Simple Soccer

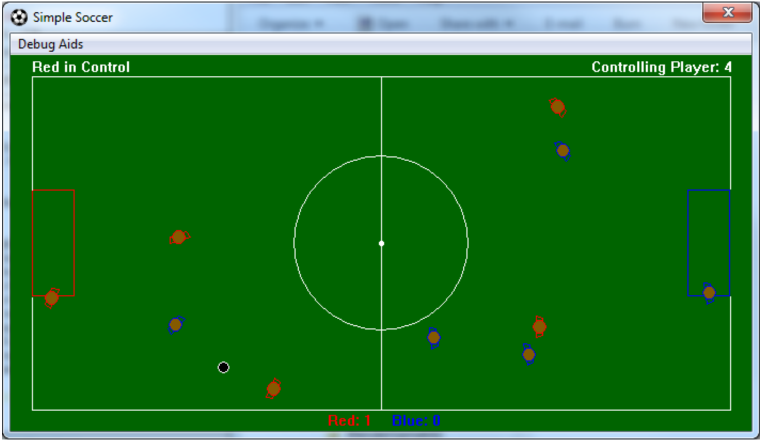
A game demo from *Programming Game AI by Example*, Mat Buckland

CSCI 321

WWU

November 27, 2017

Simple Soccer

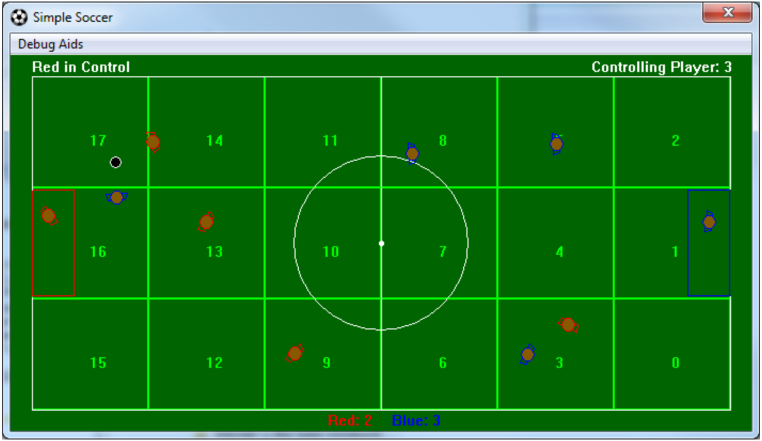


Simple Soccer

- Soccer pitch
- Goals
- Ball
- Teams
- Field players
- Goalies

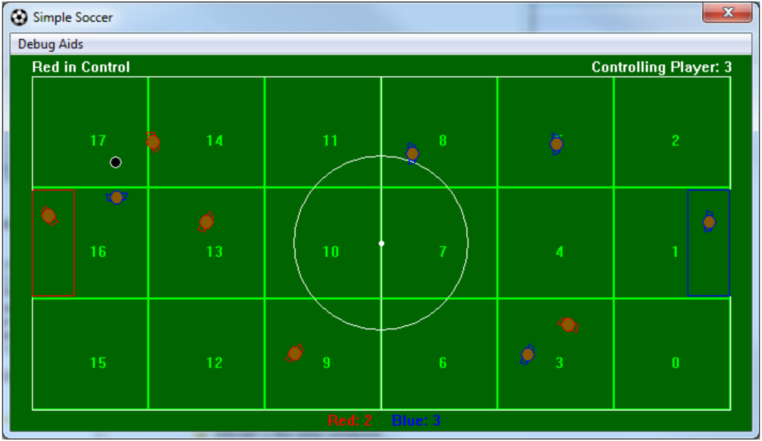
Pitch

- Holds center spot for game restart.
- Booleans: `GameOn`, `GoalieHasBall`



Pitch divided into regions

- Players head for home region if nothing else to do.
- Home region may change during play.



