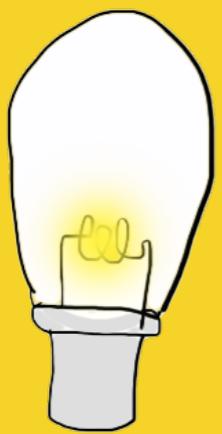


Game in Haskell

Press <Enter> to start

@elise_huard



Concept



Game Idea Machine @gameideamachine · 2h

A game where one of the bosses is a big super powered worm which mirrors your actions and is vulnerable when it charges up



Game Idea Machine @gameideamachine · 6h

A racing game where you ride motorbikes through a dark arena



Game Idea Machine @gameideamachine · 10h

A game where you carry a backpack which is attached to a pair of glasses but has massive recoil



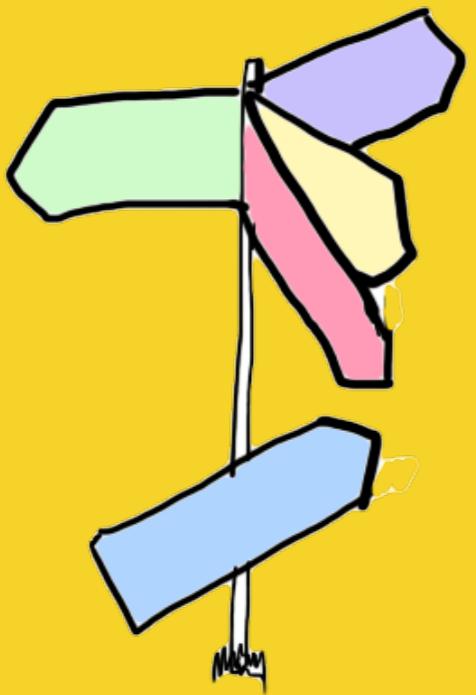
Game Idea Machine @gameideamachine · 14h

A game where you use a sword that sparkles and damages the environment



Game Idea Machine @gameideamachine · 18h

A bohemian student project with difficult moral choices and a focus on loneliness



choices

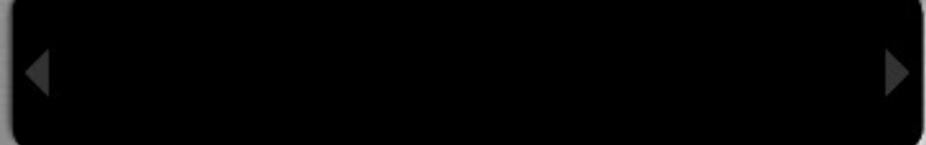
THE HITCHHIKER'S GUIDE TO THE GALAXY
Infocom interactive fiction - a science fiction story
Copyright (c) 1984 by Infocom, Inc. All rights reserved.
Release 59 / Serial number 851108

You wake up. The room is spinning very gently round your head. Or at least it would be if you could see it which you can't.

It is pitch black.

