CELL-DEVS SIMULATION OF FOOTBALL PLAYER INTERACTION

Youssef Elfaramawy

Department of Systems and Computer Engineering
SYSC5104 - Term Project
Carleton University

ABSTRACT

This paper presents a Cell-DEVS-based simulation of football player interaction, focusing on individual player behavior governed by factors such as mental state and physical fatigue. The model aims to simulate player decisions such as passing, dribbling, or moving into open space, based on interactions within a 2D grid representing a football pitch. Key extensions include obstacle awareness, role-weighted decision logic, and zone-based off-ball movement, allowing for richer and more realistic player dynamics.

Keywords: Football, Simulation, Cell-DEVS

1. INTRODUCTION

This model represents a simplified football match where a player's performance is influenced by physical fatigue and mental state. Players continually adjust their behavior on the pitch – such as moving into open space, passing, or dribbling – based on their current physical and mental state as well as neighborhood conditions.

The football pitch is represented as a discrete 2D grid, reflecting a common football formation. Rather than explicitly modeling complex ball physics, the ball follows player decisions (e.g. dribbling, passing) and movement.

This project improves upon the previous prototype by introducing several enhancements aimed at increasing behavioral realism. Under different configurations of these enhancements, simulation results showed observable differences in player movement and decision-making.

2. BACKGROUND

This model is inspired by real-world football scenarios and Makarenko et al.'s "Towards Cellular Automata Football Models with Mentality Accounting. [1]" Their work simulates a football match using Cellular Automata (CA), considering two teams where each player's movement is determined probabilistically within a von Neumann neighborhood [1]. Player interaction is governed by a set of rules that dictate movement and ball possession [1].

One of the key challenges presented in their report on CA-based football models is the element of anticipation [1]. Anticipation, as defined in their report, is the ability of a player to predict the future state of the game [1]. The authors describe this aspect as mentality accounting [1]. In other words, players, at the atomic level, must decide their best course of action that will result in a higher chance of success (winning) [1]. However, this raises the need for complex mathematical models as well as internal state tracking [1]. Using Cell-DEVS, my model naturally addresses this challenge as each cell manages its own state. As for complex anticipation models, I introduced action costs instead to simulate mental and physical strain that could occur at any point in the game when players perform different actions. Additionally, having players be more aware of their surroundings, the anticipation element can be emulated. This helps maintain the modular nature of Cell-DEVS for discrete-event models while also incorporating the report's mentality accounting aspect.

A key simplification in this model is that it simulates only a single team with persistent possession of the ball. The focus is on understanding and evolving player decisions through controlled internal states and interactions with neighboring cells.

3. MODELS DEFINED

3.1. Conceptual Overview

The simulation environment models a football pitch as a 2D grid. Each cell on the grid may represent a player, the ball, an obstacle, or open space. Given the modularity of the Cell-DEVS formalism, each cell maintains its own state and updates it based on information it gathers from its local neighborhood. The local neighborhood is defined as an extended von Neumann.

Each player's decision-making process is dictated by two internal state variables: mental and fatigue. By meeting specific thresholds, a player may pass, dribble, move, or remain idle. The ball is always possessed by a player and cannot exist in an empty cell.

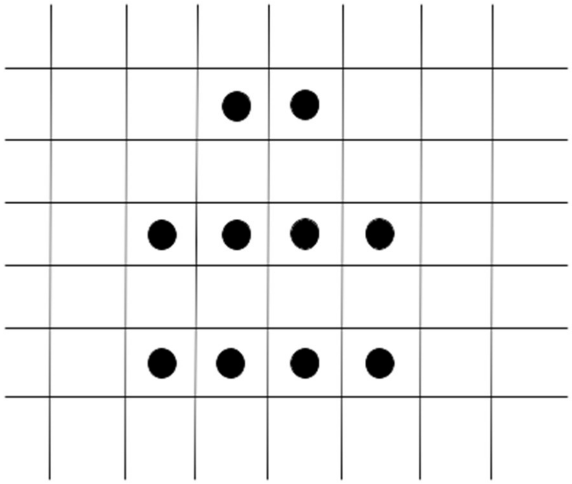

Figure 1: Conceptual Model of Football Pitch Grid

3.2. Player Model

Players can perform the following set of actions:

