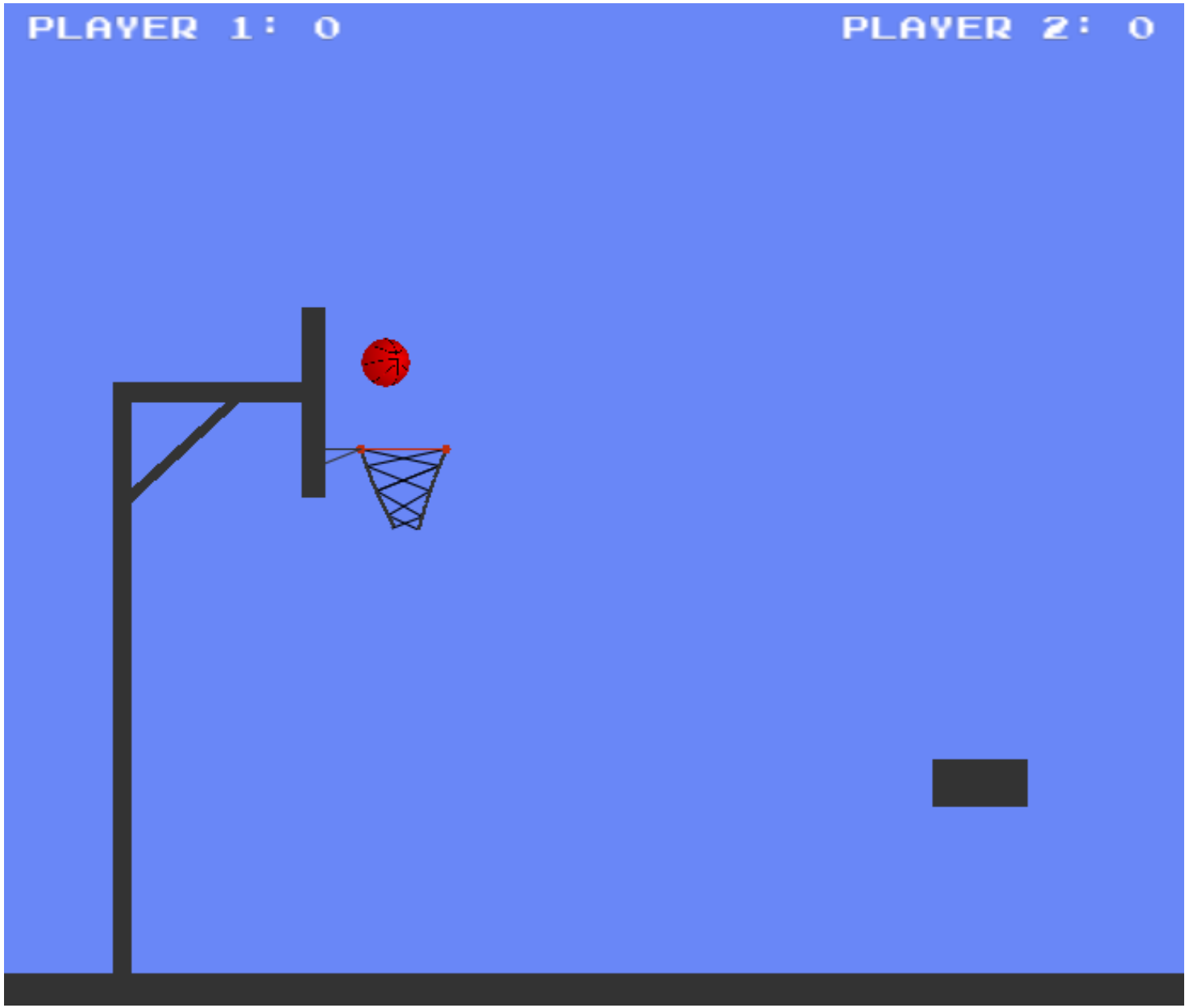