> _

**The Hitchhiker's Guide To The Galaxy
30th Anniversary Edition**



+25 518 (+5) 2 0/500 0/50 (+0) 1 9 8 2

Turn: 107 1740 AD | HELP | MENU





<https://www.bioshockinfinite.com/>



<http://www.nintendo.co.jp/>

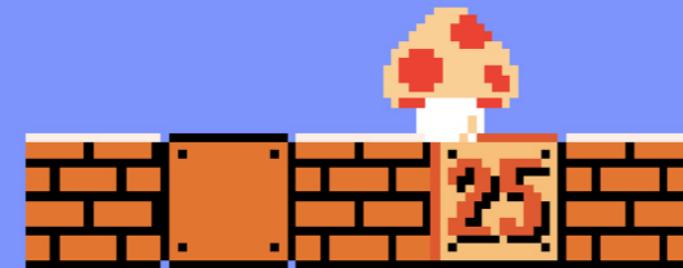
©2010 Nintendo

MARIO
000000

0 x 00

WORLD
1-1

TIME
913





G x 07
O x 00
P x 01

ARROWS

SPACE

-LIFE-

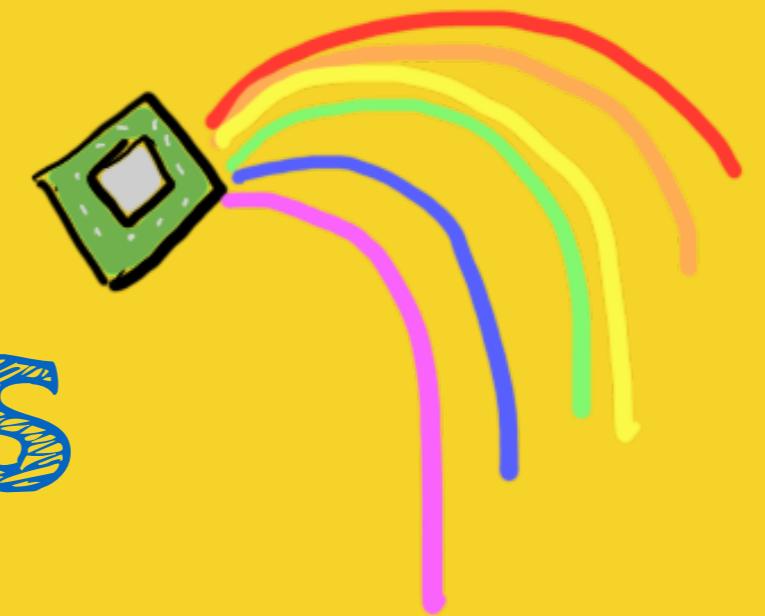


V THE HIEROPHANT



Graphics

with OpenGL



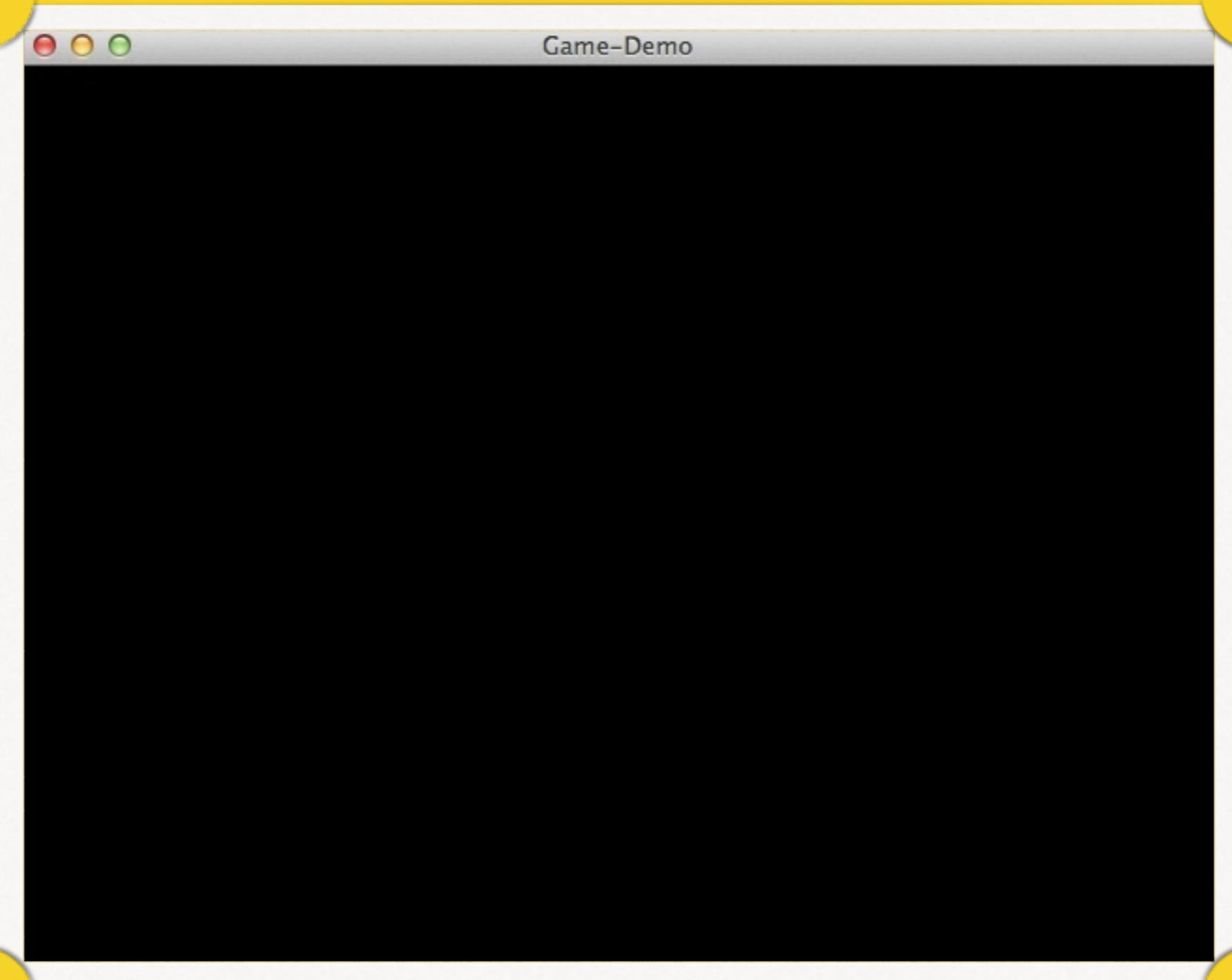
GLFW-b: window

```
main = do
    withWindow width height "Game-Demo" $ \win -> do
        initGL width height
        ...
    ...
```

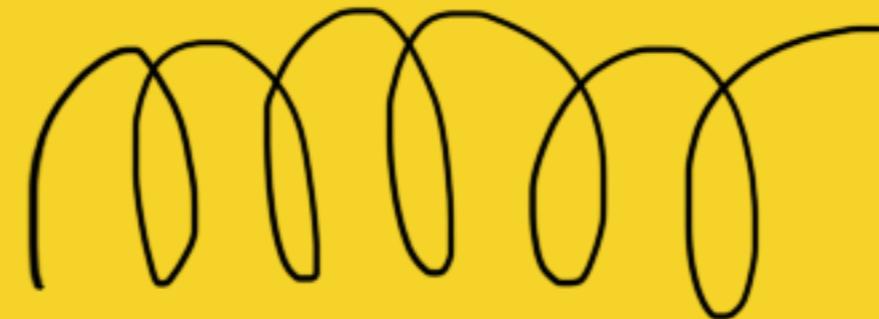
<https://github.com/bsl/GLFW-b-demo>

OpenGL

```
initGL width height = do
    clearColor $= Color4 1 1 1 1
    viewport $= (Position 0 0,
                 Size (fromIntegral width)
                       (fromIntegral height))
    ortho 0 (fromIntegral width)
            0 (fromIntegral height) (-1) 1
```



The Loop



loop:

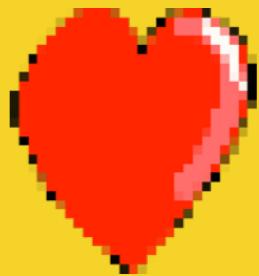
poll input events

react to input (change state)

render state

if not exit, loop again

No life without
state



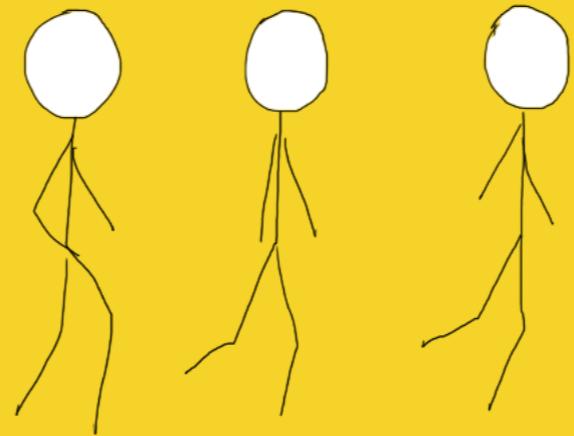
```
initialState = State { x = 200, y = 200 }

loop window state = do
    threadDelay 20000
    pollEvents
    (left, right, up, down) <- getInputKeys window
    let newState = movePlayer (left,right,up,down) state
    renderFrame newState window
    if k
        then return ()
        else loop window newState
```

```
initialState = State { x = 200, y = 200 }
```

```
loop window state = do
    threadDelay 20000
    pollEvents
    (left, right, up, down) <- getInputKeys window
    let newState = movePlayer (left,right,up,down) state
    renderFrame newState window
    if k
        then return ()
        else loop window newState
```

