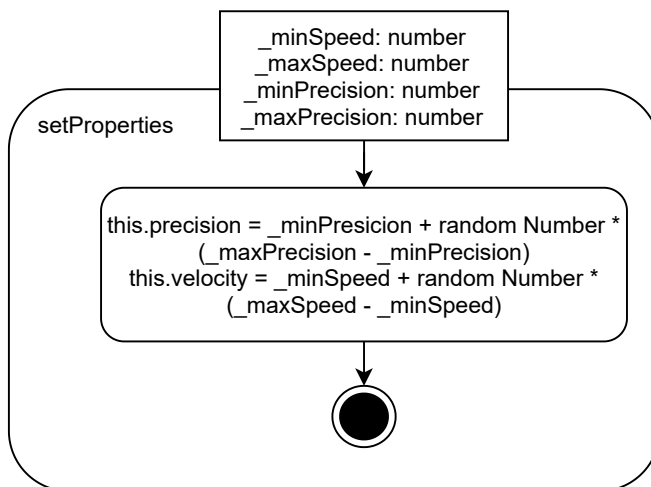
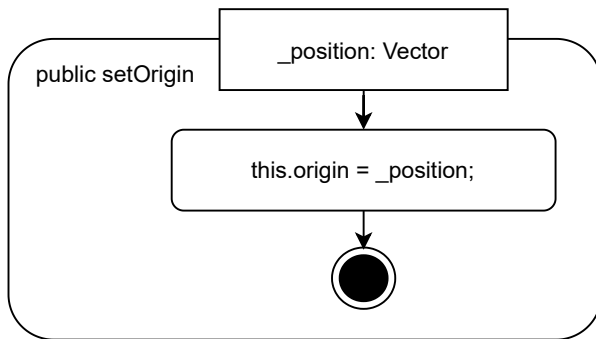
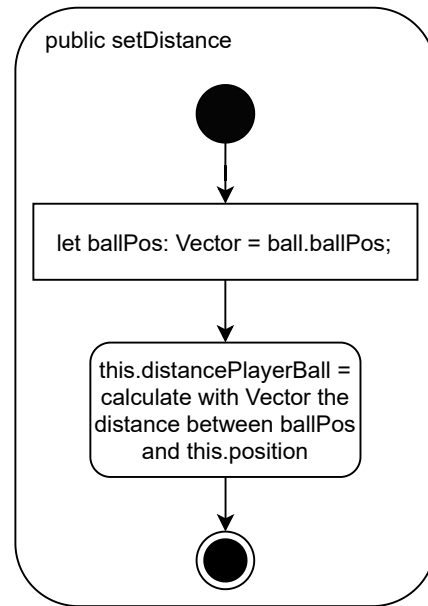
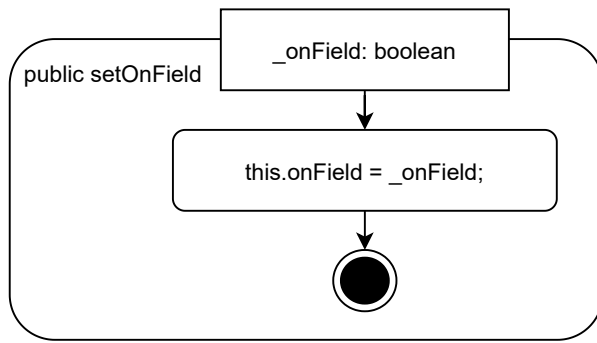