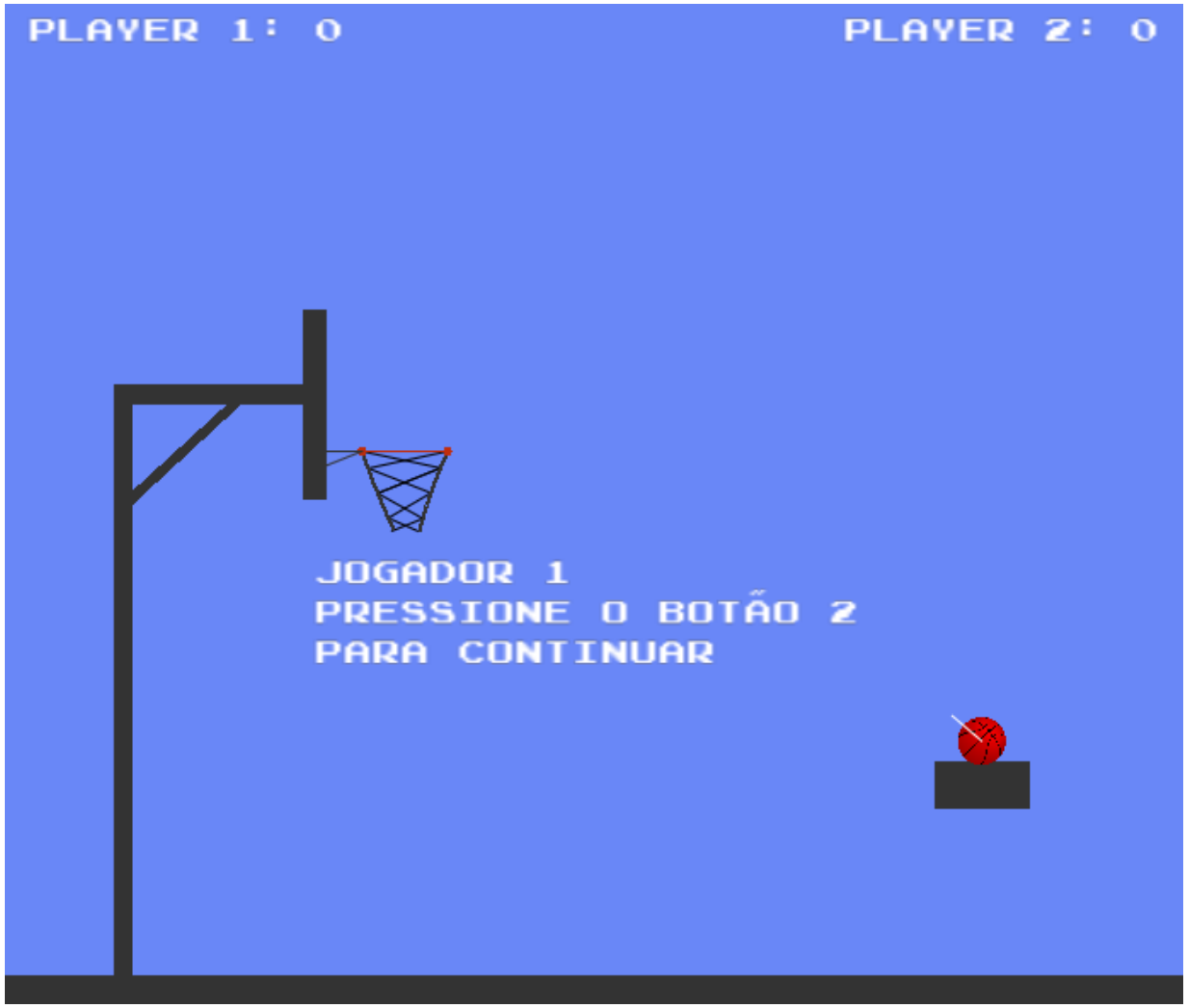