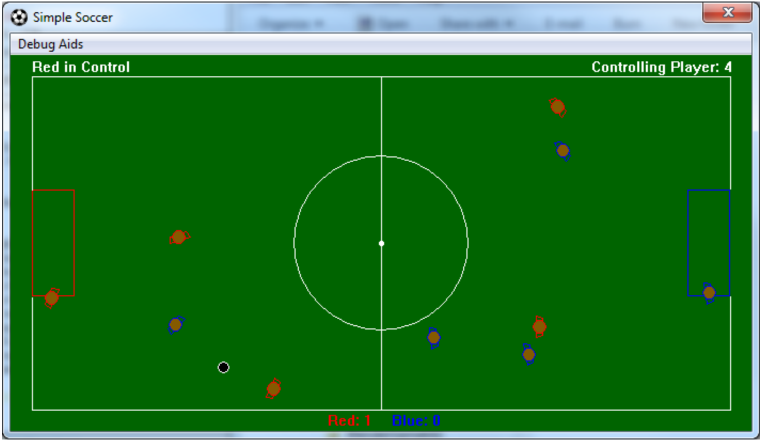
Red in Control

Controlling Player: 3

Red: 2 Blue: 3

Goals

- Checks for score (collision with ball).
- Keeps score, resets game.



Ball

- Checks for collisions with boundary
- Does not collide with players (need to dribble)
- Accelerated by kick, decelerated by friction.
- Players need to make predictions:
    - `FuturePosition`:

      $$\Delta x = u\Delta t + \frac{1}{2}a\Delta t^2$$

      Δx is distance traveled, Δt is time taken.
    - `TimeToCoverDistance`:

      $$\begin{aligned} \Delta t &= \frac{v-u}{a} \\ v &= \sqrt{u^2 + 2a\Delta x} \end{aligned}$$

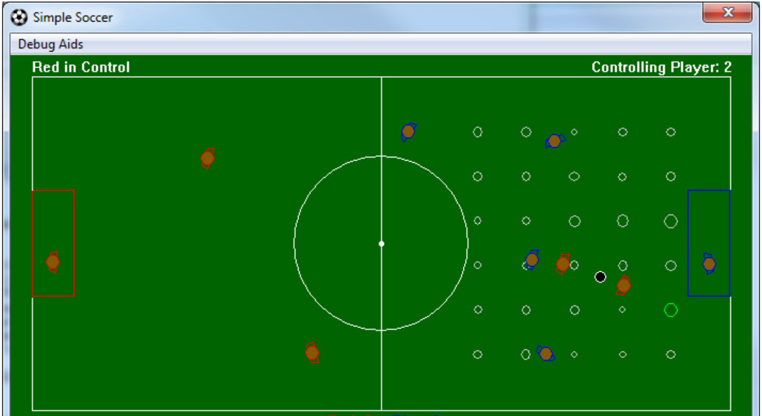
      u is the starting velocity, v is the ending velocity
- Velocities are not accumulative, but that works for this game.

Soccer Team

- A *tiered* or *hierarchical* AI, some decisions handled at the team level, some at individual level.
    - Used in RTS: unit, troop, command levels
- Players and teams have the ability to send *messages*.
    - Messages handled in each player's global state
- Important players (can be NULL):
    - Receiving player
        - After a pass has been made
    - Closest player to the ball
    - Controlling player
        - Includes receiving player
    - Supporting player

