



NELang



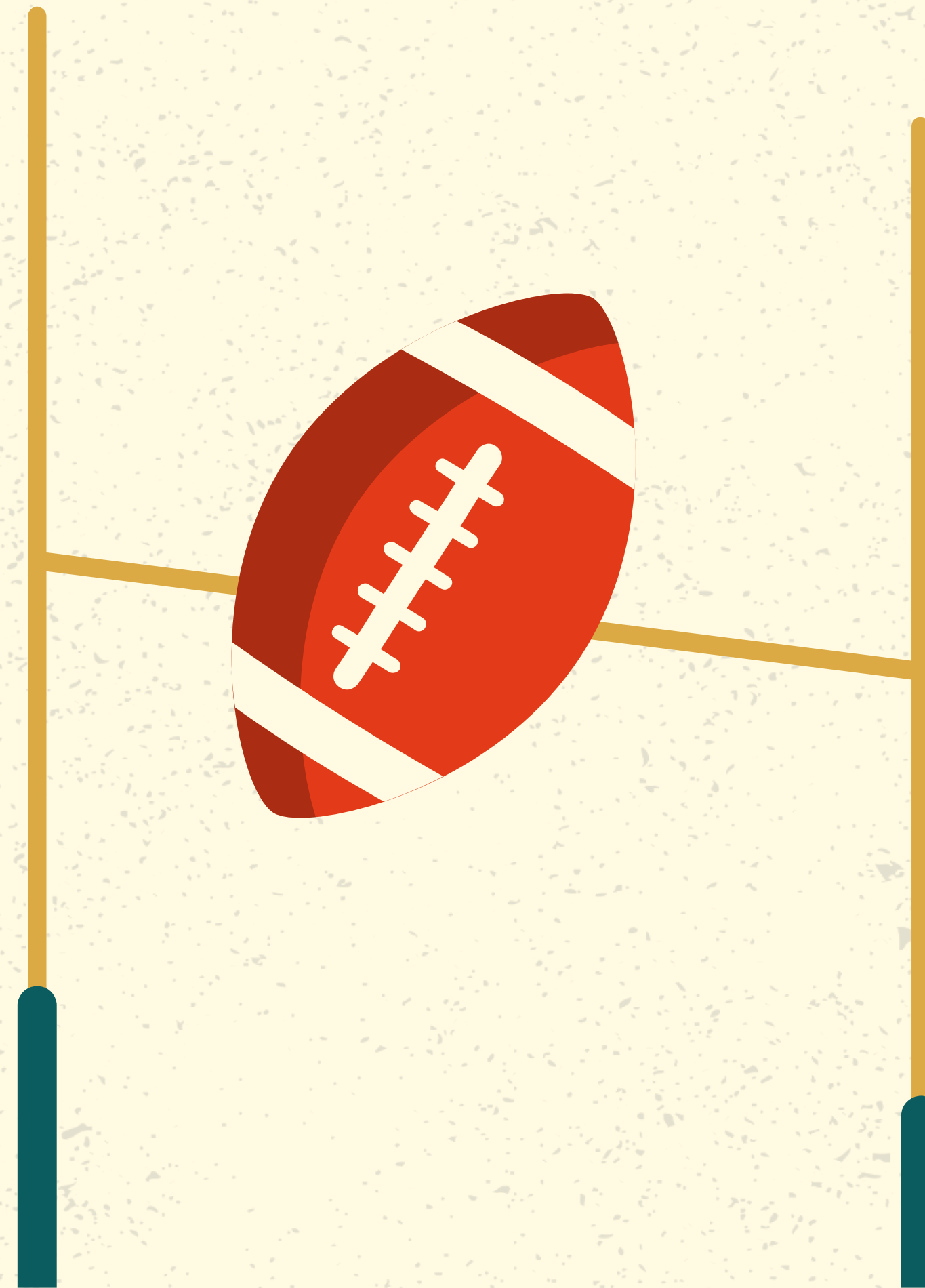
A Linguagem de programação
de Futebol Americano



★ INTRODUÇÃO ★

A NFLang é uma linguagem de programação criada com o objetivo de facilitar o design de jogadas e estratégias do jogo de Futebol Americano.

Sua pretensão é ser específica para o jogo e fácil de ser utilizada por pessoas com pouca familiaridade com programação.





CURIOSIDADES SOBRE NFL



EQUILÍBRIO

Dos 32 times que fazem parte hoje da liga, apenas 4 nunca chegaram ao Super Bowl (Lions, Browns, Texans e Jaguars)

SCORIGAMI

Mapeamento de placares únicos que já ocorreram em uma partida. Atualmente existem 1049 placares diferentes registrados.



SUCESSO

A liga possui um sucesso financeiro enorme, por exemplo, 30 segundos de comercial no Super Bowl em 2023 valeram 7 milhões de dolares (valor mais alto da história).

EVOLUÇÃO

Ao longo dos anos, o esporte passou por diversas transformações, se tornando, cada vez mais, uma liga que incentiva times ofensivos através de suas regras.



★ MOTIVAÇÕES ★

PAIXÃO

Acompanhar a NFL tem sido uma das minhas maiores paixões nos últimos 5 anos. É um esporte que cada vez mais cativa o público brasileiro, que irá sediar um jogo da liga em 2024.

ESTRATÉGIA

O Futebol Americano é um dos esportes mais estratégicos que existem, onde toda movimentação dos 22 jogadores em campo é desenhada a cada jogada. Utilizar uma linguagem de programação pode facilitar a vida de um treinador.



