

Soccer-Fun

functioneel programmeren met voetbal

<http://www.cs.ru.nl/P.Achten/SoccerFun/SoccerFun.html>

Wat is SoccerFun?

- Programmeer het brein van een voetballer
- Maak een team van elf voetballers
- Voetbal tegen andere teams en kijk of je het beste team hebt
- Geïnspireerd op een uitspraak van Johan Cruijff

Voetbalbrein

- Johan Cruijff (De Tijd, 7 mei 1982):
“Als ik een bal aan de voet heb die ik wil afspelen, dan moet ik rekening houden met mijn bewaker, de wind, het gras, de snelheid waarmee de spelers lopen. Wij berekenen de kracht waarmee je moet schoppen en de richting waarin in ééntiende seconde. De computer doet daar twee minuten over!”

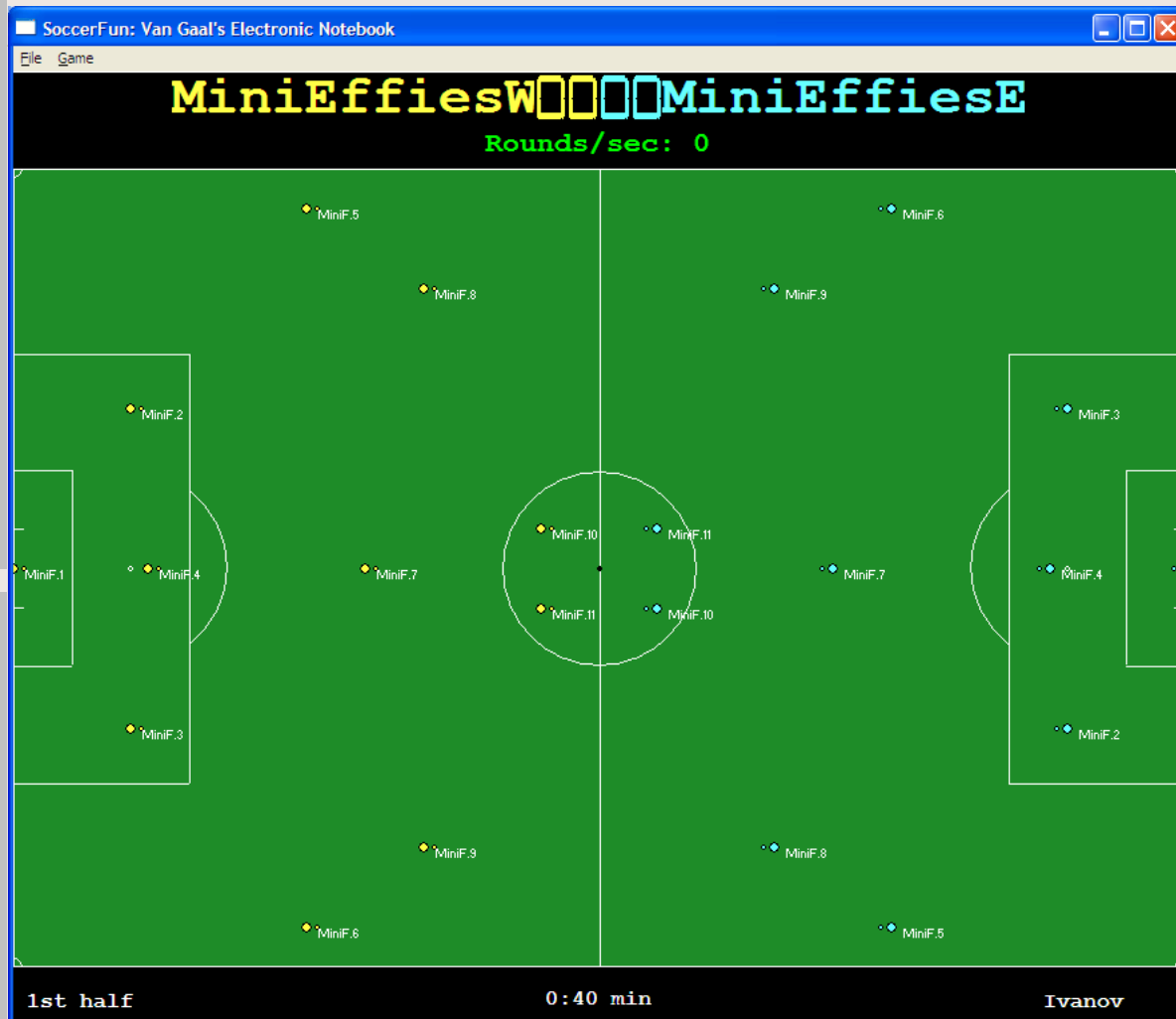


Voetbalbrein

- Johan Cruijff (De Tijd, 7 mei 1982):
*“Als ik een bal aan de voet heb die ik wil afspelen, dan moet ik rekening houden met mijn **bewaker**, de **wind**, het **gras**, de **snelheid waarmee de spelers lopen**. Wij berekenen de **kracht** waarmee je moet schoppen en de **richting** waarin in ééntiende seconde. De computer doet daar twee minuten over!”*
- Deze berekening is een functie:
brein :: Bewaker Wind Gras Spelers → (Kracht, Richting)