Activitydiagram: Referee

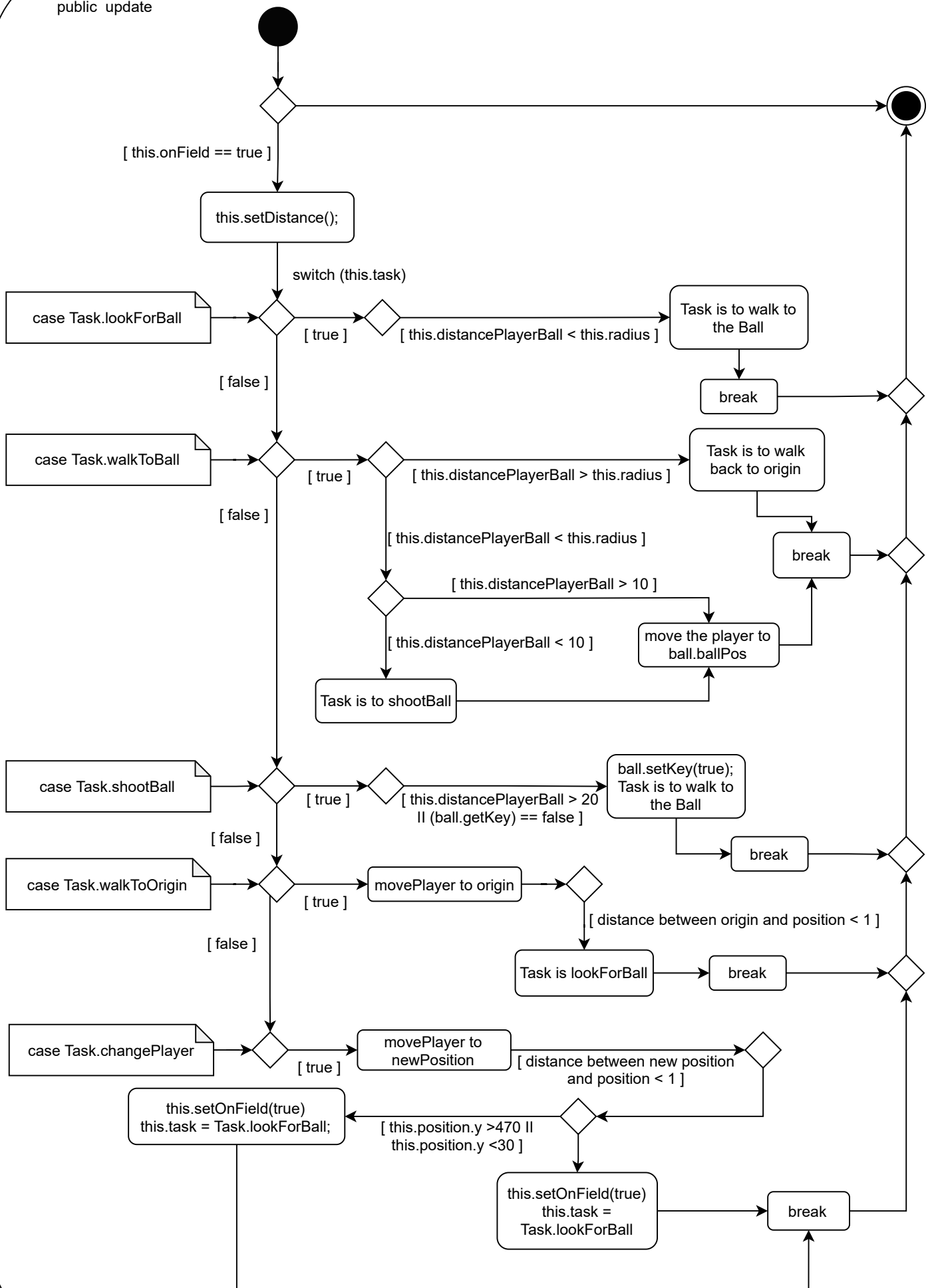


Activitydiagram: Player

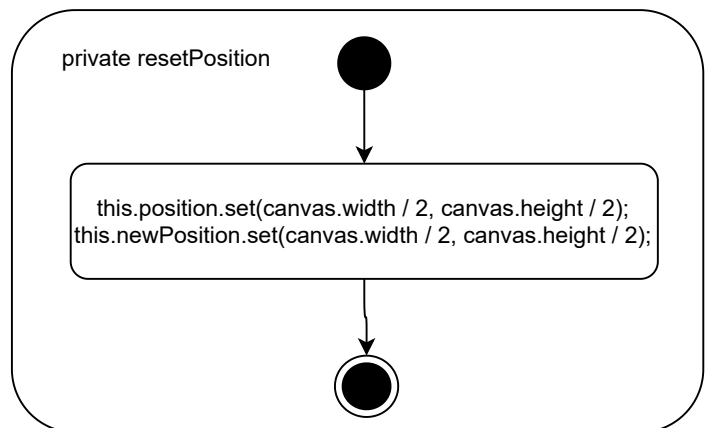
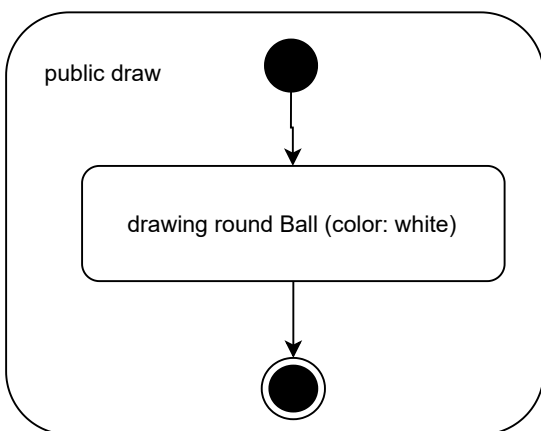
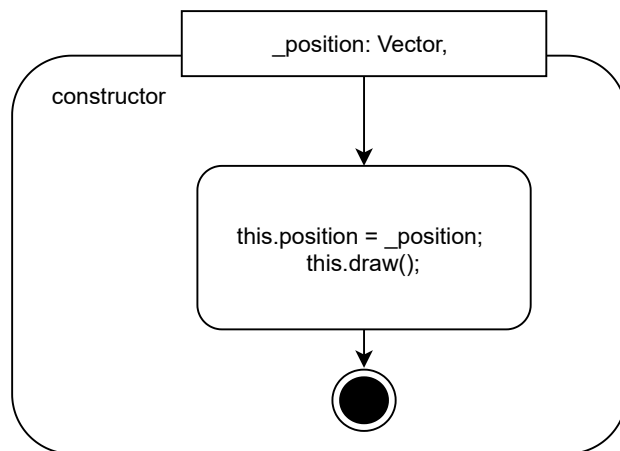
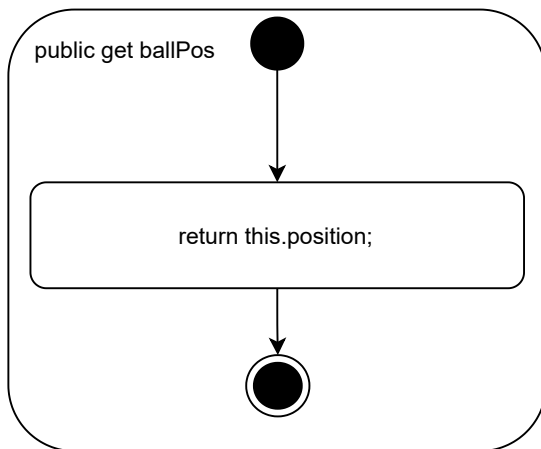
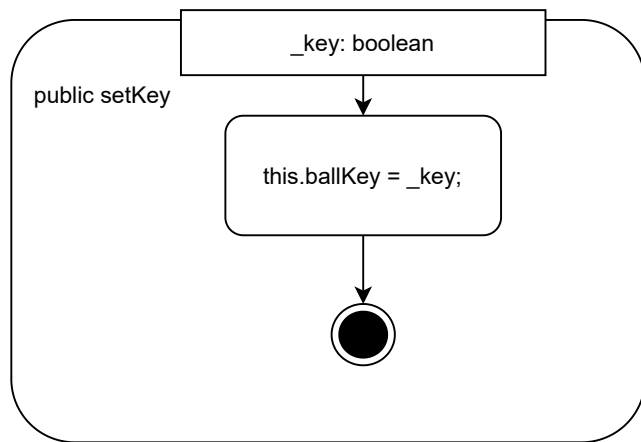
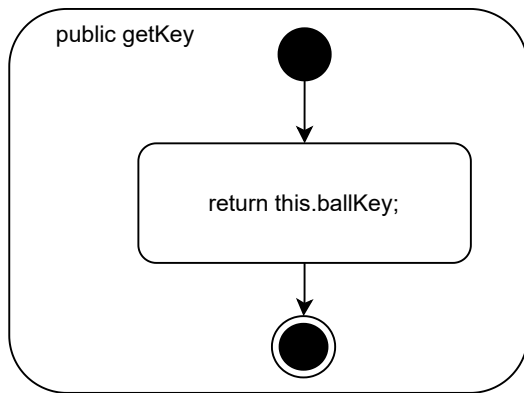