- Dribble: move with the ball into an adjacent cell (north, south, east, or west)
- Short pass: Pass to an adjacent teammate (east or west)
- Long Pass: Pass to a teammate one (direct) or two (extended) cells away (north or south)
- Move: move into an empty cell without the ball (north, south, east, or west)
- Hold: stay in position with the ball when no viable action is available
- Recover: gradually restore mental and fatigue levels when idle

3.2.1. Action logic

- Fatigue increases during actions and decreases when idle.
- Mental decreases during actions and increases when idle.

3.2.2. In possession of the ball

- Low fatigue, high mental: attempt dribble
- High fatigue, high mental: attempt long pass
- High fatigue, low mental: attempt short pass

3.2.3. Not in possession of the ball

- Move if adjacent teammate (east, west) dribbles and fatigue and mental levels meet thresholds

Ball movement is directly tied to player actions. On dribbles, the ball moves with the player. On passes, the ball transfers to the target cell.

3.3. Enhancements to Player Model

3.3.1. Obstacle-awareness behavior

Obstacles are stationary entities placed on the grid to introduce aspects of mentality accounting [1] and increase realism. By preventing both player and ball movement, we can observe how introducing another team could influence player interactions. Each cell gathers information on obstacles by tracking *has_obstacle* and *near_obstacle* flags within its neighborhood. This obstacle-awareness mechanism introduces tactical decision-making:

- Short passes are avoided if the target cell is near an obstacle (marking)
- Long passes are blocked if an obstacle is in the direction of pass trajectory (interception)

By simulating a second team as stationary obstacles, we can directly observe how the incremental addition of such a factor can pressure players into making better decisions. This abstract approach simplifies multi-team dynamics while still maintaining interaction complexity.

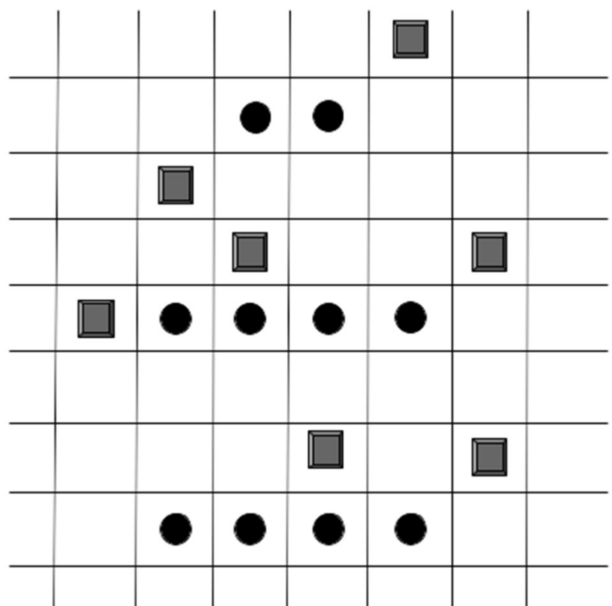

Figure 2: Football Pitch with Obstacles

3.3.2. Zone-based off-ball behavior

Players can be assigned to a zone type: Defense, Midfield, or Attack. This classification introduces unique off-ball behavior:

- Defense: players try to return to their initial row when displaced.
- Midfield: players try to reposition themselves to open spaces not in proximity to obstacles.

- Attack: players try to remain forward and break wide when blocked.

The introduction of zone-based behavior allows us to simulate player-awareness, mimicking aspects of anticipation found in [1]'s report. Without explicitly modeling the entire team strategy, we can observe things at the player level and see how they respond to obstacles.

