Tic Tac Toe

DetectWins

When a player places three of his or her marks in a horizontal, vertical, or diagonal line, the player wins;

EnforceTurns

To play, one player marks a square in a 3 by 3 grid with X, then the other player marks a square with O, then it is X's turn again, and so on;

SquareTaken

Once a square is marked, it cannot be marked again;

DefaultOMoves

When other tactics are not applicable, player O should prefer the center square, then the corners, and mark an edge square only when there is no other choice;

StartOAtCenter

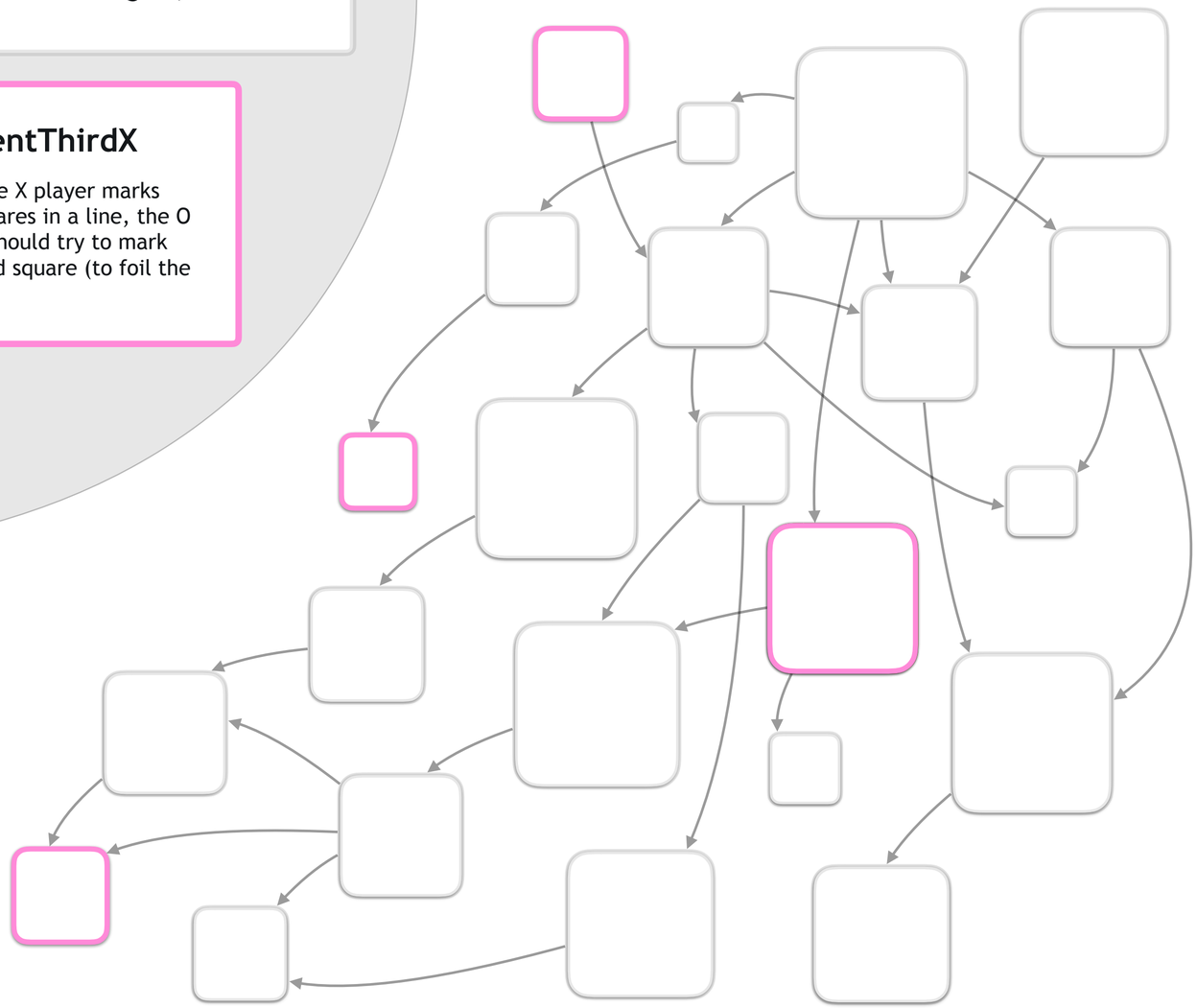
O should start playing at the center;

PreventThirdX

After the X player marks two squares in a line, the O player should try to mark the third square (to foil the attack);

"requirements world"

"implementation world"



Abstraction does not imply Hierarchy

"abstracting something away"—removing details to better see the large picture.