SoccerFun



Een voetbal(ler)model



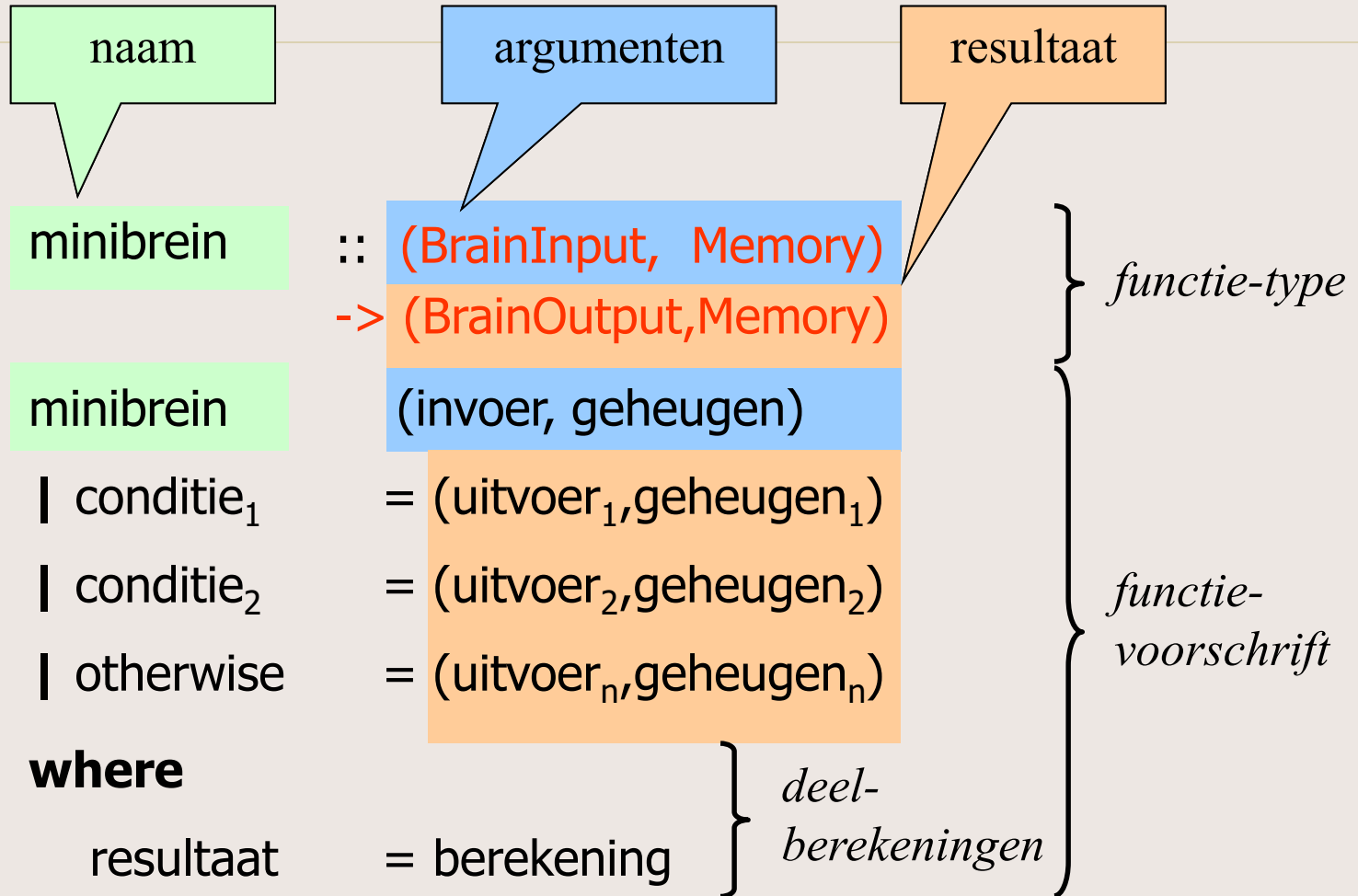
Voetballer

:: Footballer = E.m:

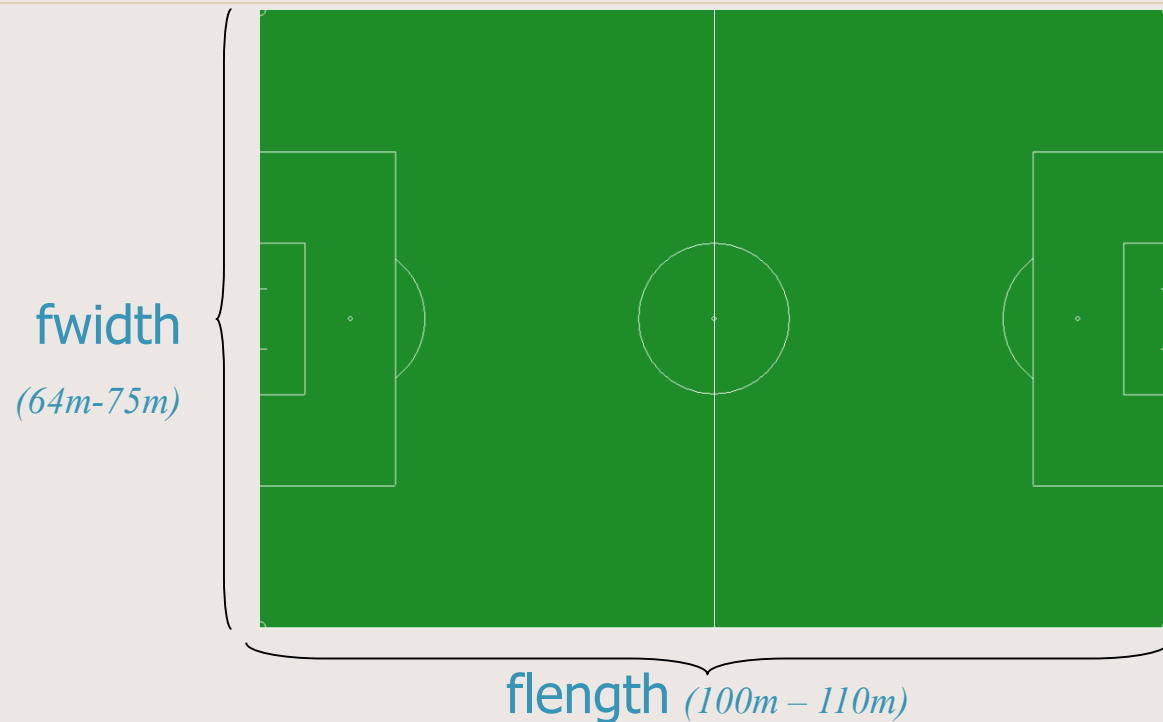
```
{ playerID      :: FootballerID, nose :: Angle
, pos          :: Position,   speed  :: Speed
, name        :: String,     length :: Length
, stamina     :: Stamina,    health  :: Health
, skills      :: MajorSkills
, effect      :: Maybe FootballerEffect
, brain       :: Brain (FootballerAI m) m
}
```



