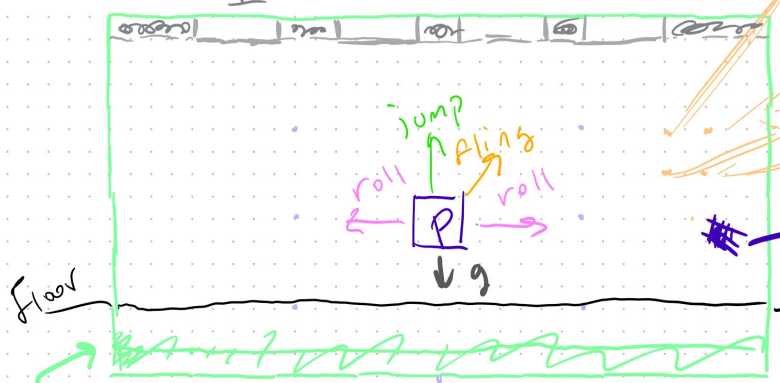
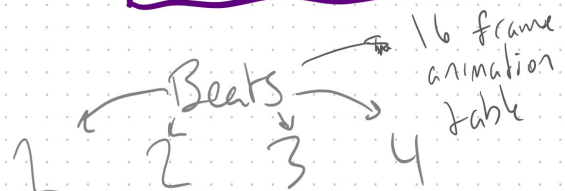


world.lua

:render()
:hud(h)
:back(h)

ng aka nsgnode horris game engine



Tiles

tile type

- stone/plank floor
- brick
- stone
- etc.

local d = { size = 64 }
d = {
 "2", "2"
 "2", "2"
}

world moves 1 tile per beat

t = { } // tile
t.data = d
t.mt = { }
on_call()

while key 3 is held or Rolling against boundaries moves world

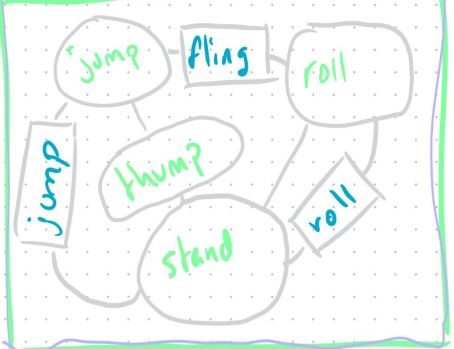
player.lua
:x() :y() => int
:xy() => vec
:busy([string]) => string

:move(
 vec,
 string)
:a_jump()
:a_stand()
:a_land()
:a_roll()

VALID -> jump, roll, nil
:move() w/ string = nil will assign one

player window is static

ANIMATION CYCLE



Frame Counters

jump

fling

roll

to eliminate if then on draw(),
globals:
px, py

pframe

for world and player draws, respectively

Summary

- interlaze + game loop must:
 - call world renderer + player renderer
 - update player x + player y
- still needed:
 - title screen
 - pause menu
 - implement timber support
 - pass timber global var + (TO BE DEFINED) table rc: triggering events (while sample, other details)
 - allow sample triggers (jump, collision)
 - allow sample loops (while walking; transist loop on + loop off)

Could I?

- buffer a backing track into softcut
- backing track is mapped by bpm to chunks each corresponding to an x coord