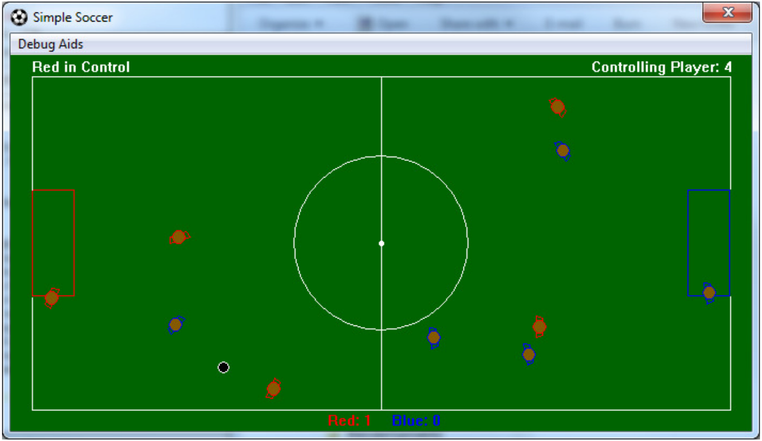
Support Spots

- Can a pass be made to there?
- Can a goal be scored from there?
- How far is it from supporting player's position?
- Is it too close or far from the controlling player?
- Need not be calculated every step.



Soccer Team States

- Change home regions, send messages:
  - `PrepareForKickOff`
  - `Defending`
  - `Attacking`



Field Player Motion

- Uses speed and velocity-aligned heading unit vector
- Heads track ball for illusion of intelligence
    - Nobody receives ball from behind
- Uses steering behaviors
    - arrive
    - seek
    - pursuit

Field Player States

- GlobalPlayerState (sends and receives messages)
- Wait
- ReceiveBall
- KickBall
- Dribble
- ChaseBall
- ReturnToHomeRegion
- SupportAttacker

Messages for GlobalPlayerState

- SupportAttacker
- GoHome
- ReceiveBall
- PassToMe
- Wait

ChaseBall

- **seek** the ball
- Changes to KickBall if ball comes in range
- Changes to ReturnToHomeRegion if not closest player to ball

Wait

- If upfield of Controller, sends message PassToMe
- If closest to ball, and no receiver, change to ChaseBall

ReceiveBall

- Entered on message ReceiveBall
- Only one player in ReceiveBall state
- Uses either **arrive** or **pursuit** based on
    - Randomness
    - Threatening opponent
    - Receiver close to opponent's goal
- Change to ChaseBall if
    - Ball comes close
    - Team loses control

KickBall

- Change to ChaseBall when:
    - Too soon after kick
    - Ball behind player
    - Player waiting to receive ball
    - Goalie has ball
- If player has a shot, or occasionally randomly (potshots):
    - Shoot ball with random noise added
    - Strength of kick determined by angle from player's heading
    - Change to Wait
    - Find support
- If pass is possible and threatened:
    - Pass ball with random noise added
    - Send message to receiver
    - Change to Wait
    - Find support
- Otherwise, Find Support and change to Dribble

Dribble

- Set controlling player to me
- If ball is downfield, make small angled kicks
- Otherwise kick upfield
- Enter ChaseBall

SupportAttacker

- **arrive** at best support spot
- If team loses the ball, change to ReturnToHomeRegion
- If can shoot, send RequestPass to controlling player
- If at best support spot:
    - Steering off
    - Track ball
    - If not threatened, send RequestPass

Goalkeepers

- Always faces ball
- Moves laterally in front of the goal and along heading axis
- States:
    - GlobalKeeperState
    - TendGoal
    - ReturnHome
    - PutBallBackInPlay
    - InterceptBall

GlobalKeeperState

- Receive message GoHome
- Receive message ReceiveBall

TendGoal

- **interpose** between ball and a corresponding position in the goal
- If ball is very close:
    - Trap ball
    - Set GoalKeeperHasBall to true
    - Change state to PutBallBackInPlay
- If ball is somewhat close:
    - Change state to InterceptBall
- If too far from goal:
    - Change state to ReturnHome

ReturnHome

- **arrive** at home region
- If in home region or team loses the ball:
    - Change to TendGoal