“Ons” brein is ook een functie



Dimensies – voetbalveld

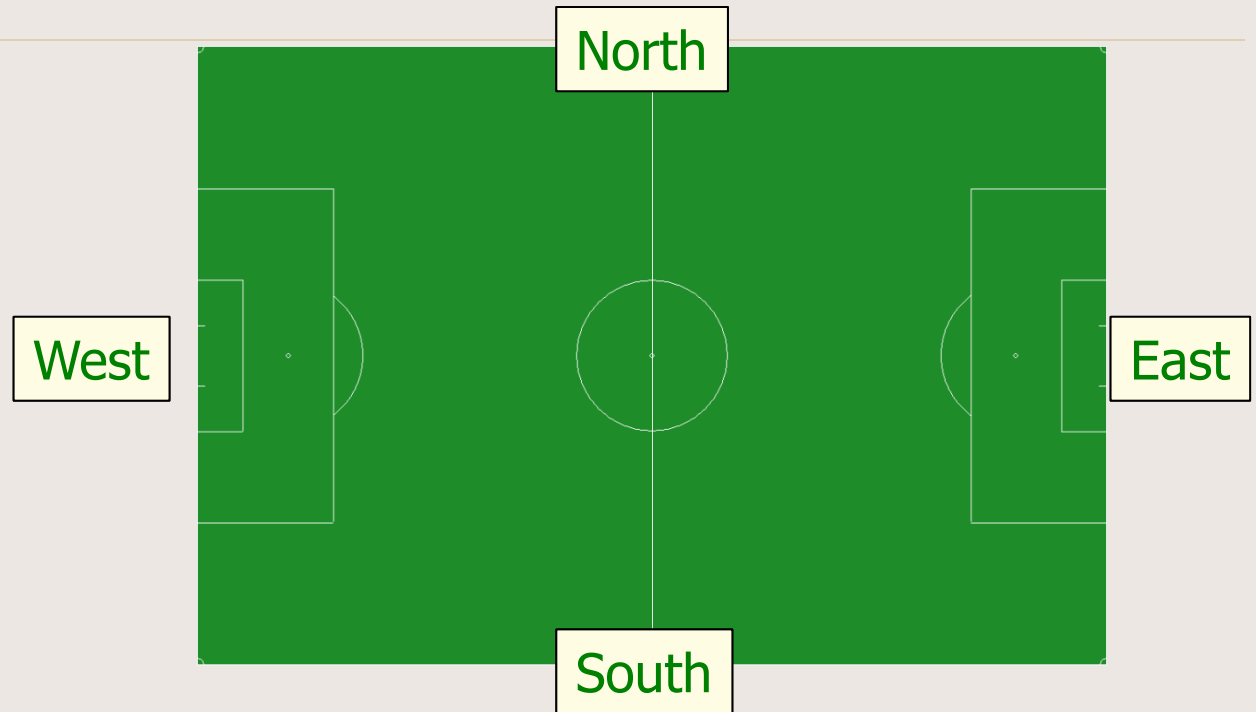


```
:: FootballField = { fwidth :: FieldWidth, flength :: FieldLength }  
:: FieldWidth   ::= Metre  
:: FieldLength  ::= Metre  
m :: Real -> Metre
```

