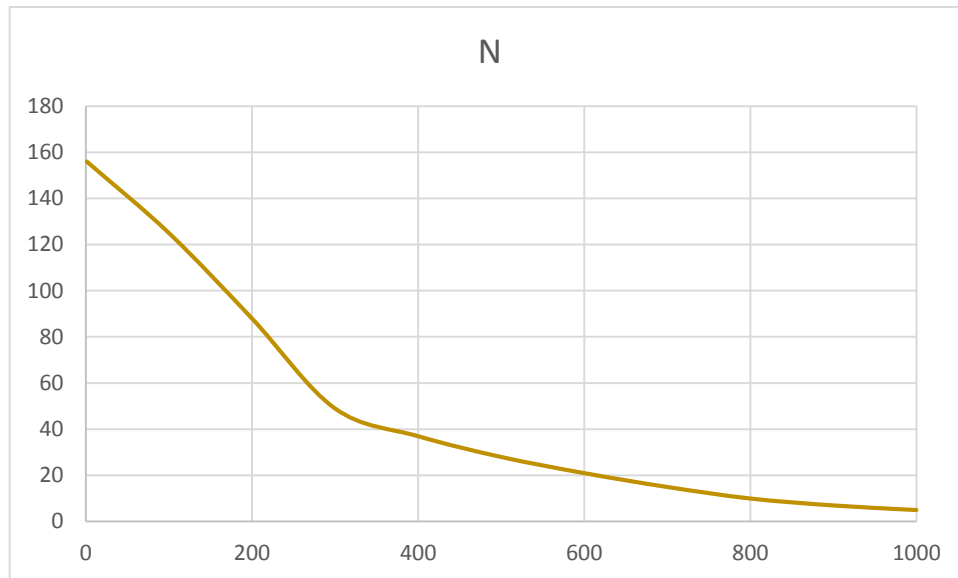


Modern intelligent IT
Lab 6 (14.11.2016)
Akira Imada
Student – Nickita Yurchuck (AI - 10)



Code:

```
function getPlayerMove(data) {
    var currentPlayer = data.yourTeam.players[data.playerIndex];
    var ball = data.ball;

    let ballStop = getBallStats(ball, data.settings)
    , sixthPartOfFieldWidth = data.settings.field.width / 6
    , goalKeeperZoneStartX = 0
    , goalKeeperZoneWidth = sixthPartOfFieldWidth * 2
    , goalKeeperIndex = 1;

    var direction, velocity;

    let boaiINGoalkeeperZone = ballStop.x > goalKeeperZoneStartX &&
        ballStop.x < goalKeeperZoneStartX + goalKeeperZoneWidth;

    if ( data.playerIndex != goalKeeperIndex || boaiINGoalkeeperZone) {
        if (ballStop.x > currentPlayer.x) {
            direction = getDirectionTo(currentPlayer, ballStop);
            velocity = data.settings.player.maxVelocity;
        } else {
            var ballRadius = ball.settings.radius;
            var stopPoint = {
                x: ballStop.x - ballRadius * 2,
                y: ballStop.y + (ballStop.y > currentPlayer.y ? - ballRadius : + ballRadius) * 2
            }
            direction = getDirectionTo(currentPlayer, stopPoint);
            velocity = getDistance(currentPlayer, stopPoint);
        }
    } else {
        let ballRadius = ball.settings.radius;
        let stopPoint = {
            y: ballStop.y + (ballStop.y > currentPlayer.y ? - 1 * ballRadius : ballRadius) * 2,
            x: goalKeeperZoneStartX + goalKeeperZoneWidth / 2,
        };
        direction = getDirectionTo(currentPlayer, stopPoint);
        velocity = getDistance(currentPlayer, stopPoint);
    }
}
```

```

    }

    return {
        direction,
        velocity,
    };
}

function getVelocity (ball, currentPlayer) {
    let percent = data.settings.field.height / Math.abs(ball.y - currentPlayer.y);
    return data.settings.player.maxVelocity * percent;
}

function getDirectionTo(startPoint, endPoint) {
    return Math.atan2(endPoint.y - startPoint.y, endPoint.x - startPoint.x);
}

function getDistance(point1, point2) {
    return Math.hypot(point1.x-point2.x, point1.y - point2.y);
}

function getBallStats(ball, gameSettings) {
    var stopTime = getStopTime(ball);
    var stopDistance = ball.velocity * stopTime
        - ball.settings.moveDeceleration * (stopTime + 1) * stopTime / 2;

    var x = ball.x + stopDistance * Math.cos(ball.direction);
    var y = Math.abs(ball.y + stopDistance * Math.sin(ball.direction));

    // check the reflection from field side
    if (y > gameSettings.field.height) y = 2 * gameSettings.field.height - y;

    return { stopTime, stopDistance, x, y };
}

function getStopTime(ball) {
    return ball.velocity / ball.settings.moveDeceleration;
}

onmessage = (e) => postMessage(getPlayerMove(e.data));

```