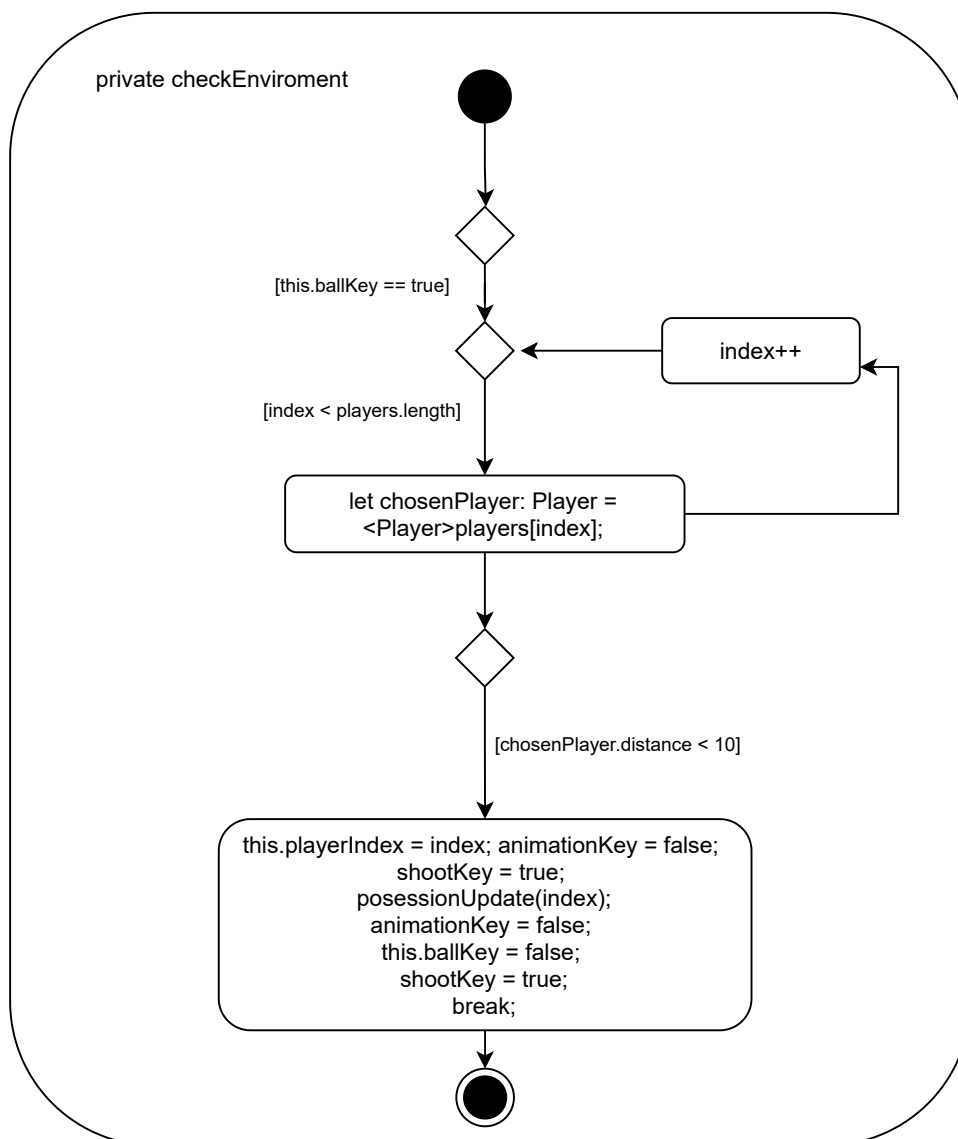
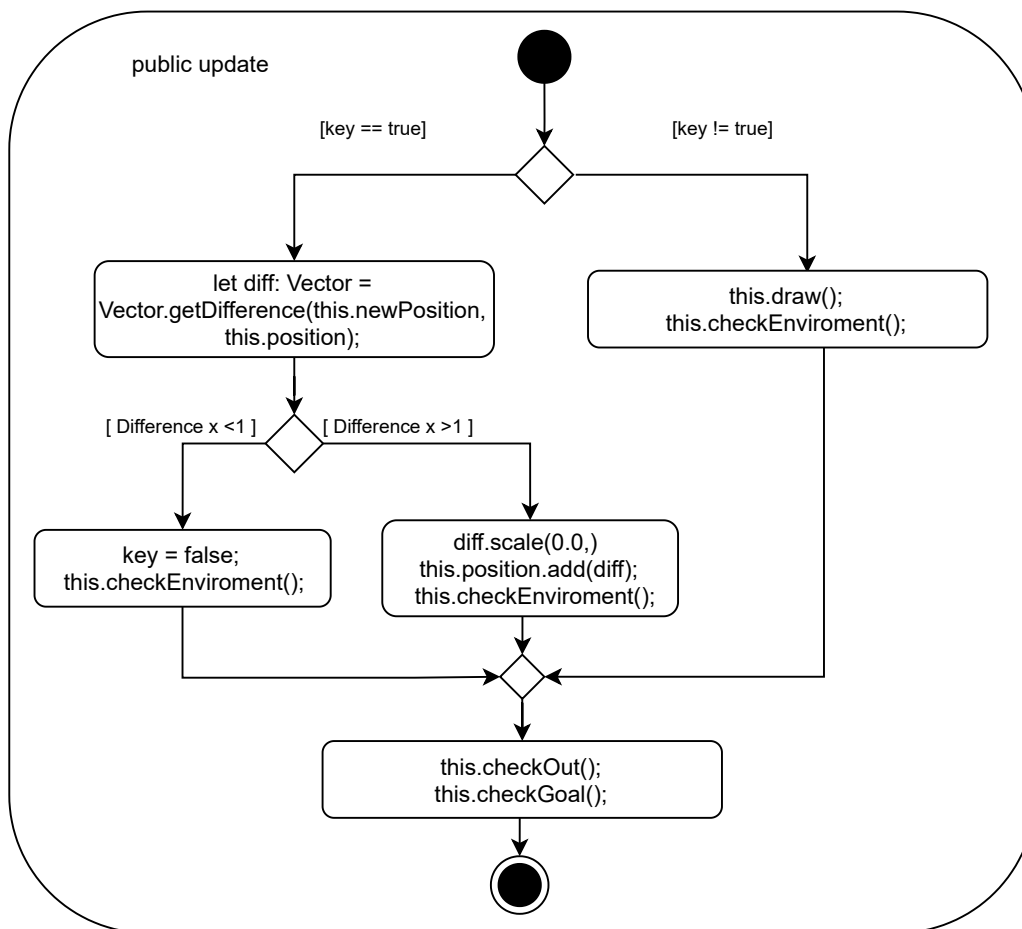