Eenheden

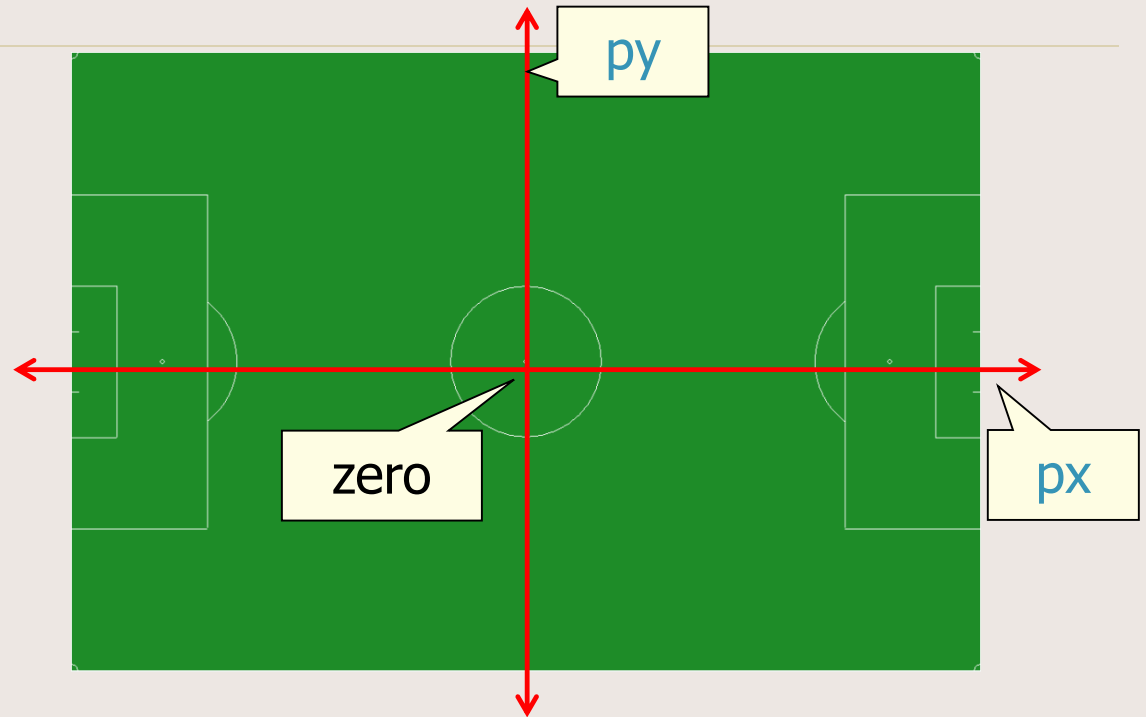
Eenheid	Nederlands	SoccerFun		Type
lengte	0 m 50,4 m	(m 0.0) (m 50.4)	of: zero	Metre
tijd	0 min 45 min	(minutes 0.0) (minutes 45.0)	of: zero	Minutes
hoek	0° 90° 180° 270°	(degree 0) (degree 90) (degree 180) (degree 270)	of: zero of: (rad (0.5*pi)) of: (rad pi) of: (rad (1.5*pi))	Angle
snelheid	0 m/s 10 m/s	(ms 0.0) (ms 10.0)	of: zero	Velocity

Dimensies – voetbalveld



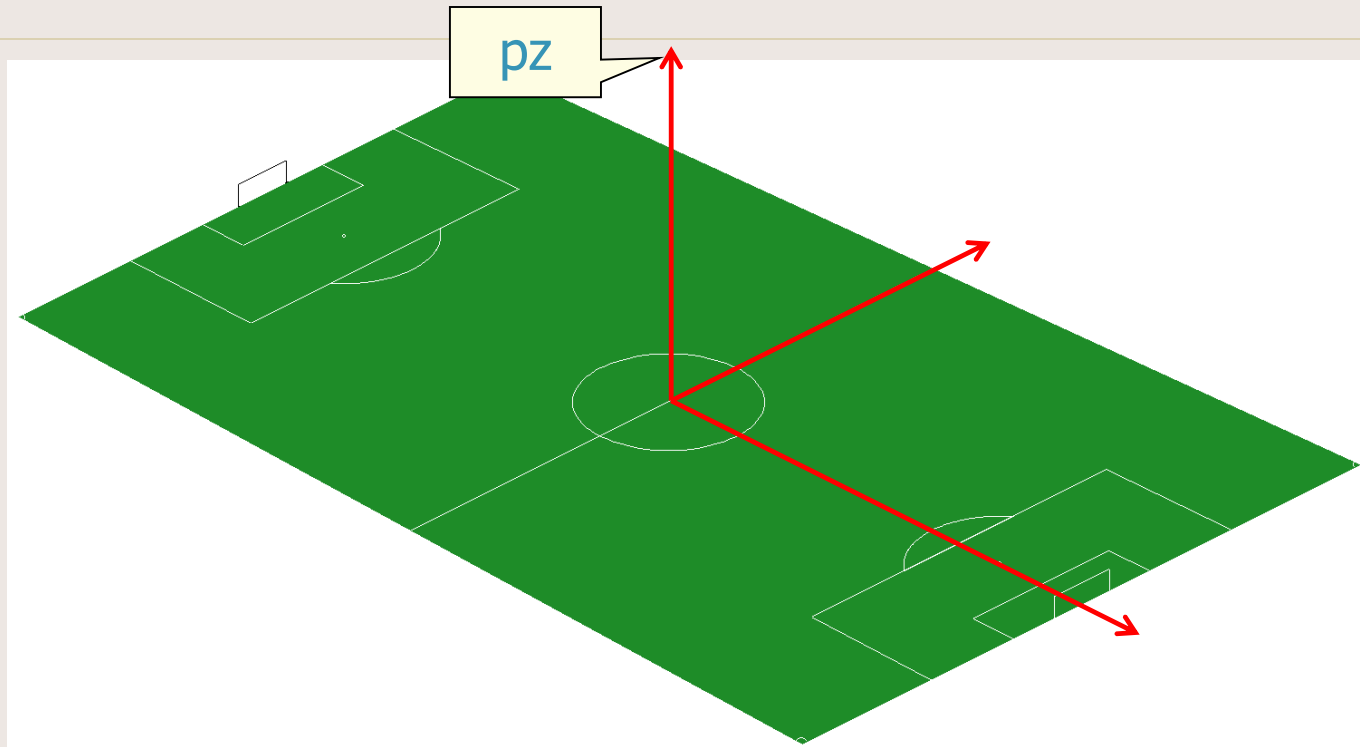
:: Edge = North | South
:: Home = West | East