PutBallBackInPlay

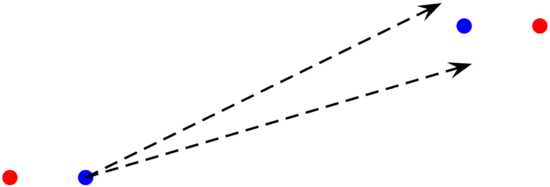
- Set the controlling player to me
- Message all field players on both teams to return home
- If pass is available:
    - Kick the ball
    - Set GoalKeeperHasBall to false
    - Send message ReceiveBall
    - Change to TendGoal

InterceptBall

- **pursuit** of ball
- If too far from goal and goalie is NOT closest to the ball:
    - Change to ReturnHome
- If ball is very close:
    - Trap the ball
    - Set GoalKeeperHasBall to true
    - Change to PutBallBackInPlay

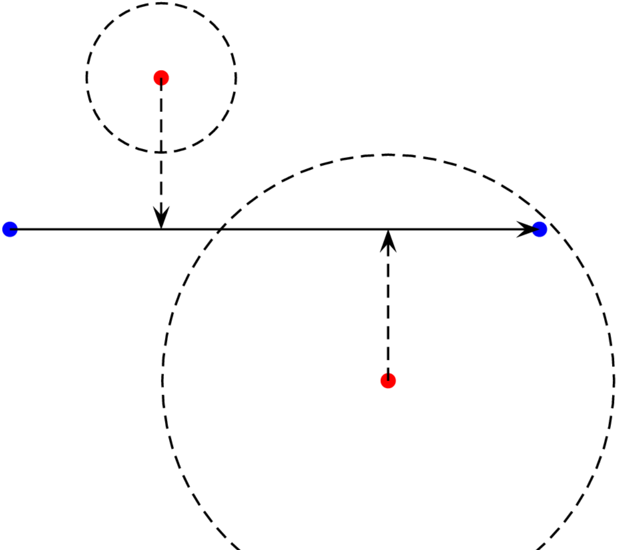
isPassSafeFromOpponent

- Yes if Opponent is behind player or farther back from target than receiver



isPassSafeFromOpponent

- Can opponent get to closest intercept point in time?



CanShoot

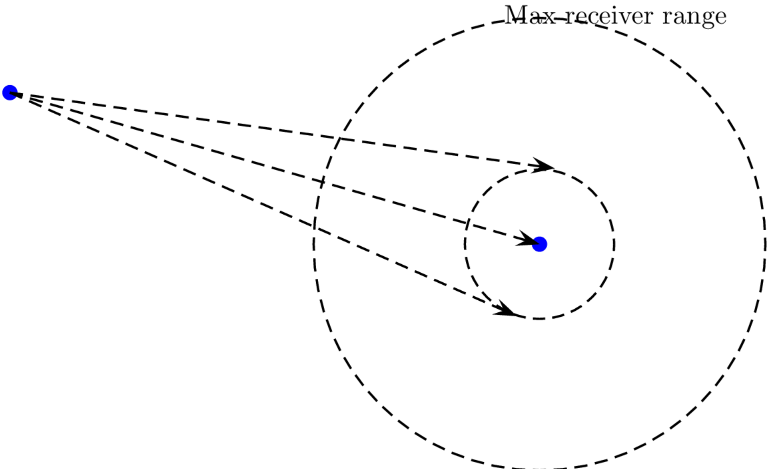
- Randomly pick several points along the goal line
- Check each one to see if opponent can intercept

FindPass

- Iterate through teammates within passing distance
- Call GetBestPassToReceiver
- Keep the one closest to goal

GetBestPassToReceiver

- Calculate how long it takes ball to get to receiver
- Find a circle one-third the size of receiver's range
- Keep safe pass closest to goal



Making Estimates and Assumptions

- Artificial stupidity
- It's more realistic
- Two ways:
  - Make it perfect and then dumb it down
  - Design it using simplifying assumptions
- Examples:
  - Adding random noise to a perfect kick
  - Using one-third size circles to estimate a receiver's range