public update

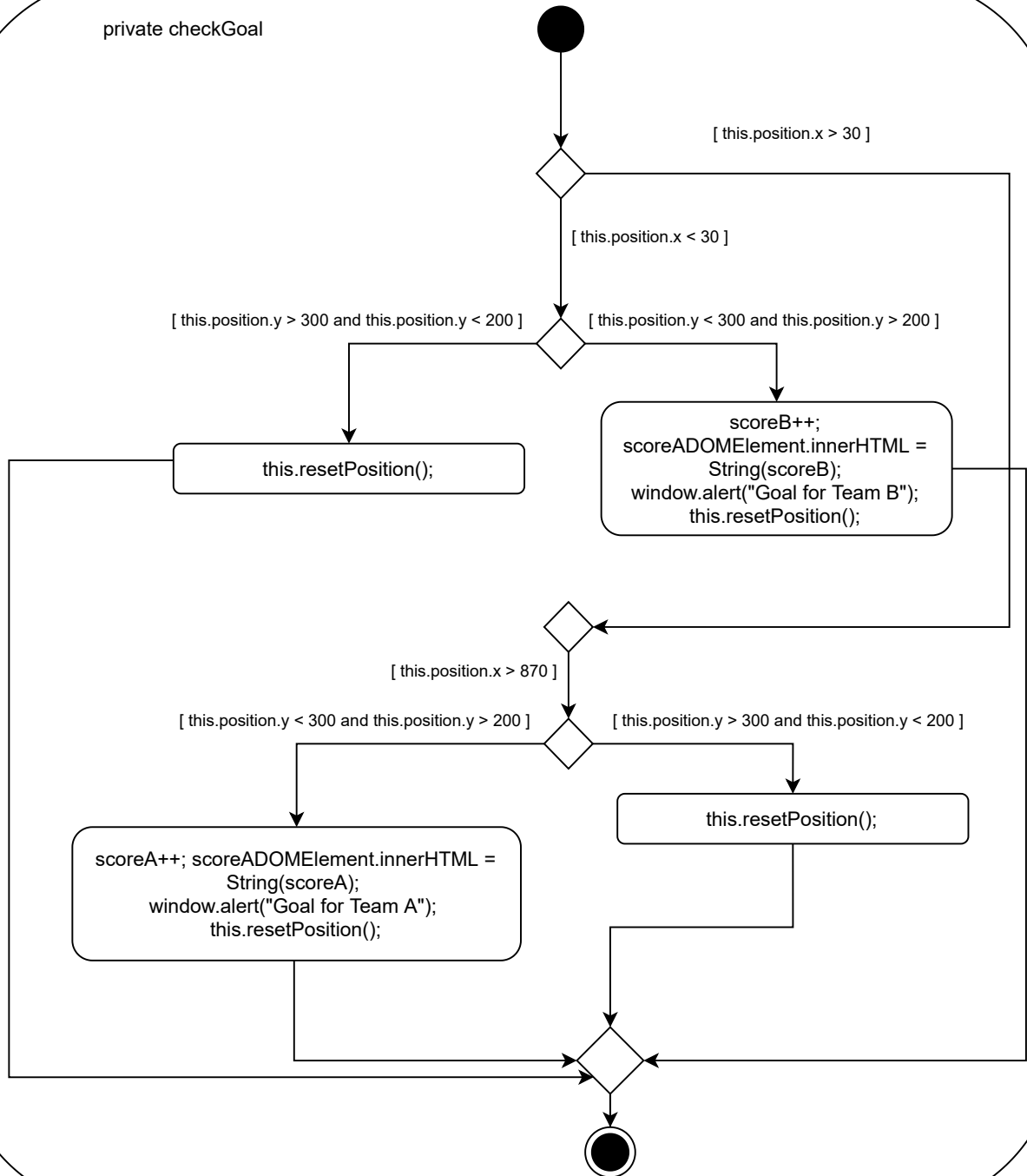


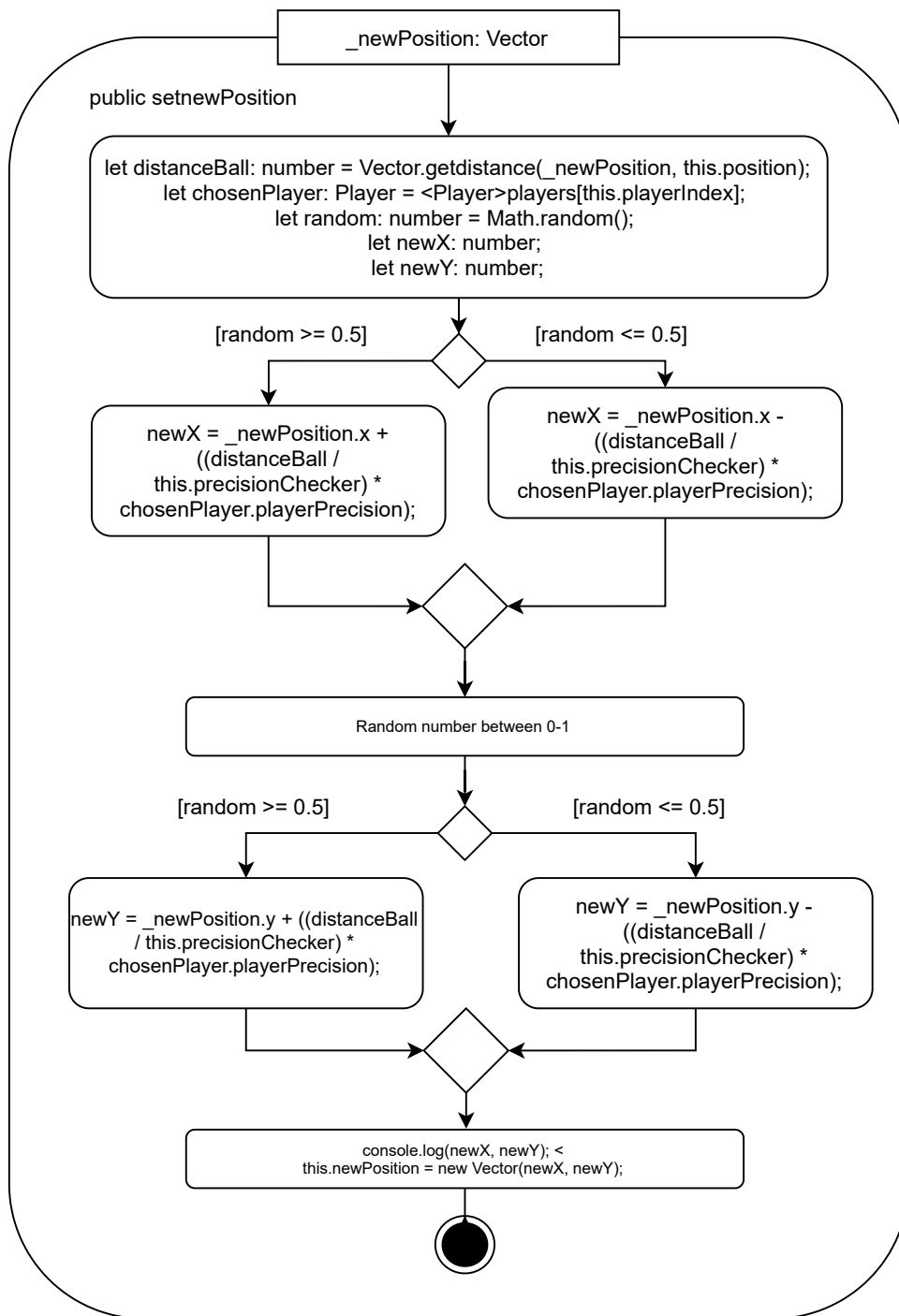