Dimensies – veldcoördinaten



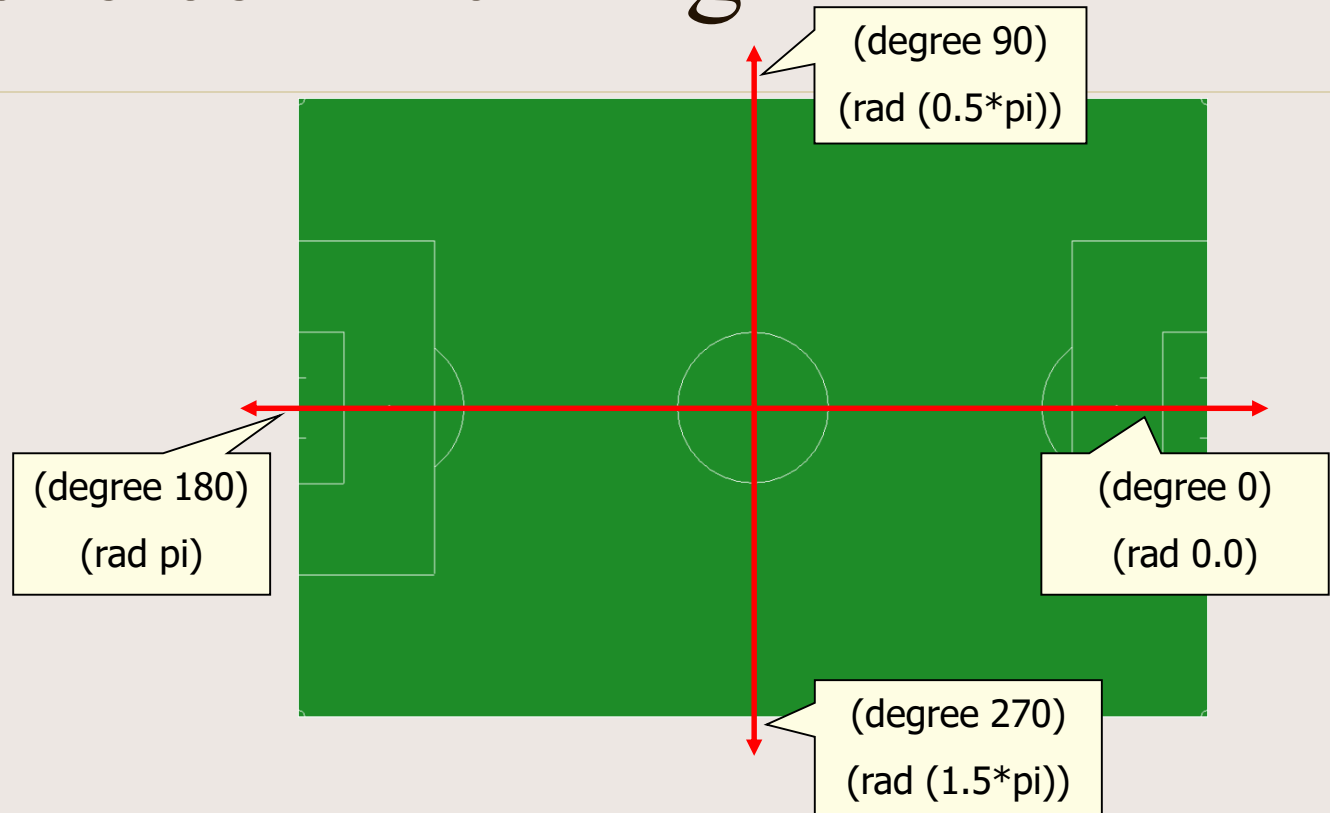
:: Position = { px :: Metre, py :: Metre }
:: Footballer = E.m: { ... pos :: Position ... }

Dimensies – hoogte



:: Position3D	= {	pxy	:: Position, pz :: Metre}
:: Football	= {	ballPos	:: Position3D
	,	ballSpeed	:: Speed3D
	}		

Dimensions – richting



:: Angle

degree :: Int -> Angle

rad :: Real -> Angle

pi ::= 3.1415926535897932384

Dimensies – bewegingssnelheid

... me.nose = v ...

{vxy={direction= β , velocity = ms 5.5}
, vz = ms 1.8 }

{direction= α , velocity = ms 6.3}

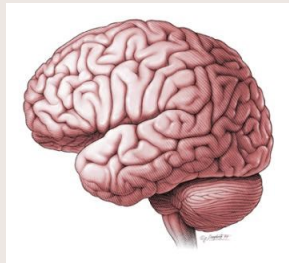
:: Speed = {direction :: Angle, velocity :: Velocity }

:: Speed3D = {vxy :: Speed, vz :: Velocity }

:: Velocity

ms :: Real -> Velocity

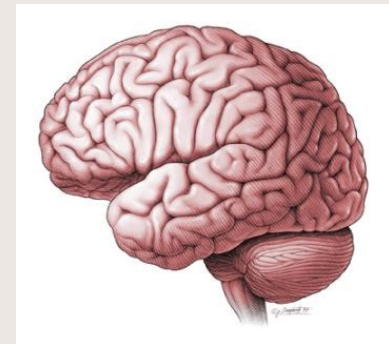
Breinfuncties maken



+



=



“Ons” brein is anders

```
:: Footballer      = E.m:
                        { ...
                        , brain :: Brain (FootballerAI m) m
                        }
```

```
:: Brain ai m      = { memory :: m, ai :: ai }
```

```
:: FootballerAI m ::=      (BrainInput, m)
                        -> (BrainOutput,m)
```

```
mijnBrein :: (BrainInput, Memory)
            -> (BrainOutput, Memory)
```

Wat weet een voetballer?