Frame rate



State: FRP

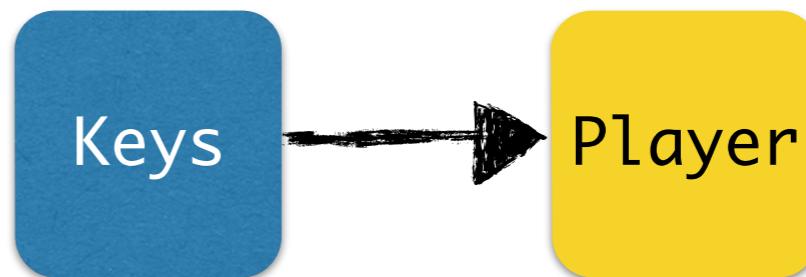
Left Key signal

False False True True True



Player position

(200,200) (200,200) (195,200) (190,200) (185,200)



```
initialState = State { x = 200, y = 200 }

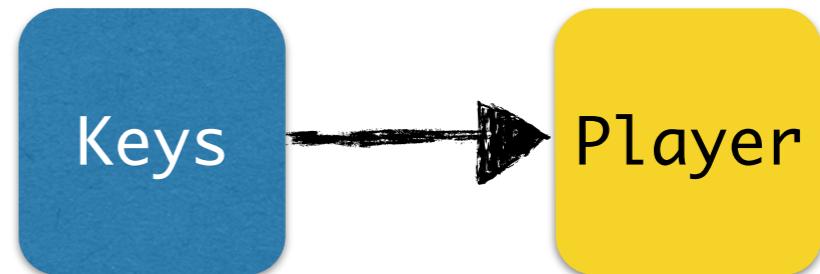
...
loop window initialState

...
loop window state = do
    threadDelay 20000
    pollEvents
    (left, right, up, down) <- getInputKeys window
    let newState = movePlayer (left,right,up,down) state 10
    renderFrame newState window
    if k
        then return ()
        else loop window newState
```

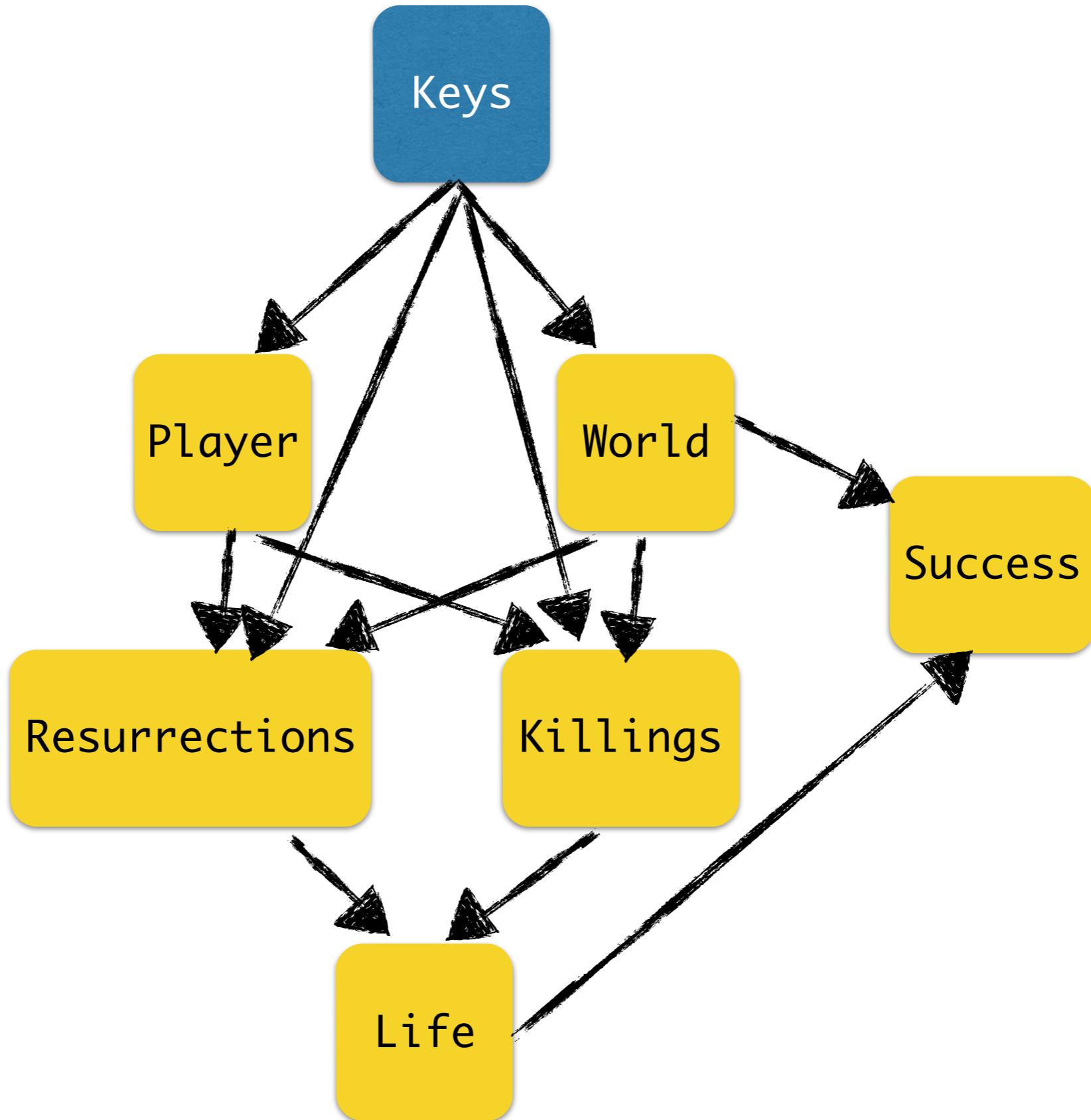
```
(directionKey, directionKeySink) <-
    external (False, False, False, False)

network <- start $ do
    state <- transfer initialState movePlayer directionKey
    return $ renderFrame win <$> state

fix $ \loop -> do
    readKeys win directionKeySink
    join network
    threadDelay 20000
    esc <- keyIsPressed win Key'Escape
    when (not esc) loop
```