Aktivitätsdiagramm: Ball



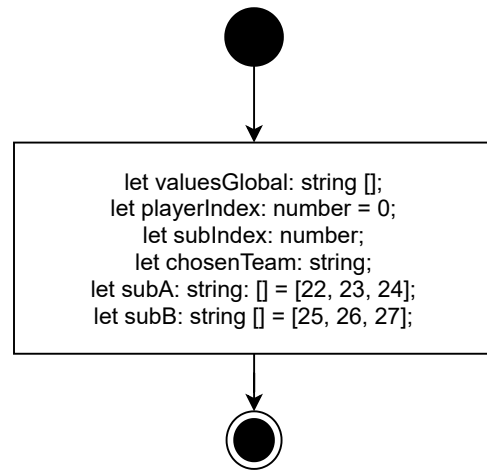


private checkGoal

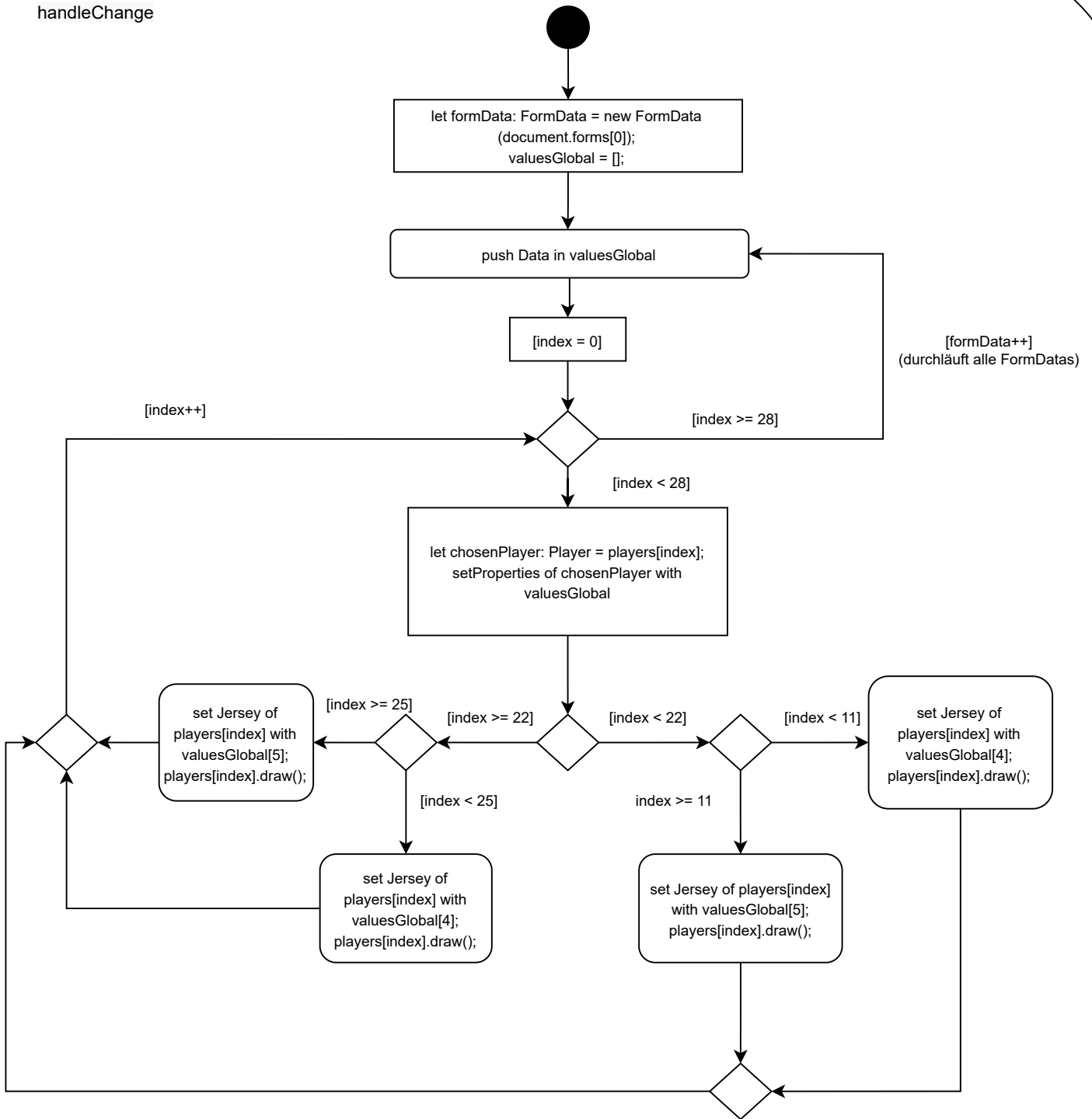