:: BrainInput

```
= { referee    :: [RefereeAction]
    , football :: FootballState
    , others   :: [Footballer]
    , me       :: Footballer
  }
```

:: FootballState

```
= Free Football | GainedBy FootballerID
```

Wat kan een voetballer doen?

:: BrainOutput ::= FootballerAction

:: FootballerAction

= Move Speed Angle | Feint FeintDirection

| GainBall | CatchBall

| KickBall Speed3D | HeadBall Speed3D

| Tackle FootballerID Velocity

Voorbeeld 1: sta stil

```
halt    :: (BrainInput, m)  
        -> (BrainOutput, m)
```

```
halt    (input, m)  
        = (Move zero zero, m)
```

- module FootballerFunctions bevat een aantal miniscule breinfuncties

Voorbeeld 2: loop naar punt

fix *point diff* (input={me}, m)

| *d < diff* = **halt** (input, m)

| **otherwise** = (move, m)

where

d = dist *me point*

a = bearing zero *me point*

r = bearing *me.nose me point*

move = **Move** {*direction*=a, *velocity*=ms (toReal d)} r

Voorbeeld 2: loop naar punt

```
afterfix doe_dan point diff (input={me}, m)
| d < diff      = doe_dan (input, m)
| otherwise     = (move, m)
where
  d      = dist me point
  a      = bearing zero me point
  r      = bearing me.nose me point
  move   = Move {direction=a,velocity=ms (toReal d)} r
```

generaliseer **fix** met
vervolgactie

Voorbeeld 3: speel bal naar punt

kick *point* (input={me}, m)

| d <= maxKickReach me = (kick, m)

| **otherwise** = **halt** (input, m)

where

ball = getBall input

d = dist me ball

v = dist me *point*

a = bearing zero me *point*

kick = **KickBall** {vxy = {direction=a, velocity=ms (toReal v)}
, vz = ms 1.0 }

Voorbeeld 4: volg bal

```
track_ball diff (input,m)  
  = fix (getBall input).ballPos.pxy diff (input,m)
```

... en de mini-effies?

```
miniFF diff (input,m)  
  = afterfix trap_bal_in_doel  
    (getBall input).ballPos.pxy diff (input,m)
```

Scheidsrechter



- Scheidsrechter observeert spelverloop
- Beslissingen van de scheidsrechter moeten worden opgevolgd
 - spelhervatting
 - speelhelft
- Scheidsrechter detecteert overtredingen

Scheidsrechter



:: BrainInput

= { referee :: [RefereeAction], ... }

- Overtredingen:
 - hands, incorrect balbezit, gevaarlijk spel, onsportief gedrag
- leidt tot reprimande / spelhervatting:
 - waarschuwing, gele kaart, rode kaart
 - bij rode kaart verwijdering uit team

Scheidsrechter



:: BrainInput

= { referee :: [RefereeAction], ... }

- Spelhervatting:

- vrije trap, inworp, doeltrap, aftrap, corner
- team die deze moet nemen: Home (West / East)
- het andere team is in overtreding als deze toch de actie uitvoert

Scheidsrechter



:: BrainInput

= { referee :: [RefereeAction], ... }

- Signaleren goal:

 :: RefereeAction = ... | Goal Home | ...

 – daarna aftrap door andere team (other)

Scheidsrechter



:: BrainInput

= { referee :: [RefereeAction], ... }

- Signaleren helftwissel:

 :: RefereeAction = ... | EndHalf | ...

– dit moeten spelers zelf bijhouden

brein (input={referee},m={helft2})
 = (... , {m & helft2 = helft2 || einde1})
where einde1 = any isEndHalf referee

Teambuilding



Teambuilding

- Team bestaat uit 11 spelers:

`:: Footballer = {playerID :: FootballerID, ...}`
`:: FootballerID = { clubName :: ClubName`
`, playerNr :: PlayersNumber`
`}`

- Keeper heeft nummer 1
- Veldspelers hebben nummers 2..
- Spelers van team spelen voor dezelfde club

Teambuilding

- Module Team bevat alle (oefen)teams
allAvailableTeams :: [Home FootballField -> Team]
- Team krijgt bij aanvang wedstrijd te horen:
 - waar de eerste helft gespeeld wordt
 - hoe groot het voetbalveld is

mijnteam :: Home FootballField -> Team
:: Team ::= [Footballer]