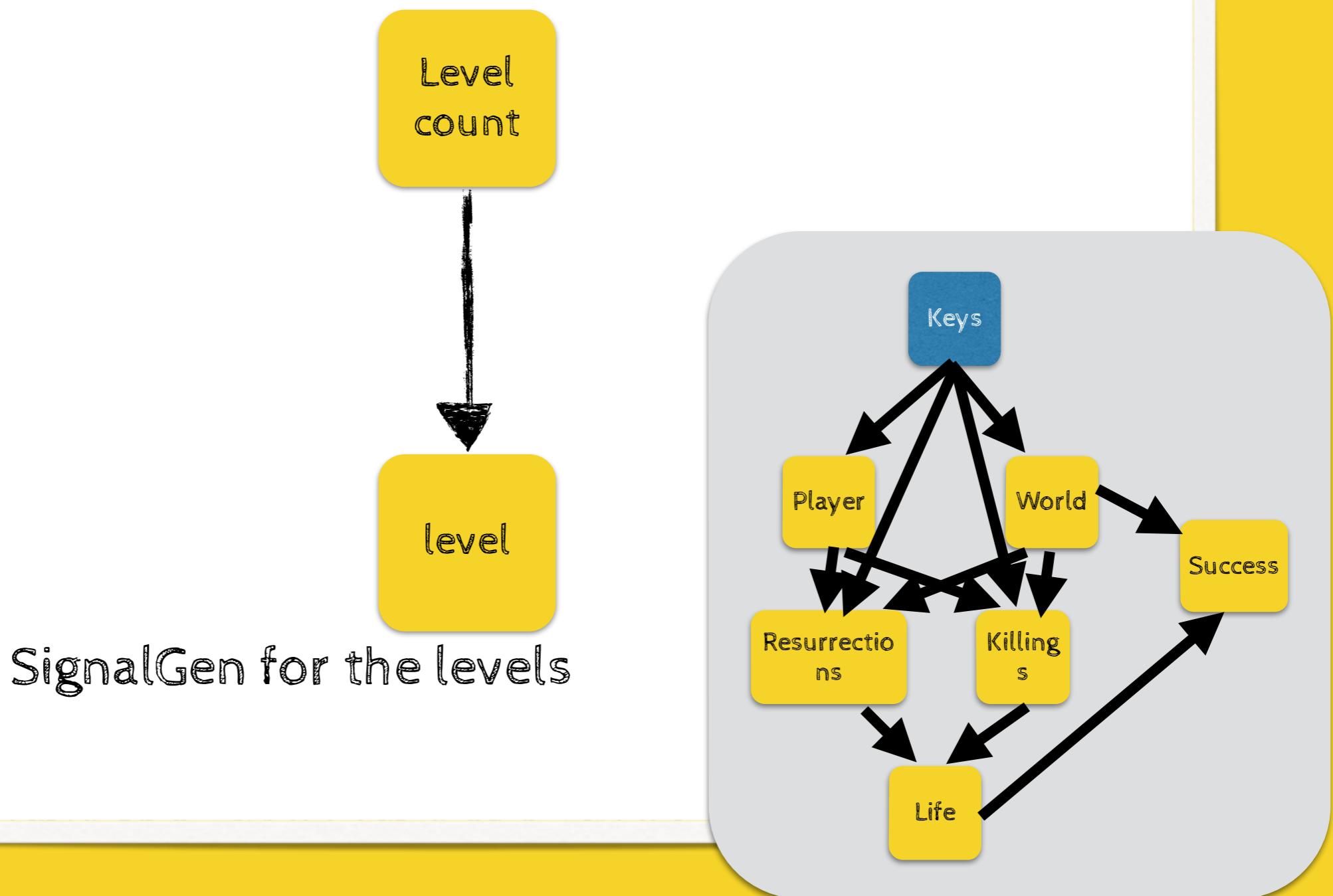


Elerea



FRP and levels

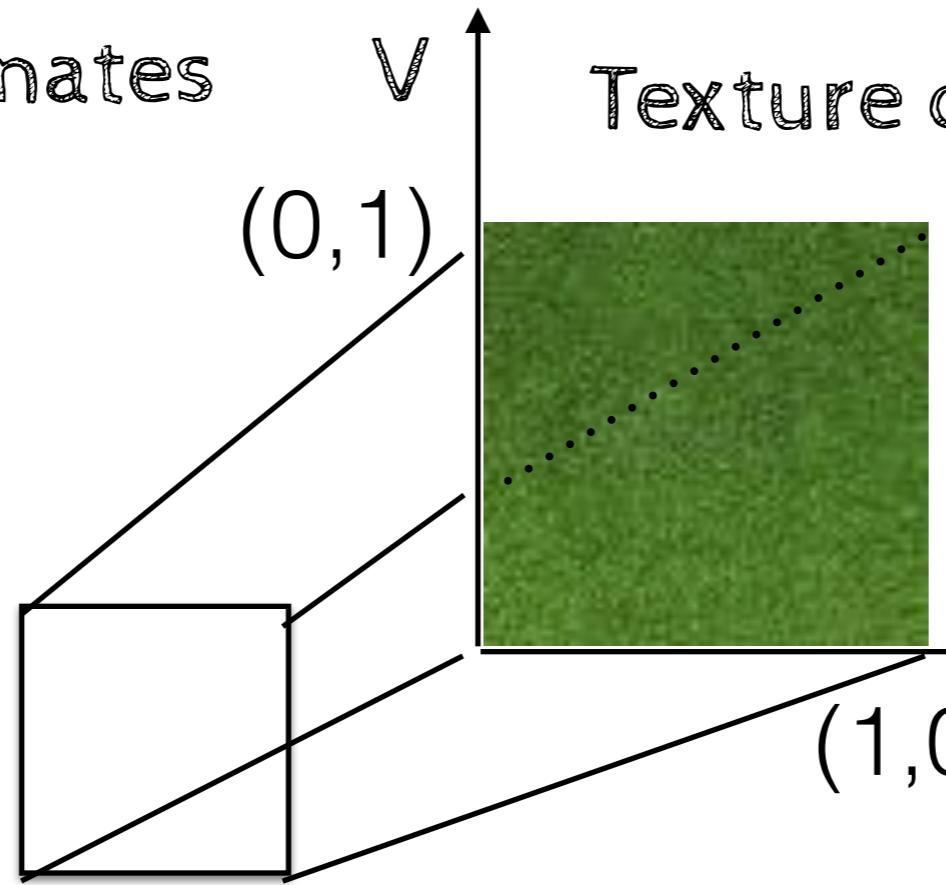
Network



Textures

Vertex coordinates

Y



Texture coordinates

X

(0,1)