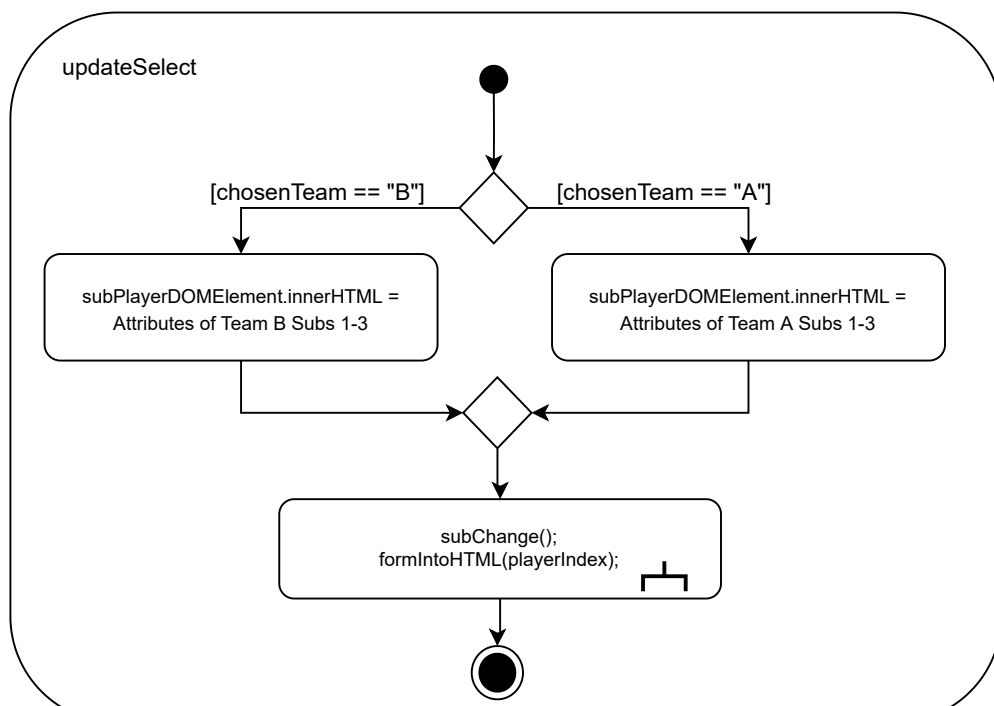
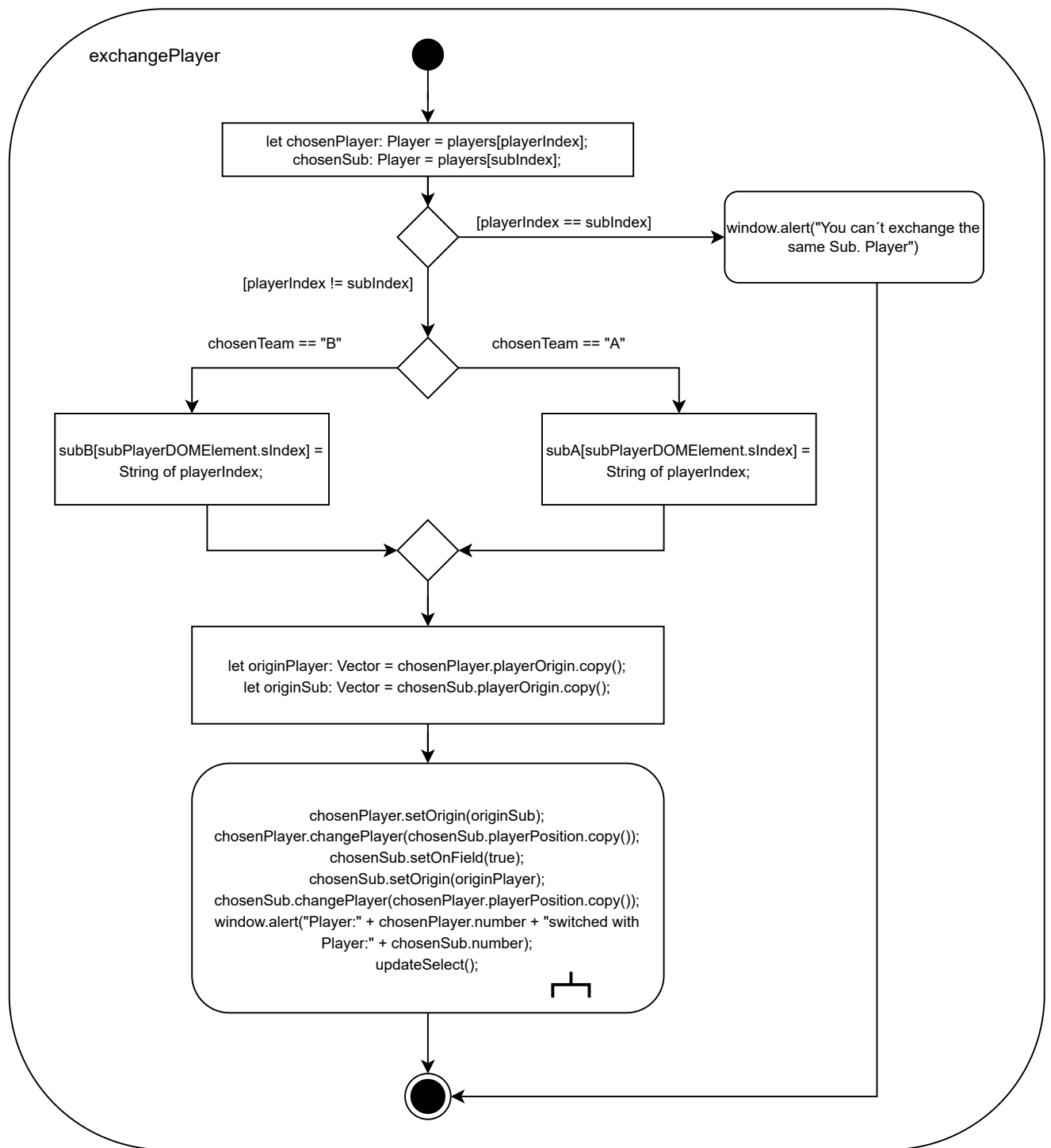


