



elm

Erica Riello



elm - Linguagem Funcional

- inferência de tipos;
- imutabilidade
- pequena;
- rápida;
- fácil de manter;



elm - Pacote Graphics

- Formas:
 - `Graphics.Collage.circle`
 - `Graphics.Collage.rect`
- Imagens:
 - `Graphics.Element.image`
- Arranjos:
 - `Graphics.Collage.move`
 - `Graphics.Collage.collage`
 - `Graphics.Element.flow`



elm - exemplo Mapa

```
grey : Color.Color
grey = Color.rgba 55 55 55 1

getCircle : Int -> Float -> Float -> List Graphics.Collage.Form
getCircle i x y =
  case i of
    -1 -> []
    0 -> []
    1 -> (Graphics.Collage.move (x, y) (Graphics.Collage.filled white (Graphics.Collage.circle 1.5)))::[]
    2 -> (Graphics.Collage.move (x, y) (Graphics.Collage.filled white (Graphics.Collage.circle 4)))::[]
    3 -> []
    4 -> []

getPacmanForm : Float -> Float -> Direction -> Graphics.Collage.Form
getPacmanForm x y dir =
  Graphics.Collage.move (x, y) (Graphics.Collage.toForm (Graphics.Element.image 15 15 (pacmanImagePath dir)))

main : Graphics.Element.Element
main =
  Graphics.Element.flow Graphics.Element.outward (
    (Graphics.Collage.collage (28*tileSize) (31*tileSize) (getAllForms++((getPacmanForm (-14*tileSize) (15*tileSize) Right)::
    [])))::[])
```



elm



Se é imutável, como pode ser reativa?



elm - Signals

- Sinais são valores que mudam ao longo de um tempo.
- São a única forma de interatividade em elm.
- Em elm, sinais são tipos que podem ser representados por outros tipos e tuplas.



elm - exemplo Mouse

```
import Graphics.Element exposing (..)
import Mouse

main : Signal Element
main =
    Signal.map show Mouse.position
```

Mouse.position é um sinal que representa a posição do mouse, uma tupla (,) que representa as coordenadas do mouse.

Quando o mouse se move, o valor de Mouse.position muda automaticamente



elm - Pacote Signal

- Sinais:
 - Time.fps
 - Keyboard.arrows
 - Keyboard.shift
- Transformação e combinação de sinais:
 - Signal.map
 - Signal.sampleOn
 - Signal.foldp



elm



Então posso usar sinais para
implementar máquinas de estado!



elm - método foldp

- O método foldp cria uma dependência com o passado
- Para isso ele recebe:
 - uma função que faz a transição entre os estados;
 - um estado inicial;
 - um sinal que, quando o valor mudar, chama a função de transição passando o estado imediatamente anterior



elm - exemplo Pacman

```
type alias Game =
    { pacman : Character }

currentGame : Game
currentGame = { pacman = { ... } }

update : (Time.Time, { x:Int, y:Int }) -> Game -> Game
update (delta, direction) game =
    changeDirection direction game
    |> updateGamePosition delta

render: Game -> Graphics.Element.Element
render game =
    Graphics.Element.flow Graphics.Element.outward(...)

main : Signal Graphics.Element.Element
main =
    Signal.map render (Signal.foldp update currentGame input)

input : Signal (Time.Time, { x:Int, y:Int })
input =
    Signal.sampleOn delta (Signal.map2 (,) delta Keyboard.arrows)

delta : Signal Time.Time
delta =
    Signal.map (\t -> t / 20) (Time.fps 25)
```

Signal.map aplica uma função ao Sinal passado como entrada. Dessa forma, quando o sinal retornado por (Signal.foldp update currentGame input) tiver seu valor alterado, ele será passado como parâmetro à chamada de render.