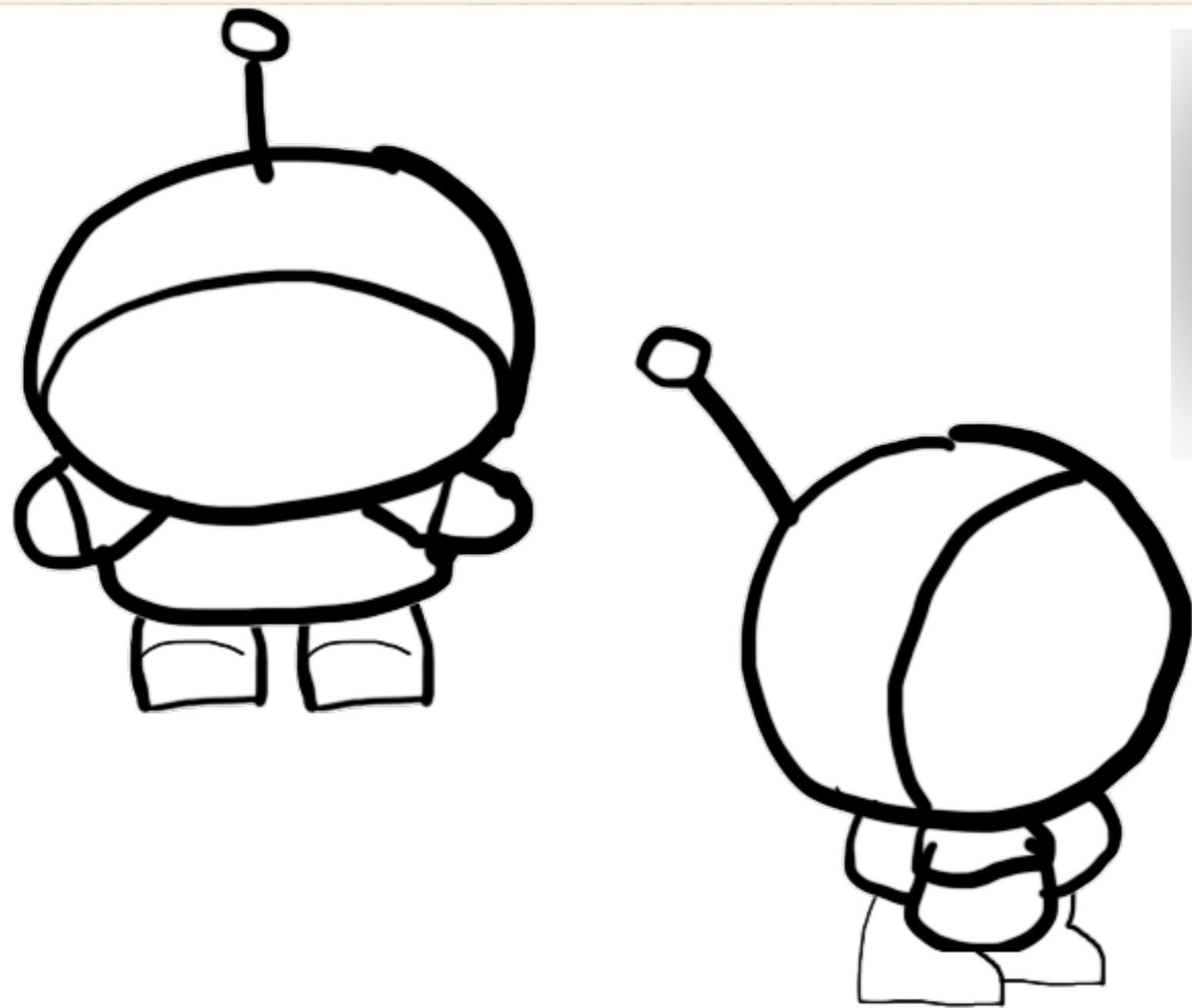
(1,0)