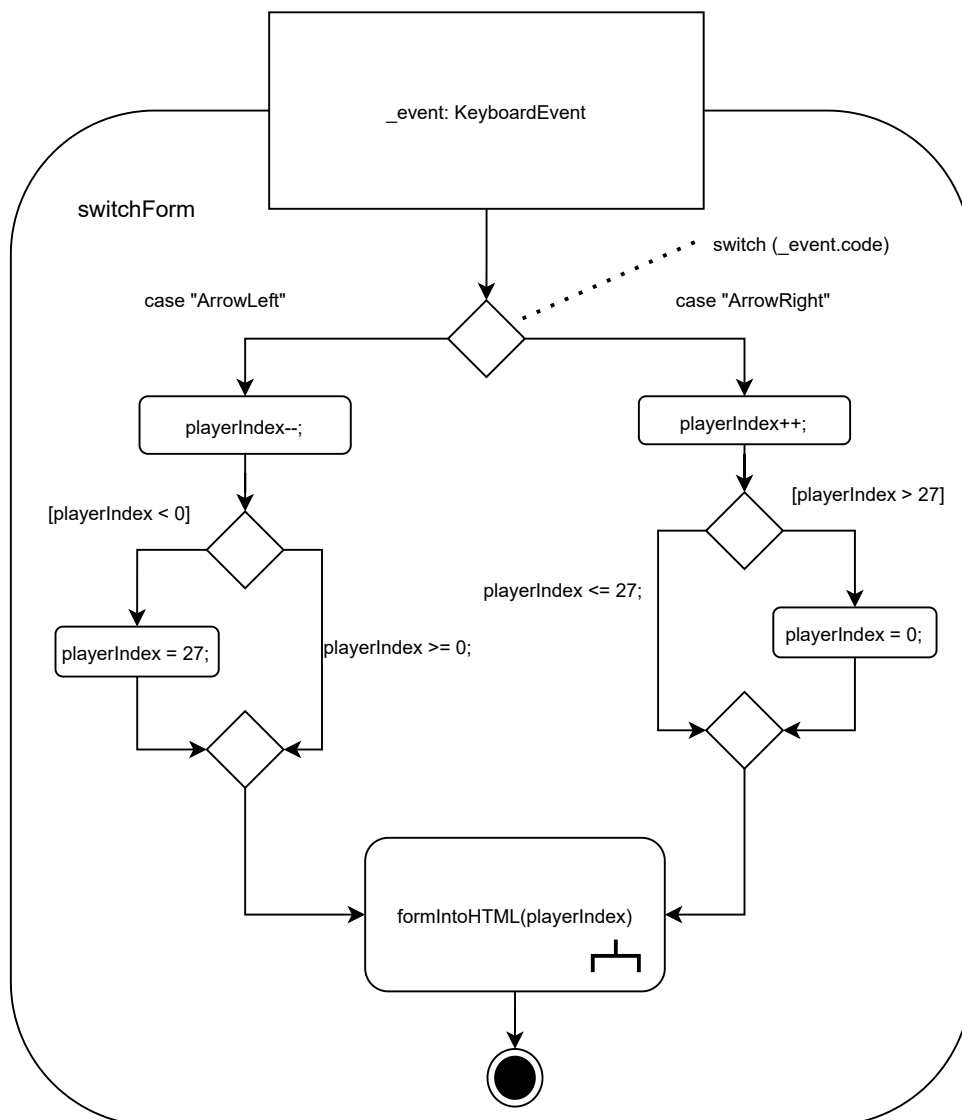
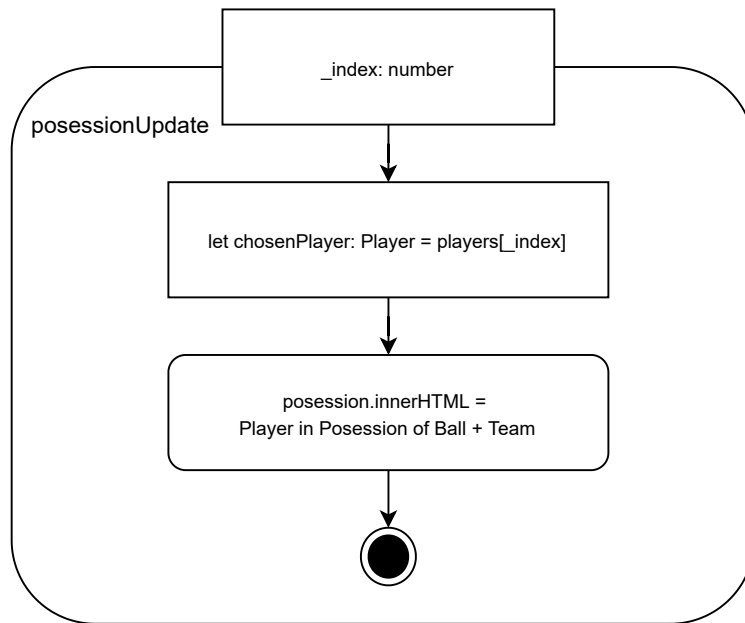
Activitydiagram: Forms