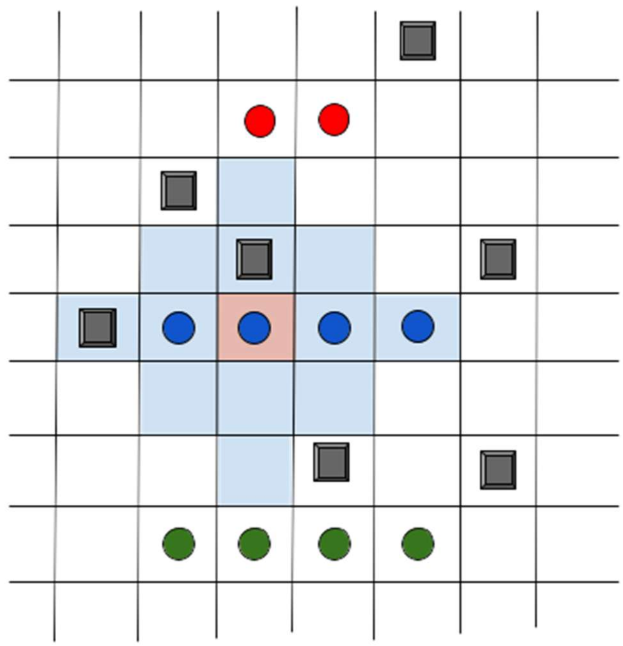

Figure 3: Football Pitch with Obstacles and Zones

3.3.3. Role-based weighted decision logic

Each player can also be assigned a role which modifies their preference to certain actions. In this case, we are looking at passing and dribbling. This is done through weights that scale mental and fatigue thresholds:

Table 1: Role-based Action Weights and Behavior Tendencies

Role	Pass Weight	Dribble Weight	Behavior
Centerback	1.3	0.7	Prioritizes passing
Fullback	0.9	1.1	Prioritizes dribbling slightly
Playmaker	1.4	0.6	Heavy passer
Winger	0.6	1.4	Heavy dribbler
Target Forward	0.8	1.2	Prioritizes dribbling
False Nine	1.2	0.8	Prioritizes passing

These weights scale the player's mental and fatigue levels, allowing thresholds to be easier or harder to meet depending on the specified role. This will allow for varied and realistic player behavior.

3.4. Formal Specification

The Cell-DEVS atomic player specification is defined as follows:

$$\langle X, Y, S, N, d, \tau, \delta_{int}, \delta_{ext}, \lambda, t_a \rangle$$

Where:

- X = Ø, no external inputs
- Y = Ø, no external outputs
- S = State variables:
  {has_player ∈ {true, false}, has_ball ∈ {true, false}, mental ∈ [0, 100], fatigue ∈ [0, 100], action ∈ {None, short_pass, long_pass, dribble, move, hold}, direction ∈ {None, north, south, east, west}, zone_type ∈ {None, defense, midfield, attack}, player_role ∈ {None, centerback, fullback, winger, playmaker, target_forward, false_nine}, initial_row ∈ ℕ, inactive_time ∈ ℕ}
- N = Neighborhood:
  {(-2, 0), (-1, 0), (0, 0), (1, 0), (2, 0), (0, -2), (0, -1), (0, 1), (0, 2), (-1, -1), (-1, 1), (1 ,-1), (1, 1)}
  This represents an extended von Neumann neighborhood.
- d = 1 time unit (Transport delay)
- τ: N → S (Local Computation Rules)

δ_{int}, δ_{ext}, λ, t_a are defined using Cell-DEVS specifications.

Possession-based rules:

Rule 1: Short Pass

Condition:

- Fatigue ∈ (20, 65)
- Mental < 65
- Adjacent teammate (east, west) not near an obstacle
- EAST > WEST

Action:

Perform short pass to that direction

Resulting State:

S = {has_ball = false, action = SHORT_PASS, direction ∈ {EAST, WEST}, fatigue += 3.0, mental -= 1.5}

Rule 2: Long Pass

Condition:

- Fatigue > 20
- Mental ∈ (20, 75]
- North or south teammate (direct or extended) teammate available
- No intercepting obstacle in pass direction
- NORTH > SOUTH

Action:

Perform long pass

Resulting State:

S = {has_ball = false, action = LONG_PASS, direction ∈ {NORTH, SOUTH}, fatigue += 4.0, mental -= 2.5}

Rule 3: Dribbling
Condition:

- Fatigue < 40
- Mental ≥ 60
- At least one cell empty (north, south, east, west)
- NORTH > EAST > SOUTH > WEST

Action:
Perform dribble in first available direction
Resulting State:
S = {has_player = false, has_ball = false, action = DRIBBLE, direction ∈ {NORTH, SOUTH, EAST, WEST}, fatigue += 7.0, mental -= 3.0}

Rule 4: Hold Ball
Condition:

- No other option or action is possible

Action:
Hold the ball
Resulting State:
S = {action = HOLD, fatigue += 2.0, mental -= 1.0}

Off-ball rules:
Rule 5: Move following a Dribbling Teammate
Condition:

- Fatigue < 40
- Mental > 50
- Adjacent teammate (east or west) is dribbling north or south
- Target cell in dribbling direction is empty

Action:
Move into empty cell in dribbling direction
Resulting State:
S = {has_ball = false, action = MOVE, direction ∈ {NORTH, SOUTH}, fatigue += 5.0, mental -= 2.0}

Rule 6: Zone-based Off-ball Behavior
Zone: Defense

- If behind *initial_row*, move north
- If ahead *initial_row*, move south
- If north/south not possible, try moving west or east

Zone: Midfield

- If *near_obstacle*, try repositioning west or east into open space
- If not successful, try moving north or south

Zone: Attack

- If behind *initial_row*, move north
- If blocked by obstacle, drift wide or drop back south

Resulting State:
S = {has_ball = false, action = MOVE, direction ∈ {NORTH, SOUTH, EAST, WEST}, fatigue += 5.0, mental -= 2.0}

Rule 7: Recovery
Conditions:

- No viable actions to move

Action:
Remain idle and slowly recover mental and physical fatigue
Resulting State:
S = {fatigue -= 1.0, mental += 1.0}

3.5. Phase-Based Event Flow in the Cell-DEVS Model

The execution of the Football Player Cell-DEVS follows a structured phase-based event flow, ensuring that player actions/interactions are processed and propagated correctly over time. This approach divides the simulation cycle (single time step) for each cell to operate as follows: data collection, action decision, receiving delayed actions, and cleanup. Each phase is crucial to maintaining a consistent sequential update of states, allowing for a synchronized simulation.

3.5.1. Data collection

At the beginning of each simulation cycle, a cell gathers information from its neighboring cells (*extended von Neumann*) to track different states, allowing it to assess its available options. This includes:

- The presence of teammates in adjacent or extended neighborhood cells.
- The presence of obstacles or neighboring cells that are near obstacles.
- Whether an adjacent or extended neighboring cell has recently performed a *short* or *long pass*.
- Whether an adjacent neighboring cell has recently performed a *dribble* or a *move* action.
- Source player attributes (mental, fatigue, initial_zone, zone_type) of players looking to move or dribble, used for state inheritance when receiving delayed actions.

3.5.2. Action decision

Once the data is collected, a cell evaluates it and its current conditions (mental and fatigue levels) to determine whether an action can be taken. The decision-making process follows a hierarchical structure:

- If a player has possession of the ball, he/she can attempt to pass, dribble, or hold based on predefined thresholds and neighboring conditions.
- If a player does not have possession of the ball, he/she can attempt to move if a neighboring player dribbles or based on off-ball movement logic (e.g. obstacle avoidance, zone behavior). Otherwise, they remain in place and recover.

Constraints are placed on mental and fatigue thresholds to try to prevent overlapping conditions. In the event a player satisfies more than one condition (rule), priority is given to passing over dribbling, followed by holding.

3.5.3. Receiving delayed actions
In this phase, delayed neighborhood inputs (state transitions are broadcasted after d = 1.0) are processed. Actions such as passing, dribbling, and moving from the previous cycle are finalized in this phase. This allows for correct propagation when players pass the ball or move/dribble into a target cell. This phase is responsible for:

- Transferring the ball to a target player, when a *short* or *long pass* occurs.
- Moving a player to a target cell, when a *dribble* or off-ball *move* action occurs. This also includes the target cell inheriting the source cell's mental/fatigue/initial_row/zone_type attributes.