U

V



<http://www.wacom.com/en/ca/creative/intuos-s>







Sound

with OpenAL

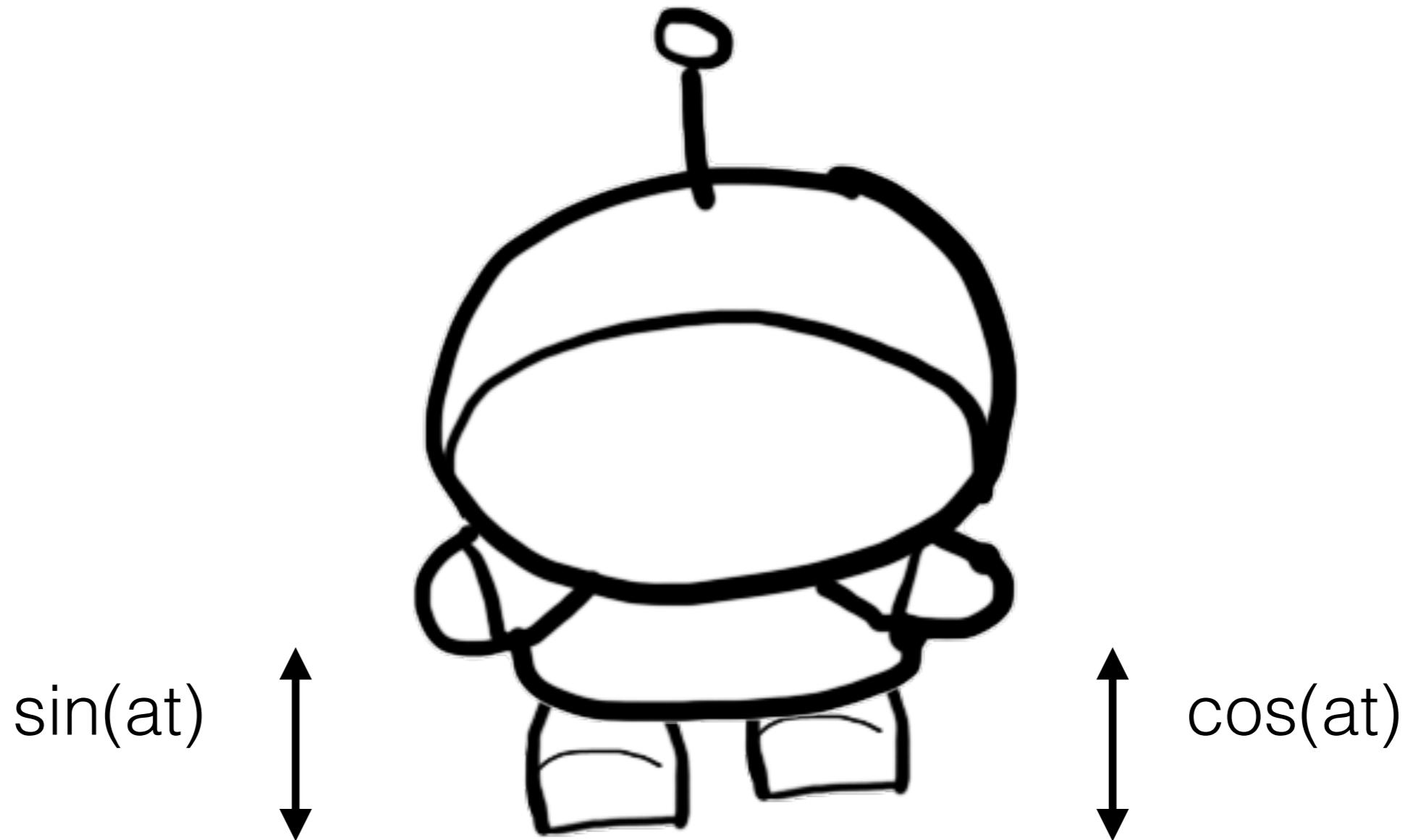
```
playSound path = do
    withProgNameAndArgs runALUT $ \progName args -> do
        buf <- createBuffer (File path)
        -- Generate a single source, attach the buffer to it and
        start playing.
        source <- genObjectName
        buffer source $= Just buf
        play [source]
        errs <- get alErrors
        unless (null errs) $ do
            hPutStrLn stderr (concat (intersperse ", " [ d |
ALError _ d <- errs ]))
        return ()
```

```
outputFunction levelState soundSignals = (renderFrame
levelState) >> (playSounds soundSignals)

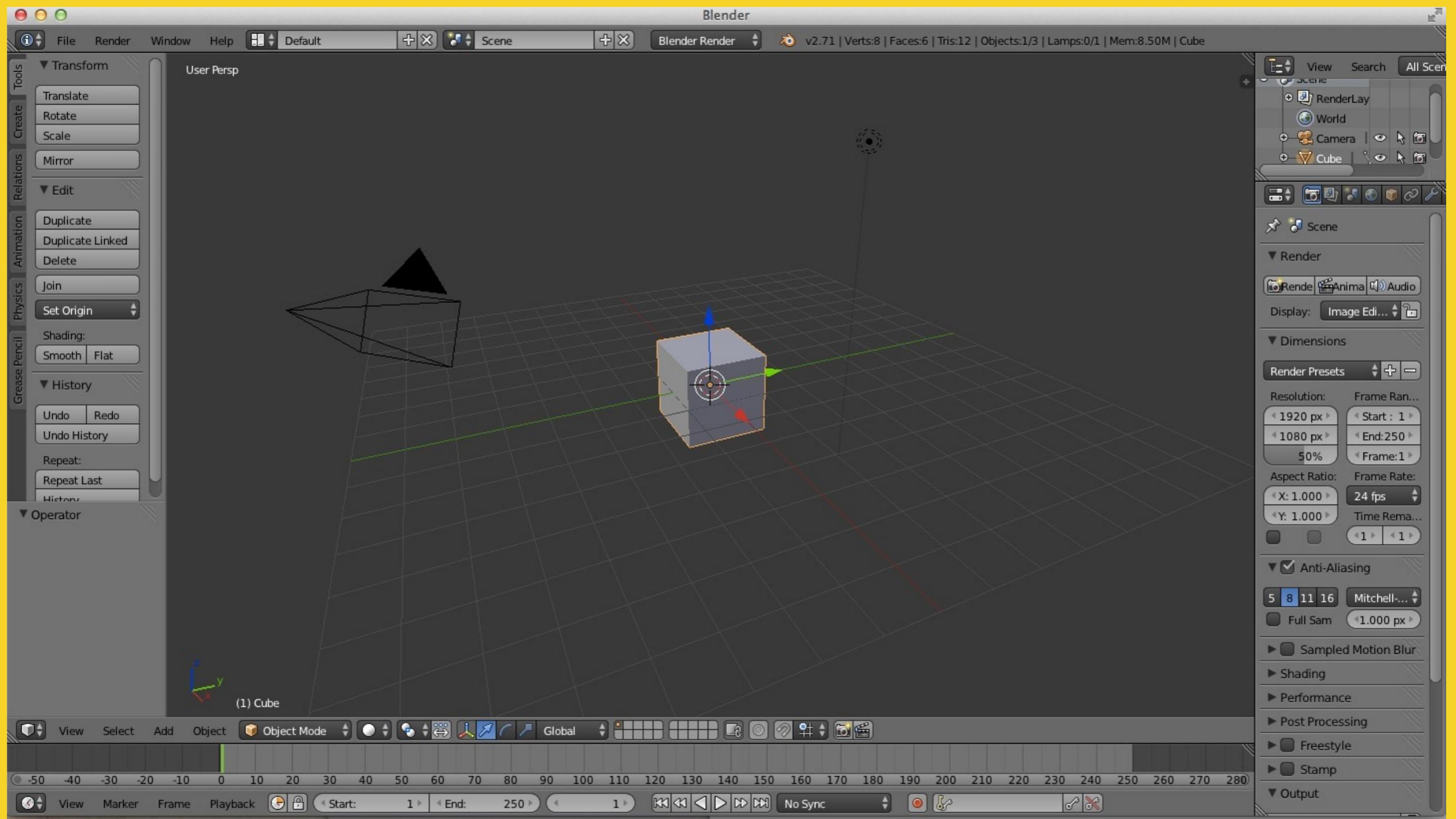
network <- start $ do
    state <- transfer initialState movePlayer directionKey
    return $ renderFrame <$> state

network <- start $ do
    state <- transfer initialState movePlayer directionKey
    ...
    return $ outputFunction <$> state <*> soundState
```

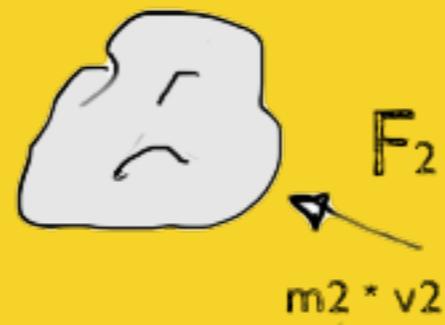
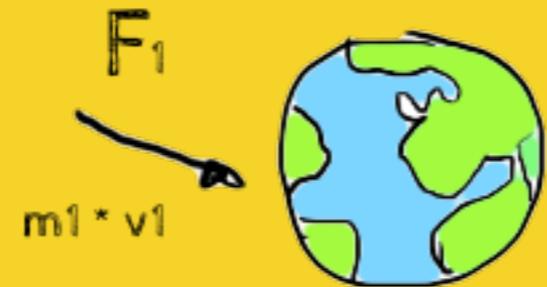
Animations

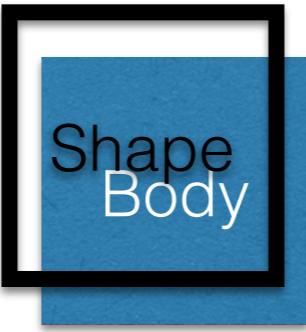


Keyframe animations
Skeletal animations



Physics





Gravity



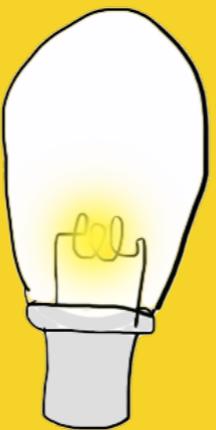
Friction, elasticity, joints, ...