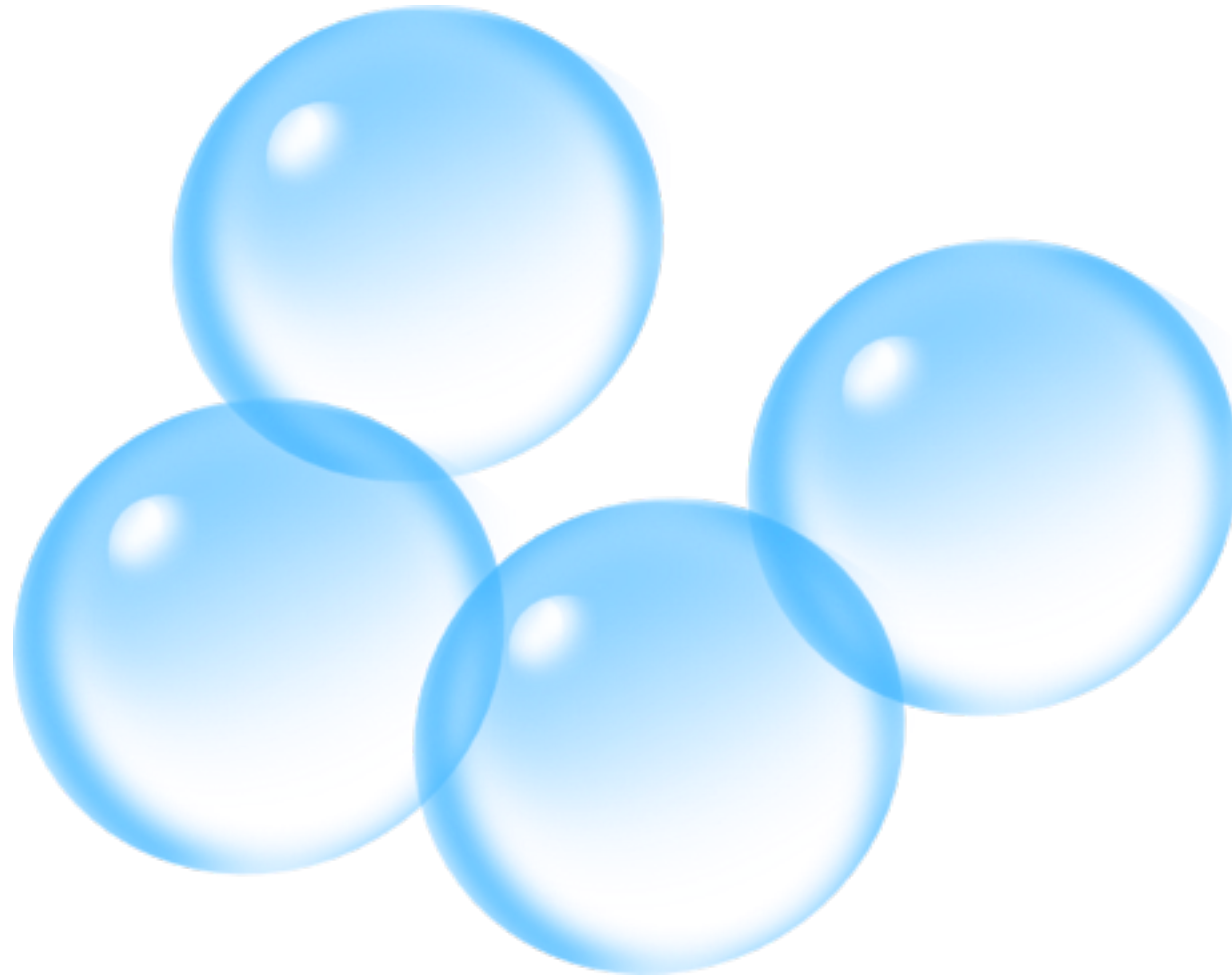
update é a função que faz a transição entre os estados; currentGame é o estado inicial e input o sinal que vai provocar a mudança de estado.

sampleOn faz com que a cada alteração no sinal delta, seja amostrado o sinal retornado por (Signal.map2 (,) delta Keyboard.arrows)

O sinal delta é atualizado a uma taxa de 25 fps, porém o valor dele é o tempo entre o frame corrente e o anterior (t), dividido por 20.