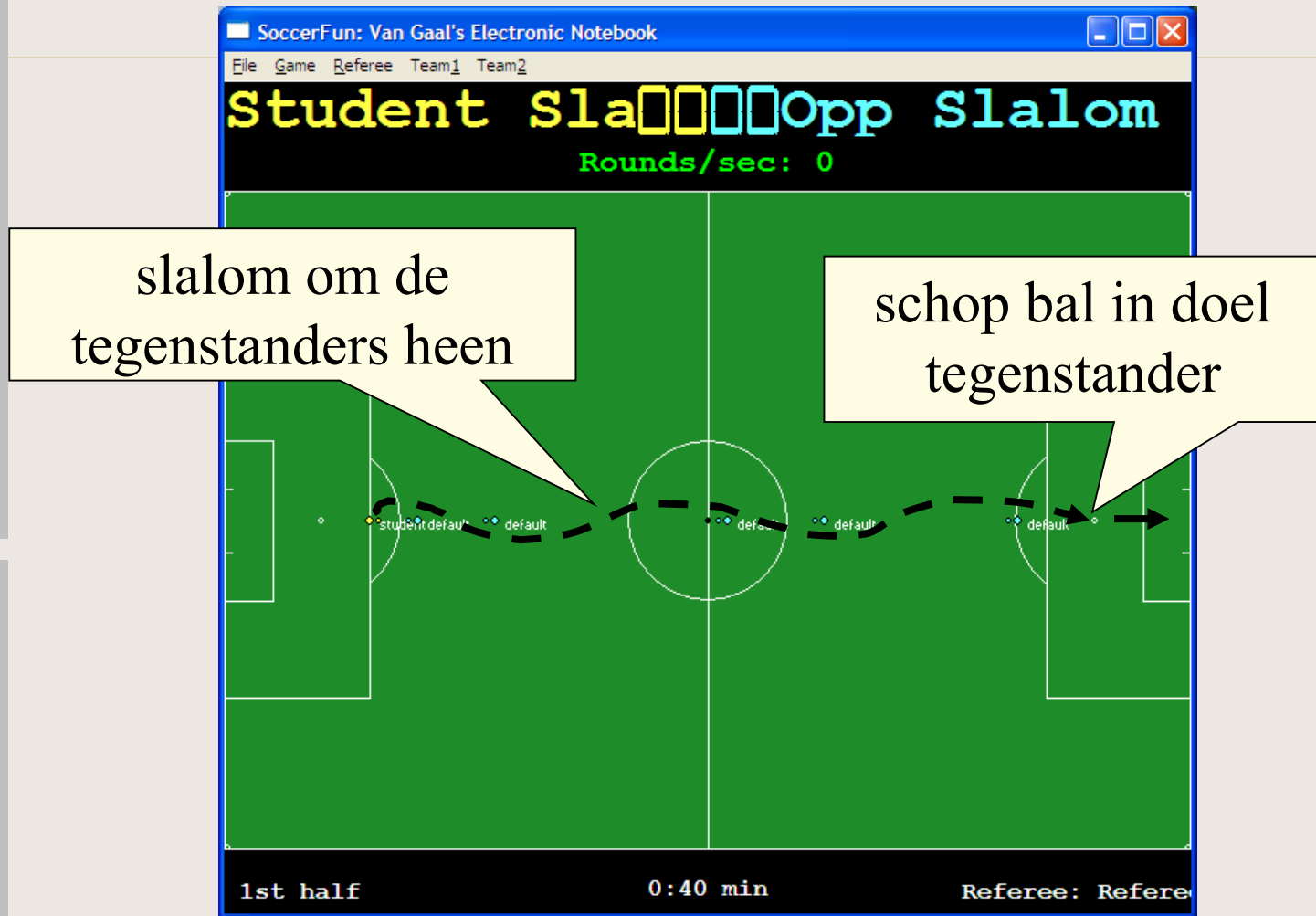
Braintraining



Training 1: slalommen

- Scheidsrechter: RefereeCoach_Slalom
- Tegenstander: Opp_Slalom_W/E
- Jezelf: Student Slalom_E/W
- Module: Team_Student_Slalom_Assignment.icl