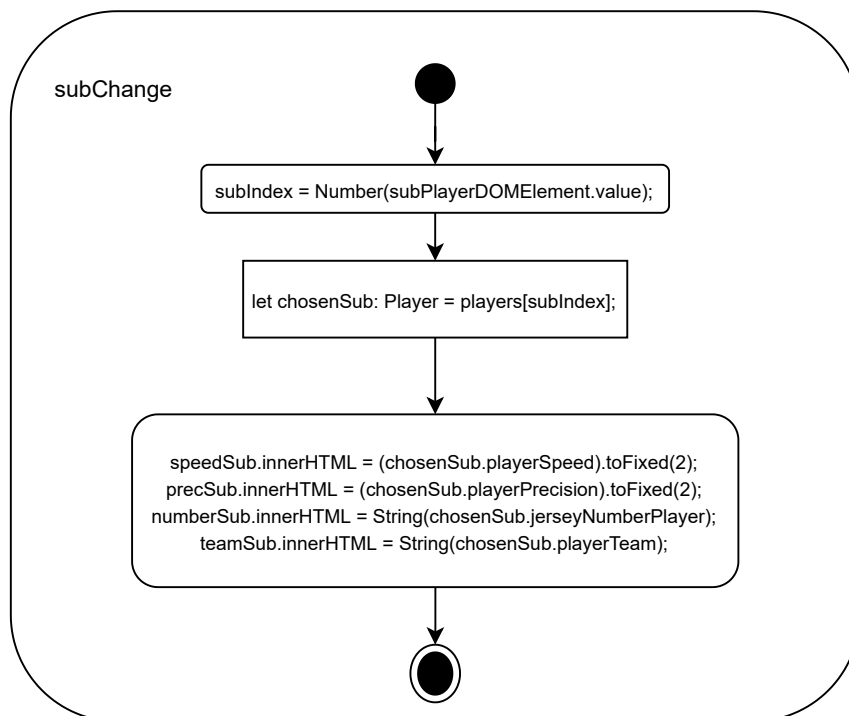
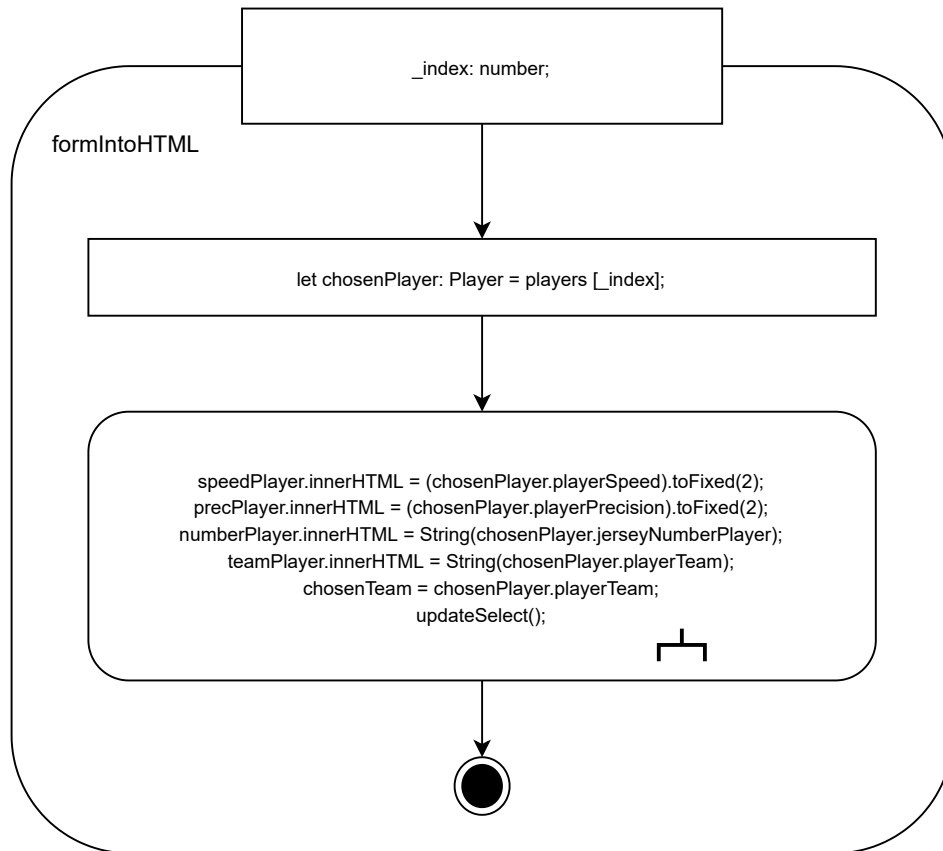


handleChange









Activitydiagram: Vector