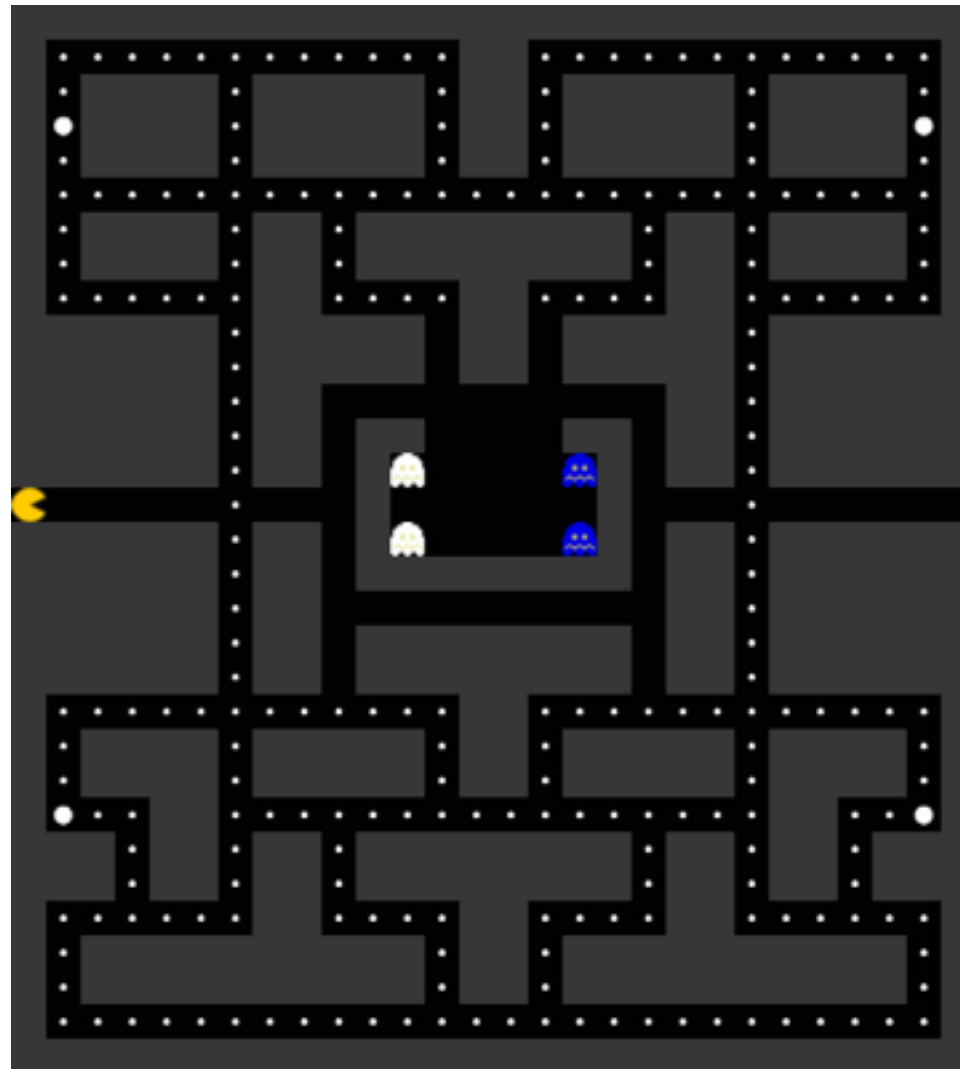
elm - exemplo Fantasma



Dá pra fazer spawn de objetos



elm - exemplo Jogo



E se juntarmos tudo?



Obrigada ;-)

Erica Riello