Due to the modular nature of Cell-DEVS, each cell can only modify its state and not that of others. Consider the following example:

- Suppose Cell (2, 2) wants to dribble north to Cell (1, 2), an empty cell
- During the action phase:
  - Cell (2,2) sends an action to *dribble*, causing it to lose player and ball.
    - S = {has_player = true, has_ball = true, action = None, direction = None, mental = 70, fatigue = 30}
    - S = {has_player = false, has_ball = false, action = dribble, direction = north, mental = 67, fatigue = 37}
- During the delayed phase:
  - Cell (1,2) sees action *dribble*, inherits Cell (2,2) mental/fatigue attributes, and gains a player and a ball.
    - S = {has_player = false, has_ball = false, action = None, direction = None, mental = 50, fatigue = 0} (default state)
    - S = {has_player = true, has_ball = true, action = None, direction = None, mental = 67, fatigue = 37}

3.5.4. Cleanup phase
This final phase ensures that cells that have moved/dribbled during the previous step are correctly reset after a certain time. This includes:

- Maintaining source cell output for one time step, ensuring their actions are correctly detected by neighboring cells.
- Clearing *action* and *direction* state attributes to prevent redundant updates in future cycles. This ensures that the next cycle starts with a more accurate presentation of the current pitch (grid) state.

This cleanup step helps prevent constant state updates that carry over during the simulation cycle.

4. SIMULATION RESULTS

4.1. Testing Strategies and Documentation

Given the complexity of player and ball interactions, a hierarchically structured testing approach was adopted. A smaller 3x3 grid framework was used to reduce the number of players and isolate the testing scope to specific actions (*dribble*, *short pass*, *long pass*). This allows for a testing environment that is both controlled and observable when looking at the outputs in the Cell-DEVS viewer.

Each of the following test cases, as mentioned, will focus on a specific gameplay aspect before transitioning to a wider experimental frame.

4.2. Baseline Behavior (No Obstacles, No Zones, No Roles)

4.2.1. Test 1: Short pass (3x3)
This test aimed to validate that players could successfully perform short pass actions to one another. The test was configured as follows:

- Three players (red) positioned in adjacent cells: (2,0), (2.1), (2,2).
- The middle cell (2,1) starts with the ball (blue).
- The middle and right cell meet the criteria for a *short pass* (fatigue and mental meet thresholds).

Results

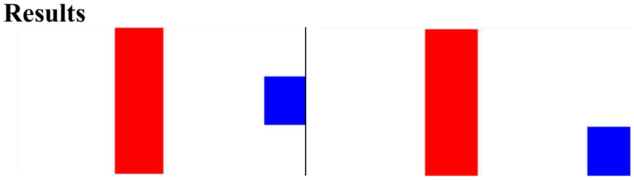

Figure 4: Results of Test #1

The output confirms that:

- Player (2,1) loses possession of the ball after short passing east.
- Player (2,2) gains possession of the ball in the next cycle.

Discussion
By observing the simulation output, we confirm that short passes were correctly executed. This confirms:

- The short pass action rule is functioning as intended.
- The source player loses the ball when passing.

- The target player receives the ball after the expected delay (d = 1).

4.2.2. Test 2: Long pass (3x3)
This test aimed to validate that players could successfully perform long pass actions to one another. The test was configured as follows:

- Two players are positioned in the same column: (2,1), (0,1).
- The bottom cell (2,1) starts with the ball.
- The bottom and top cell meet the criteria for a *long pass*.

Results

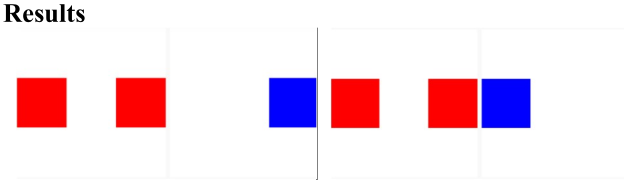

Figure 5: Results of Test #2

The output confirms that:

- Player (2,1) loses possession of the ball after long passing north.
- Player (0,1) gains possession of the ball in the next cycle.