So far



<https://vimeo.com/106632572>

Lessons Learned



Haskell

Types

Pattern matching

nice bindings

Haskell

unanswered questions: packaging,
porting

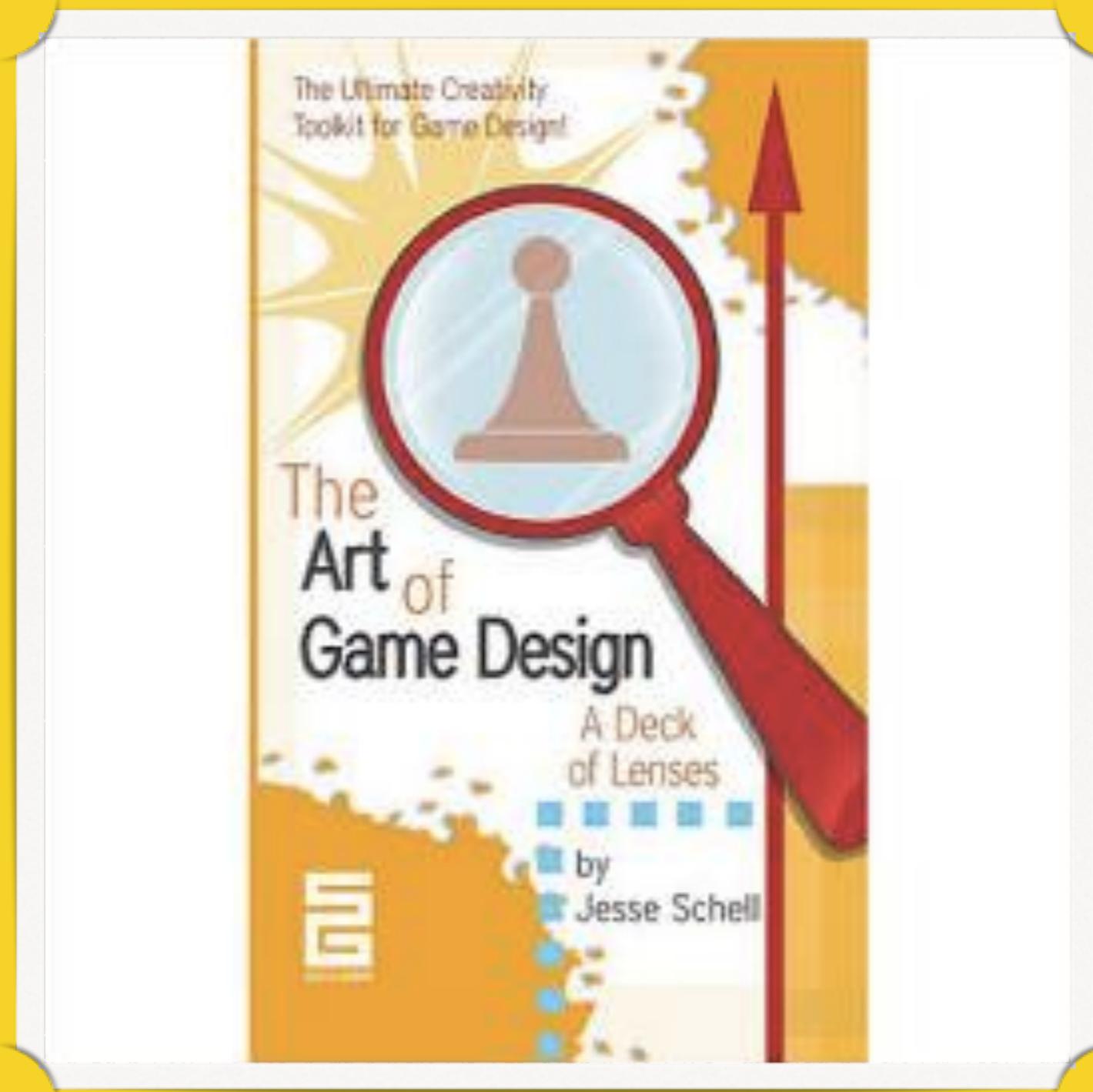
performance?

strict (eager) when possible

getting things started was painful

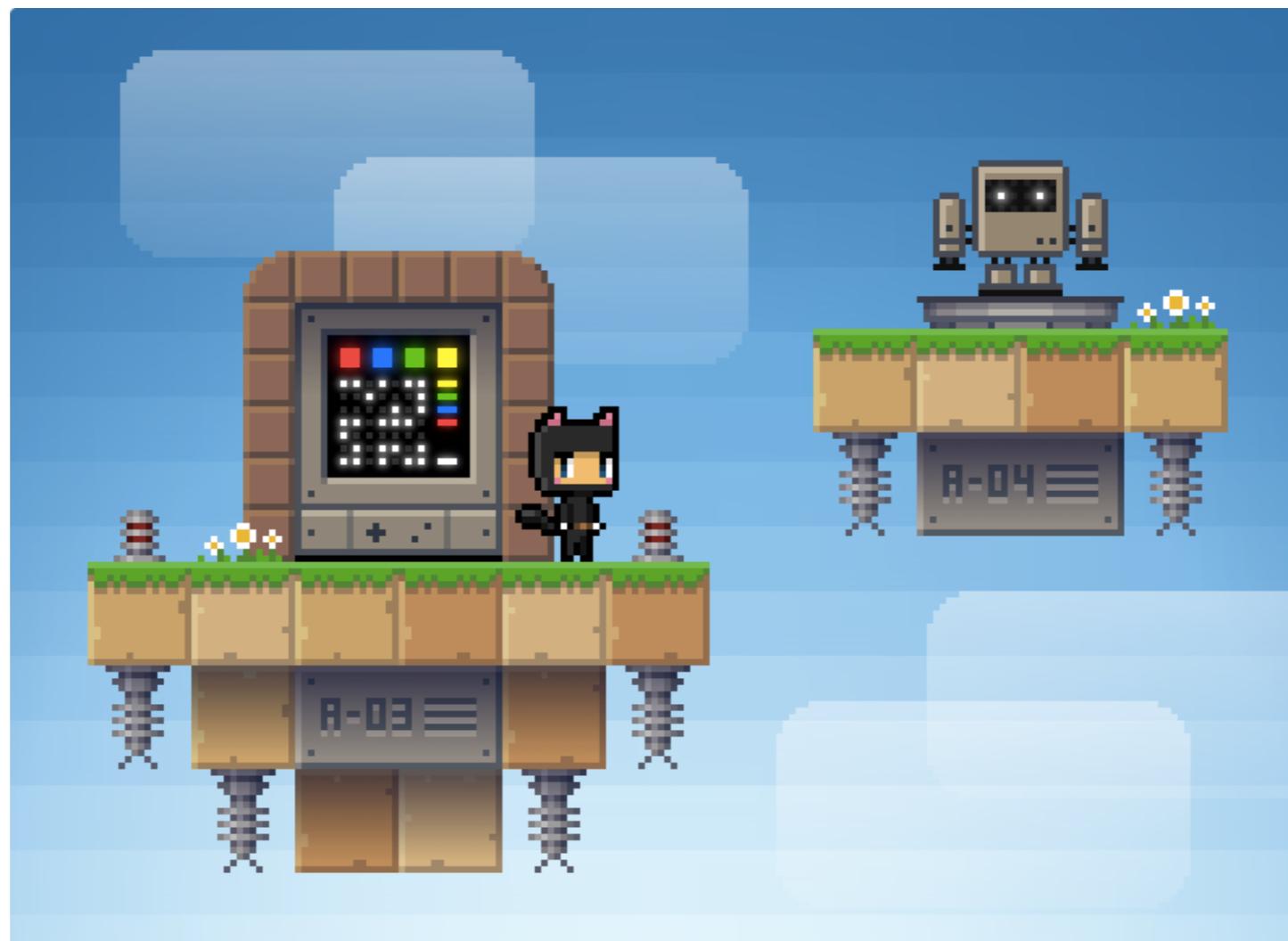
compilation ...