Tic Tac Toe

DetectWins

When a player places three of his or her marks in a horizontal, vertical, or diagonal line, the player wins;

EnforceTurns

To play, one player marks a square in a 3 by 3 grid with X, then the other player marks a square with O, then it is X's turn again, and so on;

SquareTaken

Once a square is marked, it cannot be marked again;

DefaultOMoves

When other tactics are not applicable, player O should prefer the center square, then the corners, and mark an edge square only when there is no other choice;

StartOAtCenter

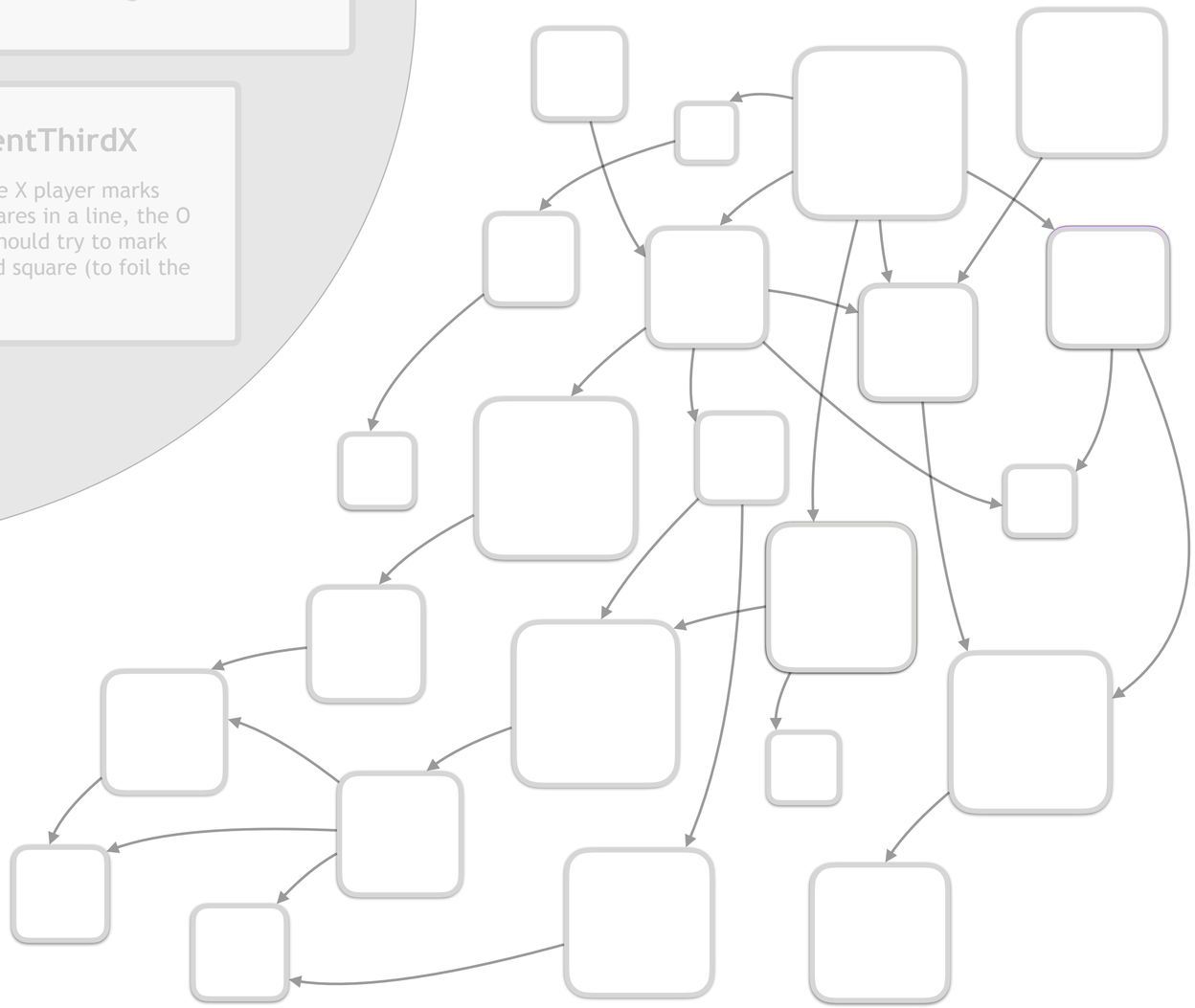
O should start playing at the center;

PreventThirdX

After the X player marks two squares in a line, the O player should try to mark the third square (to foil the attack);

"requirements world"

"implementation world"



Append only

New requirements are 'piled-atop' it, with no component specific interface, connectivity, or ordering requirements; refining previously stated sentences. Without even seeing old requirements.