

Tarefa 1 - Arkanoid - Primeira Parte

Crie um novo projeto chamado *collision*:

```
> stack new collision simple
> cd collision
> stack setup
> stack build
> stack exec collision
```

Substitua o código do arquivo Main.hs por:

```
module Main where

-- | Verifica se tem interseção entre duas faixas de valores.
-- Dado duas tuplas, (xmin, xmax) (ymin, ymax):
-- xmin          xmax
-- |-----|
--          |-----|
--          ymin    ymax
-- você pode verificar se elas se sobrepõem testando se
-- xmin vem antes de ymax
-- e ymin vem antes de xmax
overlap :: Ord a => (a, a) -> (a, a) -> Bool
overlap (xmin, xmax) (ymin, ymax) = undefined

-- | Cria uma faixa de valores centrado em x e com raio r.
-- retorna a tupla contendo x-r e x+r
range :: Num a => a -> a -> (a, a)
range x r = undefined

-- | Retorna se a bola colidiu com o jogador:
-- (px, py) é o centro do retângulo representando o jogador
-- (l, a) é a largura e altura do retângulo
-- (x, y) é o centro da bola
-- r é o raio da bola
paddleCollision :: (Float, Float) -> (Float, Float) -> (Float, Float) -> Float -> Bool
paddleCollision (px, py) (l, a) (x, y) r = yCollision && xCollision
  where
    yCollision = undefined
    xCollision = undefined
    metadeAltura = a / 2
    metadeLargura = l / 2

-- | Verifica se a bola atinge a borda de algum bloco em um dos eixos
-- (vertical ou horizontal)
{-
```

Entrada:

(bx, by) - coordenada do centro do bloco
(l, a) - largura e altura do bloco (a coordenada horizontal vai de
bx - 0.5l a bx + 0.5l)
(cx, cy) - coordenada do centro da bola
r - raio da bola

Verificar se o lado esquerdo da bola colide com o canto direito do bloco OU
se o lado direito da bola colide com o canto esquerdo do bloco
E
se o lado de cima da bola colide com o canto inferior do bloco OU
se o lado de baixo da bola colide com o canto superior do bloco

Os cantos horizontais do bloco são definidos pelas coordenadas:

bx - 0.5l, bx - 0.3l e
bx + 0.3l, bx + 0.5l

Os cantos verticais do bloco são definidos pelas coordenadas:

by - 0.5a, by - 0.3a
by + 0.3a, by - 0.5a

Utilizando a função overlap, complete o código abaixo

-}

```
inCorner :: (Float, Float) -> (Float, Float) -> (Float, Float) -> Float -> Bool
inCorner (bx, by) (l, a) (cx, cy) r = (leftOfBall || rightOfBall)
                                     && (topOfBall || bottomOfBall)
```

where

```
leftOfBall    = undefined
rightOfBall   = undefined
bottomOfBall  = undefined
topOfBall     = undefined
(lmin, lmax)  = (0.3*l, 0.5*l)
(amin, amax)  = (0.3*a, 0.5*a)
(cxmin, cxmax) = range cx r
(cymin, cymax) = range cy r
```

main :: IO ()

main = do

```
print (overlap (1, 2) (3, 4))           -- False
print (overlap (1, 3) (2, 4))           -- True
print (range 5 1)                        -- (4,6)
print (paddleCollision (0, 0) (10, 5) (15, 10) 3) -- False
print (paddleCollision (0, 0) (10, 5) (0, 0) 3)   -- True
print (inCorner (0,0) (10, 6) (6,-4) 2)           -- True
print (inCorner (0,0) (10, 6) (0,0) 2)            -- False
```

Substitua as instruções `undefined` pelo código correto, compile e execute. Esse código será reaproveitado para o desenvolvimento do jogo Arkanoid na próxima aula!