Discussion
By observing the simulation output, we confirm that short passes were correctly executed. This confirms:

- The long pass action rule is functioning as intended.
- The source player loses the ball when passing.
- The target player receives the ball after the expected delay (d = 1).

4.2.3. Test 3: Dribble (3x3)
This test aimed to validate that players could successfully dribble, and if a neighboring cell (east/west) detected that action, it moved accordingly. The test was configured as follows:

- Three players positioned in adjacent cells: (2,0), (2.1), (2,2).
- The middle cell (2,1) starts with the ball.
- The middle cell short passes to right cell (2,2).
- The right cell meets the criteria for *dribble*.
- The middle cell meets the criteria for *move*.

Results

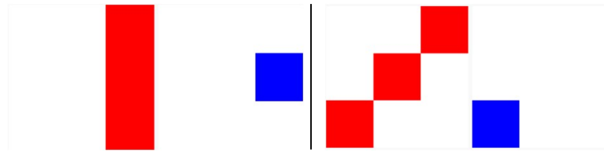

Figure 6: Results of Test #3

The output confirms that:

- Player (2,1) loses possession of the ball after long short passing east.
- Player (2,2) gains possession of the ball in the next cycle.
- Player (2,2) dribbles north to cell (1,2).
- Player (2,1) detects east neighbor's action and moves north to cell (1,1).
- Player (1,2) dribbles north to cell (0,2).
- Player (2,1) remains in place as it doesn't meet move threshold.

Discussion
By observing the simulation output, we confirm that short passes were correctly executed. This confirms:

- The dribble action rule is functioning as intended
- The source cell loses its player and ball when dribbling.
- The target empty cell receives the player and the ball after the expected delay (d = 1).
- The move action rule is functioning as intended.
- The source cell loses its player when moving.
- The target player receives the player after the expected delay (d = 1).

4.3. Obstacle-Awareness Behavior (Obstacles only)

4.3.1. Test 4: Short pass near obstacle (3x3)
This test aimed to validate that players could correctly avoid short passing to teammates near obstacles. The test was configured as follows:

- Two players positioned in adjacent cells: (2,0), (2,1).
- An obstacle (green) positioned adjacent to player (2,1) at (2,2).
- Cell (2,0) starts with the ball and meets criteria for a *short pass*.

Results

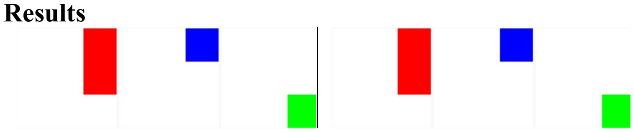

Figure 7: Results of Test #4

The output confirms that:

- Player (2,0) holds possession of the ball and avoids passing to adjacent teammate at (2,1) due to the nearby obstacle.
- No ball transfer occurs across simulation cycles, indicating correct short pass obstacle avoidance logic.

Discussion
By observing the simulation output, we confirm that short passes were correctly avoided when a teammate was near an obstacle. This confirms:

- The obstacle detection logic is functioning as intended
- The source cell retains possession when an open passing is not available.

4.3.2. Test 5: Long pass near obstacle (3x3)
This test aimed to validate that players could correctly avoid long passing to teammates when obstacles are present along the passing trajectory. The test was configured as follows:

- Two players are positioned in the same column: (2,1), (0,1).
- An obstacle is positioned at (2,1) in the same column to intercept the pass.
- The bottom cell (2,1) starts with the ball.
- The bottom and top cell meet the criteria for a *long pass*.

Results

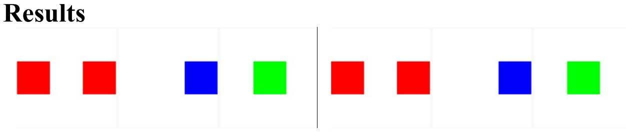

Figure 8: Results of Test #5

The output confirms that:

- Player (2,1) holds possession of the ball and avoids long passing to teammate at (0,1) due to the obstacle in the passing path.
- No ball transfer occurs during simulation cycles, indicating correct long pass obstacle avoidance logic.