Projeto Final

Danillo Catalão



- Jogo de arremesos livres
- 2 jogadores
- Cada jogador se utiliza de um nodeMcu
- Como só há 2 botões, um deles é responsável para iniciar o turno de um jogador e o outro é para o arremesso
- Um vetor que parte do centro da bola é rotacionado a cada update e indica em que direção será feito o arremesso.

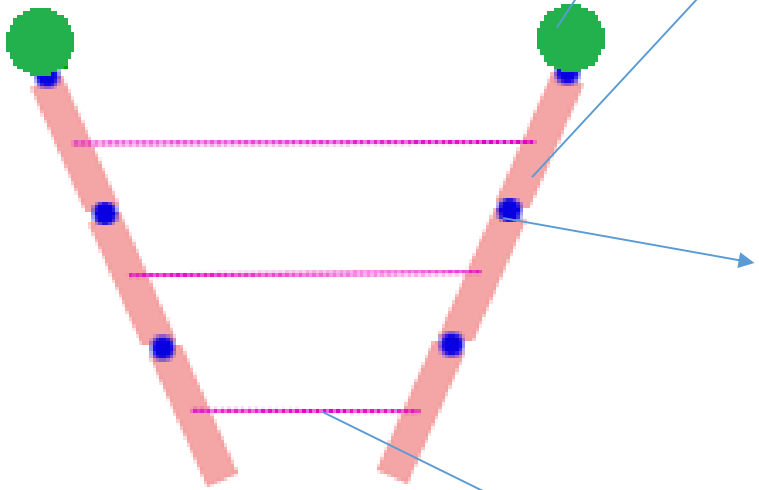
Utilização do love.physics

```
objects.ball = {}
objects.ball.motion = false
objects.ball.body = p.newBody(world, 700, 355, "dynamic")
objects.ball.shape = p.newCircleShape(20)
objects.ball.fixture = p.newFixture(objects.ball.body, objects.ball.shape, 1)
objects.ball.fixture:setRestitution(0.50)
--camera:lookAt(objects.ball.body:getX(), objects.ball.body:getY())

objects.block1 = {}
objects.block1.body = p.newBody(world, -200, 300)
objects.block1.shape = p.newRectangleShape( 20, 600)
objects.block1.fixture = p.newFixture(objects.block1.body, objects.block1.shape, 5)

objects.block2 = {}
objects.block2.body = p.newBody(world, -102.5, -10)
objects.block2.shape = p.newRectangleShape( 215, 20)
objects.block2.fixture = p.newFixture(objects.block2.body, objects.block2.shape, 5)

objects.block3 = {}
objects.block3.body = p.newBody(world, 0, 0)
objects.block3.shape = p.newRectangleShape( 25, 200)
objects.block3.fixture = p.newFixture(objects.block3.body, objects.block3.shape, 5)

objects.block4 = {}
objects.block4.body = p.newBody(world, 0, 0)
objects.block4.shape = p.newPolygonShape(-200,100,-200,115,-65,-10,-80,-10)
objects.block4.fixture = p.newFixture(objects.block4.body, objects.block4.shape, 5)
```

- Criação de corpos
  - Estáticos ou dinâmicos
- Forma
- Junção corpo/forma
  - Densidade
  - restituição

```
p.setMeter(100)
world = p.newWorld(0, 9.81*63, true)
objects = {}

createObjects()
```

```
function love.update(dt)
  world:update(dt)
```



```
objects.hoop1 = {}
objects.hoop1.body = p.newBody(world, 50, 50)
objects.hoop1.shape = p.newCircleShape(5)
objects.hoop1.fixture = p.newFixture(objects.hoop1.body, objects.hoop1.shape, 5)
```

```
objects.net11 = {}
objects.net11.body = p.newBody(world,50,69,"dynamic")
objects.net11.shape = p.newRectangleShape(4,28)
objects.net11.fixture = p.newFixture(objects.net11.body, objects.net11.shape)
```

```
jointNetHoop1 = love.physics.newRevoluteJoint( objects.net11.body, objects.hoop1.body, 50, 50, true )
jointNetNet11 = love.physics.newRevoluteJoint( objects.net11.body, objects.net12.body, 50, 84, true )
jointNetNet12 = love.physics.newRevoluteJoint( objects.net12.body, objects.net13.body, 50, 112, true )

jointNetHoop2 = love.physics.newRevoluteJoint( objects.net21.body, objects.hoop2.body, 140, 50, true )
jointNetNet21 = love.physics.newRevoluteJoint( objects.net21.body, objects.net22.body, 140, 84, true )
jointNetNet22 = love.physics.newRevoluteJoint( objects.net22.body, objects.net23.body, 140, 112, true )
```

```
jointNets1 = love.physics.newDistanceJoint( objects.net11.body, objects.net21.body, objects.net11.body:getX(),objects.net11.body:getY(),
                        objects.net21.body:getX(),objects.net21.body:getY(), false )
jointNets1:setLength(60)
jointNets1:setFrequency(2)
jointNets1:setDampingRatio( 0.4 )
```

```lua
function throw()
    return function()
        m:publish("throw","1",0,0)
    end
end

function turn()
    return function()
        m:publish("turn","1",0,0)
    end
end


gpio.trig(sw1, "down", throw() )
gpio.trig(sw2, "down", turn() )
```

```lua
function throw()
    return function()
        m:publish("throw","2")
    end
end

function turn()
    return function()
        m:publish("turn","2")
    end
end


gpio.trig(sw1, "down", throw() )
gpio.trig(sw2, "down", turn() )
```

Arquivo de cada mcu só muda o identificador de jogador.O jogo controla o fluxo dos turnos sabendo de quem ele espera mensagens.