★ CARACTERÍSTICAS ★

TIPADA

A linguagem é fortemente tipada. Existem 5 tipos: athlete, play, number, down e empty.



FUNCIONAL

A linguagem oferece suporte a funções, que podem retornar cada um dos 5 tipos.

CONDICIONAIS E LOOPS

Possui desvios condicionais e dois tipos de loops (drive on, playuntil).



★ ESPECIALIZAÇÕES ★

JOGADAS

É possível definir jogadas (tipo play), onde o usuário consegue definir uma ação para cada posição do jogo.



DOWNS

Tipo especial para contar os *downs* da campanha. É possível fazer um loop em cada *down*, por exemplo.

ATLETAS

Tipo especial para definir um atleta dentro de um jogo. Equivalente a uma string de uma linguagem de programação convencional.





EXEMPLO 1



1. Aqui definimos duas jogadas, uma para a defesa e outra para o ataque. Definimos o atleta que será o QB.

3. Chamamos a função e mostramos o resultado no terminal.

```
play deepBomb is {WR: "left-deep-route", TE: "crossing-route"}
play coverTwo is {CB: "man-to-man", LB: "cover mid field"}
athlete qb is "Patrick Mahomes"

action play readDefense(play offensivePlay, play defensivePlay, athlete qb) {
  when defensivePlay.CB equals "man-to-man" and qb equals "Patrick Mahomes" then {
    offensivePlay.QB is "Throw to TE"
    result offensivePlay
  } otherwise {
    offensivePlay.QB is "Throw deep to WR"
    result offensivePlay
  }
}

play playAfterRead is readDefense(deepBomb, coverTwo)
call(playAfterRead)
```

2. Nesse trecho foi criada uma função para decidir qual será a ação do QB, dependendo da maneira como a defesa está formada.

```
paulo@paulo-Latitude-3420:~/projetosFacul/logcomp/NFlang$ ./nfl examples/play.nfl
{QB: "Throw to TE", TE: "crossing-route", WR: "left-deep-route"}
```




EXEMPLO 2



1. Aqui estamos fazendo um loop de drive (estilo for each do python), nos valores de down de firstdown até fourthdown.

```
number yardsGainedOnDrive is 0
drive down currentDown on (firstdown, fourthdown) {
    call("Current down is " + currentDown)
    call("Enter result of the play")
    yardsGainedOnDrive is yardsGainedOnDrive + signal()
}

call("Yards gained on drive is " + yardsGainedOnDrive)
```

2. Utilizamos a função signal para obter o input do usuário de quantas jardas foram conquistadas.

3. Saída do programa.

```
paulo@paulo-Latitude-3420:~/projetosFacul/logcomp/NFlang$ ./nfl examples/drive.nfl
Current down is firstdown
Enter result of the play
2
Current down is seconddown
Enter result of the play
2
Current down is thirddown
Enter result of the play
1
Current down is fourthdown
Enter result of the play
-3
Yards gained on drive is 2
```




EXEMPLO 3



1. Por fim, um programa mais robusto que simula o último drive de um jogo.

3. Na saída é analisado se o resultado foi ou não touchdown.

```
number yardsLeftOnDrive is 75
number timeLeft is 90
down currentDown is firstdown
number yardsLeftOnDown is 10
number resultOfPlay

playuntil timeLeft equals 0 or currentDown equals toondowns or yardsLeftOnDrive equals 0 {
  call("*****")
  call("Current down is " + currentDown)
  call("Yards left on down is " + yardsLeftOnDown)
  call("Yards left on drive is " + yardsLeftOnDrive)
  call("Time left on drive is " + timeLeft)

  call("The result of the play is ")
  resultOfPlay is signal()
  yardsLeftOnDrive is yardsLeftOnDrive - resultOfPlay
  call("Time consumed in play is ")
  timeLeft is timeLeft - signal()

  yardsLeftOnDown is yardsLeftOnDown - resultOfPlay

  when yardsLeftOnDown < 1 then {
    yardsLeftOnDown is 10
    currentDown is firstdown
  } otherwise {
    currentDown is >>currentDown
  }
}

when yardsLeftOnDrive equals 0 then {
  call("Touchdown!!!")
} otherwise {
  call("Game over!")
}
```

2. Para tanto, definimos um loop playuntil e capturamos os resultados de cada jogada do usuário através de signal. Jogamos até ficarmos sem downs ou o tempo acabar ou marcarmos o touchdown.



Rodando o programa e inserindo inputs até sua finalização.

```

*****
Current down is firstdown
Yards left on down is 10
Yards left on drive is 75
Time left on drive is 90
The result of the play is
30
Time consumed in play is
30
*****
Current down is firstdown
Yards left on down is 10
Yards left on drive is 45
Time left on drive is 60
The result of the play is
5
Time consumed in play is
20
*****
Current down is seconddown
Yards left on down is 5
Yards left on drive is 40
Time left on drive is 40
The result of the play is
3
Time consumed in play is
10
*****

```

```

*****
Current down is firstdown
Yards left on down is 10
Yards left on drive is 17
Time left on drive is 15
The result of the play is
7
Time consumed in play is
8
*****
Current down is seconddown
Yards left on down is 3
Yards left on drive is 10
Time left on drive is 7
The result of the play is
10
Time consumed in play is
7
Touchdown!!!

```



OBRIGADO!