Discussion
By observing the simulation output, we confirm that long passes were correctly avoided when an obstacle could intercept it. This confirms:

- The obstacle detection logic is functioning as intended
- The source cell retains possession when an open passing is not available.

4.4. Zone-based Off-ball Behavior (Obstacles + Zones)

4.4.1. Test 6: Defender behavior (3x3)
The test aimed to validate that defenders could successfully track back to their original row while avoiding obstacles. The test was configured as follows:

- One defender is initially placed at (1,1).
- The defender's *initial row* is set at row 2.
- An obstacle is placed at (2,1), directly blocking the defender's southward path.

Results

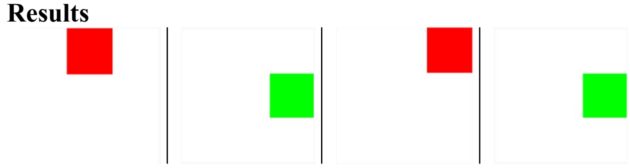

Figure 9: Results of Test #6

The output confirms that:

- The defender immediately attempts to return to their initial row.
- Since the southward path is blocked (obstacle), the defender moves west to (1,0), then south to (2,0), successfully tracking back.

Discussion
By observing the simulation output, we confirm that off-ball movement logic for defenders was correctly executed. This confirms:

- Defenders prioritize returning to their original row.
- Obstacle avoidance logic is functioning as intended when repositioning.
- Movement decisions are reactive and occur immediately based on starting configuration.

4.4.2. Test 7: Midfielder behavior (3x3)
The test aimed to validate that midfielders could successfully reposition into open spaces that are not in proximity to obstacles. The test was configured as follows:

- One midfielder is initially placed at (1,1).
- Two obstacles are placed at (2,1) and (0,0).
- The cell (1,0) has its *near_obstacle* flag set to true

Results

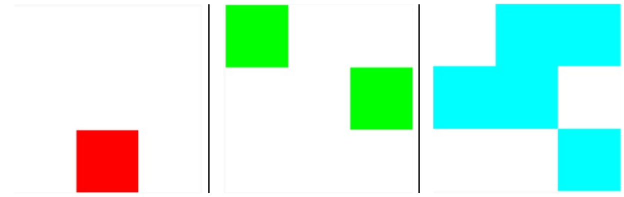

Figure 10: Results of Test #7

The output confirms that:

- The midfielder immediately attempts to reposition himself.
- Since north, south, and west movements are blocked or near obstacles, the best available option is to move east, which player performs.
- The *near_obstacle* flags (cyan) across the grid are set correctly and indicate where open spaces are located.

Discussion

By observing the simulation output, we confirm that off-ball movement logic for midfielders was correctly executed. This confirms:

- Midfielders prioritize being an open passing option.
- Obstacle avoidance logic is functioning as intended when repositioning.
- Movement decisions are reactive and occur immediately based on starting configuration.

4.4.3. Test 8: Attacker behavior (3x3)

The test aimed to validate that attackers could successfully stay forward or drift wide when blocked by obstacles. The test was configured as follows:

- One attacker is initially placed at (2,1).
- The attacker's *initial row* is set at row 0.
- An obstacle is placed at (0, 1)

Results

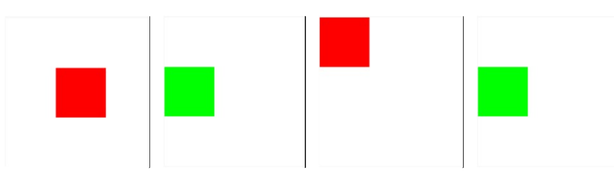

Figure 11: Results of Test #8

The output confirms that:

- The attacker immediately attempts to reposition himself.
- Since attackers try to stay forward, he moves north to (1,1).
- Due to the obstacle at (0,1), the attacker drifts wide to (1,2), and then moves north to (0, 2) reaching his *initial row* of 0.

Discussion

By observing the simulation output, we confirm that off-ball movement logic for attackers was correctly executed. This confirms:

- Attackers prioritize staying forward toward their initial row.
- Obstacle avoidance logic is functioning as intended when repositioning.
- Movement decisions are reactive and occur immediately based on starting configuration.