Training 1: slalommen

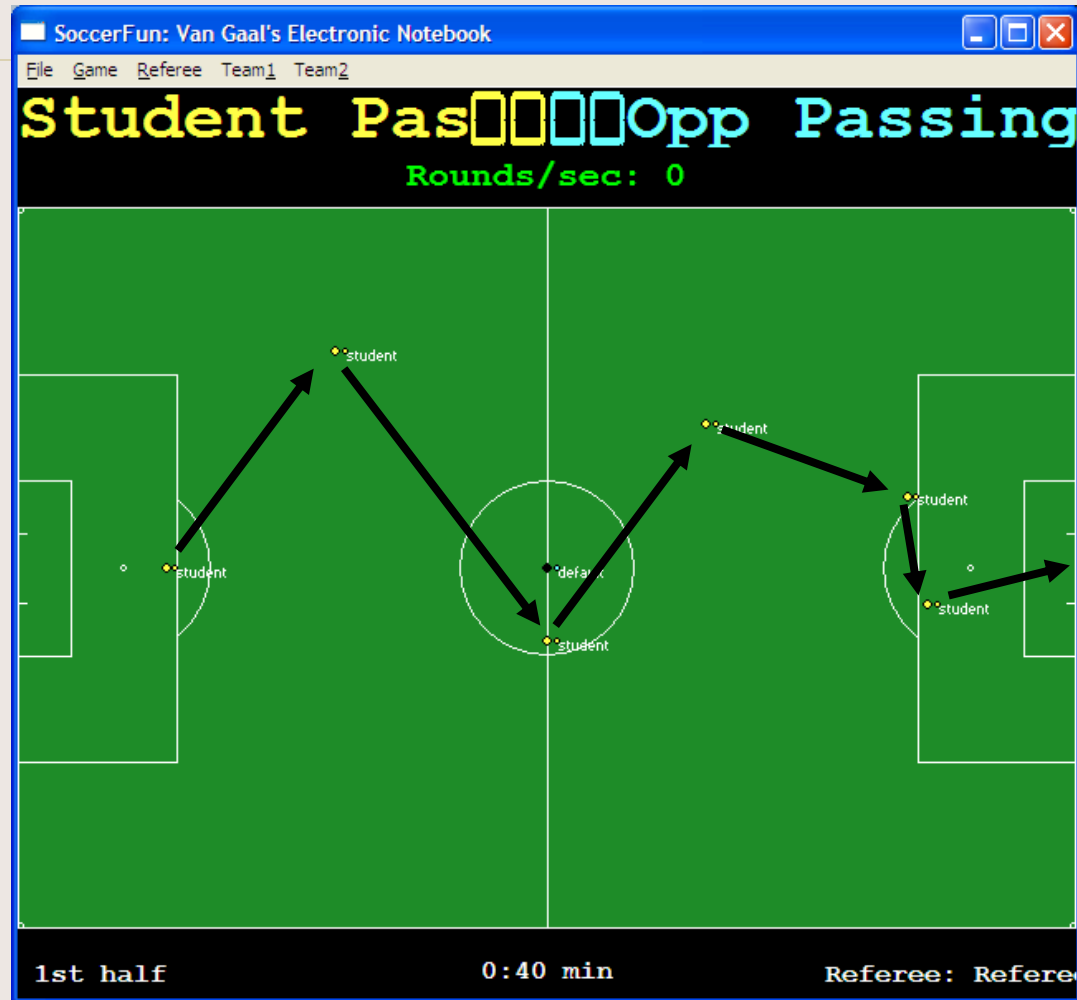


Gebruik predictable mode voor training

Training 2: bal overspelen

- Scheidsrechter: RefereeCoach Passing
- Tegenstander: Opp_Passing_W/E
- Jezelf: Student Passing_E/W
- Module: Team_Student_Passing_Assignment.icl

Training 2: bal overspelen

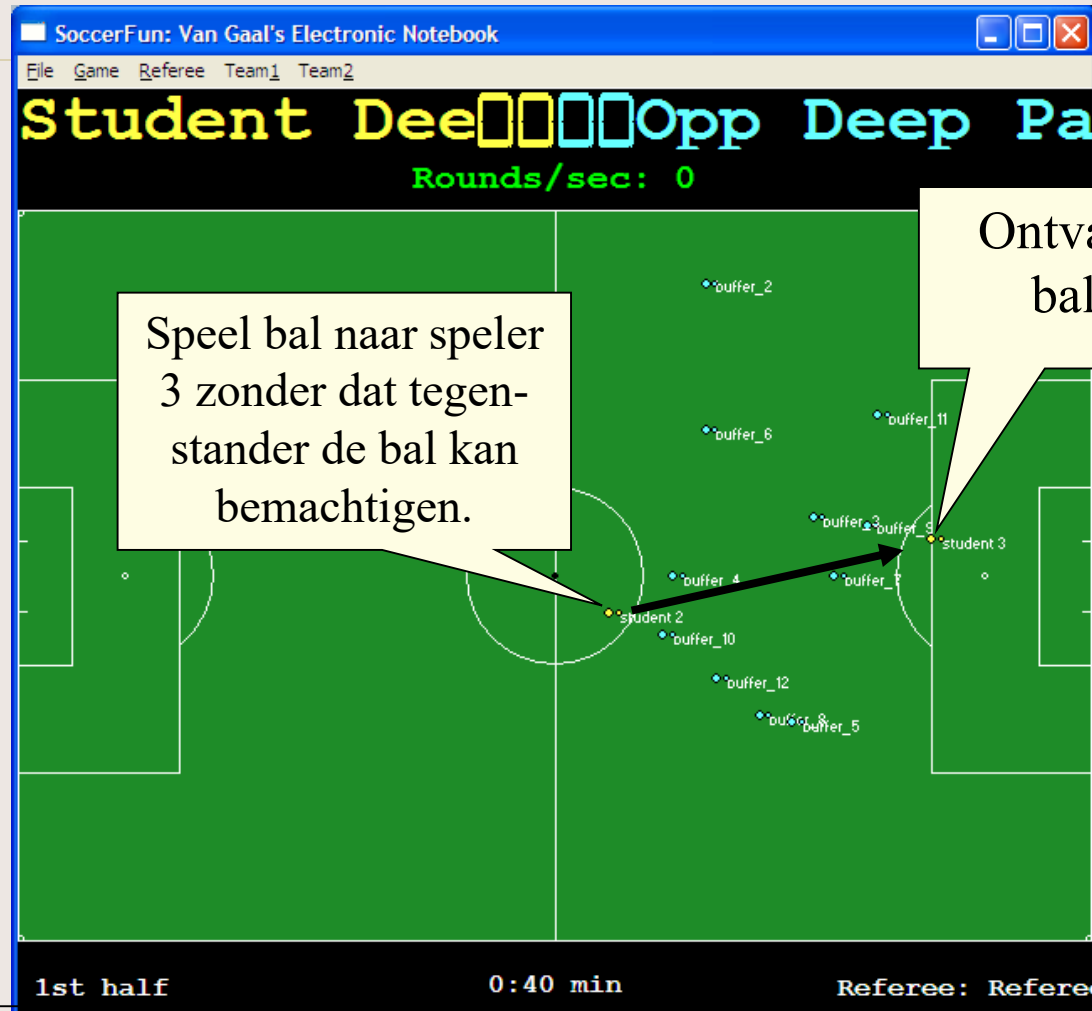


Gebruik predictable mode voor training

Training 3: zinvol aanspelen

- Scheidsrechter: RefereeCoach DeepPass
- Tegenstander: Opp_Deep_Pass_W/E
- Jezelf: Student Deep Pass_E/W
- Module: Team_Student_DeepPass_Assignment.icl

Training 3: zinvol aanspelen



Gebruik predictable mode voor training

Training 4: doel afschermen

- Scheidsrechter: RefereeCoach Keeper
- Tegenstander: Opp_Keeper_W/E
- Jezelf: Student Keeper_E/W
- Module: Team_Student_Keeper_Assignment.icl

Training 4: doel afschermen



Gebruik predictable mode voor training

Meer lezen:

- Overzicht Soccer-Fun: zie SoccerFun\doc\quickstart_SoccerFun.pdf
- De Soccer-Fun home site:
<http://www.cs.ru.nl/P.Achten/SoccerFun/SoccerFun.html>