Game



Nikki and the robots

[http://steamcommunity.com/sharedfiles/
filedetails/?id=107105028](http://steamcommunity.com/sharedfiles/filedetails/?id=107105028)



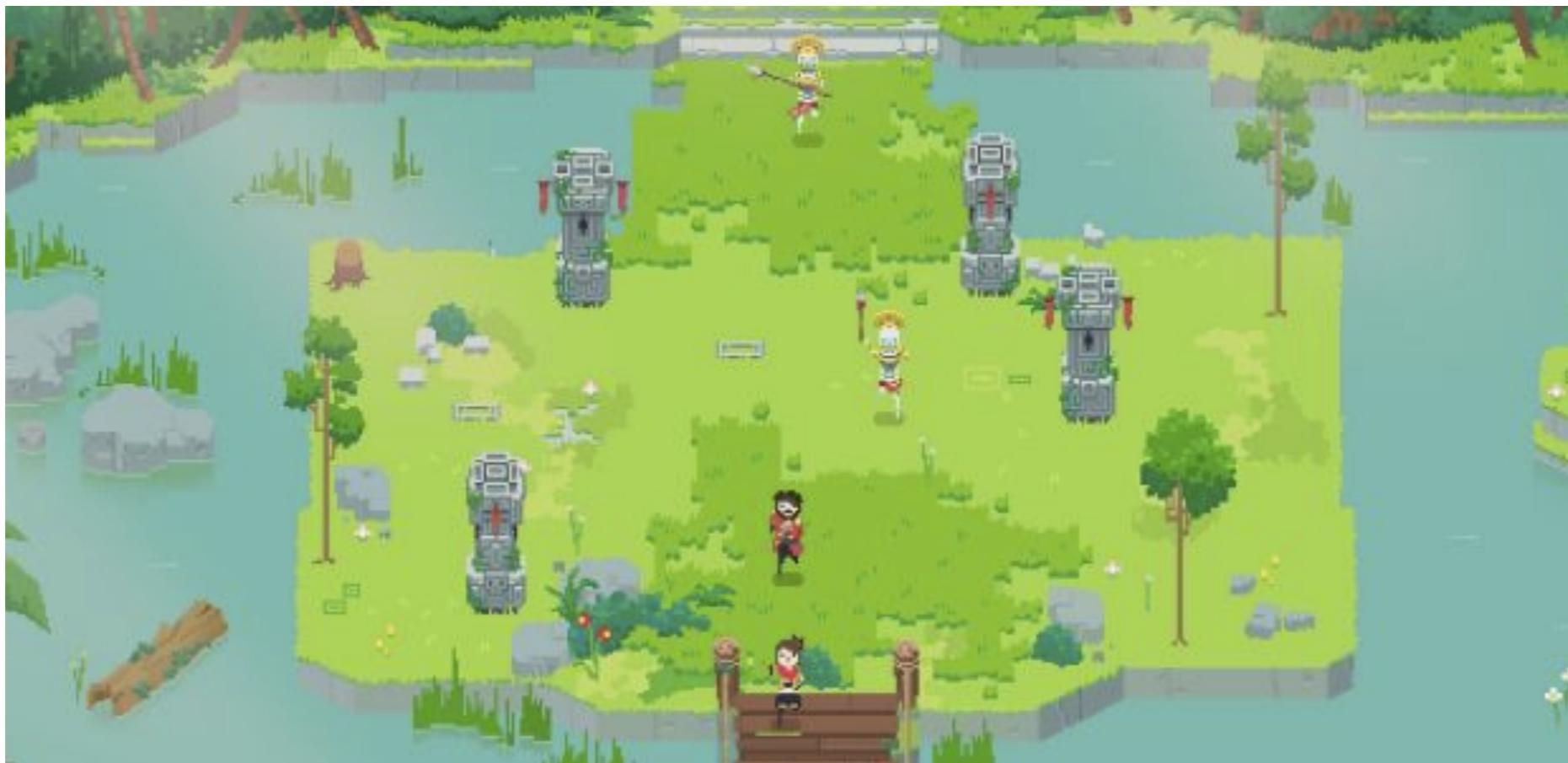
Mobile games in Haskell?

<http://keera.co.uk/blog/2014/08/13/most-inspiring-green-screen-you-will-ever-see/>

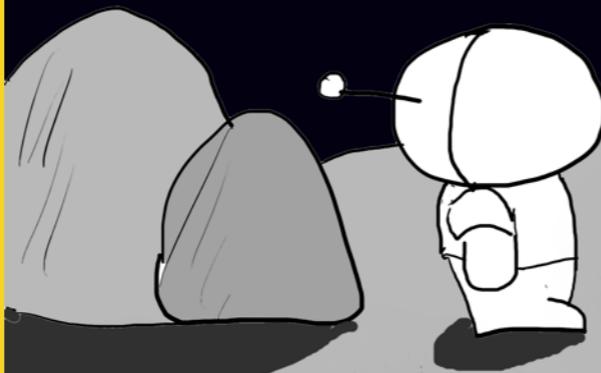


Wayward Tides

<http://blog.chucklefish.org/?p=154>



Game programming in Haskell



Elise Huard

<https://leanpub.com/gameinhaskell>
@elise_huard

Images and sounds

<http://www.freesound.org/people/Taira%20Komori/sounds/213423/>

<http://www.bbc.co.uk/programmes/articles/1g84mOsXpnNCv84GpN2PLZG/the-hitchhikers-guide-to-the-galaxy-game-30th-anniversary-edition>

<https://www.civilization.com/en/home/>

<http://mario.nintendo.com/>

<https://www.bioshockinfinite.com/>

<http://bindingofisaac.com/>