4.5. Experimentation

For experimentation, the focus was on scaling up the simulation to a 10x10 grid with 10 players under different configurations. The goal was to produce an environment to examine whether players could dynamically adjust their actions based on mental/fatigue states, local neighborhood conditions, and the presence of obstacles, zones, and role-based behaviors.

The objective was to observe a higher degree of interaction between players and actions in a larger-scale simulation. As long as players responded dynamically, executed valid actions, and respected game constraints, it indicated that the system functioned as intended.

Table 2: Simulation Configurations

Configuration	Obstacles	Zones	Roles
1	No	No	No
2	Yes	No	No
3	Yes	Yes	No
4	Yes	Yes	Yes

Each simulation was recorded using the [Cell-DEVS Viewer] for visual demonstration of the state changes between cells, allowing us to see in real-time player decisions and ball interactions. To better understand the dynamics across configuration, the number of short passes, long passes, dribbles, and off-ball moves were recorded, as shown in Table 3.

Table 3: Action Counts by Configuration

Configuration	Short Passes	Long Passes	Dribbles	Moves
1	2	2	16	10
2	2	2	10	4
3	2	2	10	18
4	2	4	16	18

4.5.1. Discussion

The recorded simulation provides a clear way to analyze the interactions across different 10x10 grid configurations. Observing the video, we can see that players dynamically adjust their behaviors, execute specific actions (short pass, long pass, dribble, move, hold), and respect game constraints (e.g. only one ball can be present on the pitch at any given moment).

A link to the recording for each configuration can be found here (inside the videos folder; 10x10_full): https://github.com/yelfaram/Cadmium-Cell-DEVS-Football-Player-Interaction

Key observations include:

- Short passes remained consistent across all configurations, indicating little impact from the introduction of obstacles, zones, or roles.
- Introducing obstacles (config 2) significantly reduced the number of dribbles and off-ball moves compared to config 1. This confirms that players were more restricted.
- Introducing zones (config 3) greatly increased off-ball movements as players repositioned themselves based on their roles and obstacle proximity.
- Introducing roles (config 4) caused an increase in dribbles and long passes, indicating an influence over player behavior.
- Across all simulations, players successfully performed actions based on their surroundings. Ball possession transitioned correctly between players for passes and between cells for dribbles.

It is important to note that, across all configurations, final simulation outputs differed. This demonstrates that player behavior dynamically adjusted to its field conditions.

5. CONCLUSION

The football player interaction model, developed using Cell-DEVS and Cadmium, successfully demonstrated short passing, long passing, dribbling, and off-ball movement behaviors within a dynamic grid environment. Scaling from a 3x3 to a 10x10 grid revealed increasing complexity in player interactions, validating the model's ability to manage modular state updates and delayed event propagation.

Enhancements such as stationary obstacles, zone-based off-ball behavior, and role-specific decision-making demonstrated meaningful changes in player movement patterns and improved realism, although some limitations remain. Actions are still primarily threshold-driven, and external game factors (e.g., scoreline) are not yet incorporated. A possible future extension could involve adopting the "18-zone" pitch model to introduce greater tactical variation [2], alongside asymmetric or hexagonal neighborhoods to allow more natural movement patterns.

Overall, the model provides a solid foundation for simulating player decisions in a discrete-event framework. Further refinement could extend its applicability to more complex tactical environments such as sports.

6. ACKNOWLEDGEMENTS

I would like to express my thanks to Professor G. Wainer.

7. REFERENCES

[1] A. Makarenko, D. Krushinski, A. Musienko, and B. Goldengorin, "Towards Cellular Automata Football Models with Mentality Accounting," *Lecture Notes in Computer Science*, pp. 149–152, 2010, doi: https://doi.org/10.1007/978-3-642-15979-4_16.

[2] "Football/Soccer Session (Moderate): Positional Awareness," *Sport Session Planner*. https://www.sportsessionplanner.com/s/Ho5lb/Positional-Awareness.html?interface=en