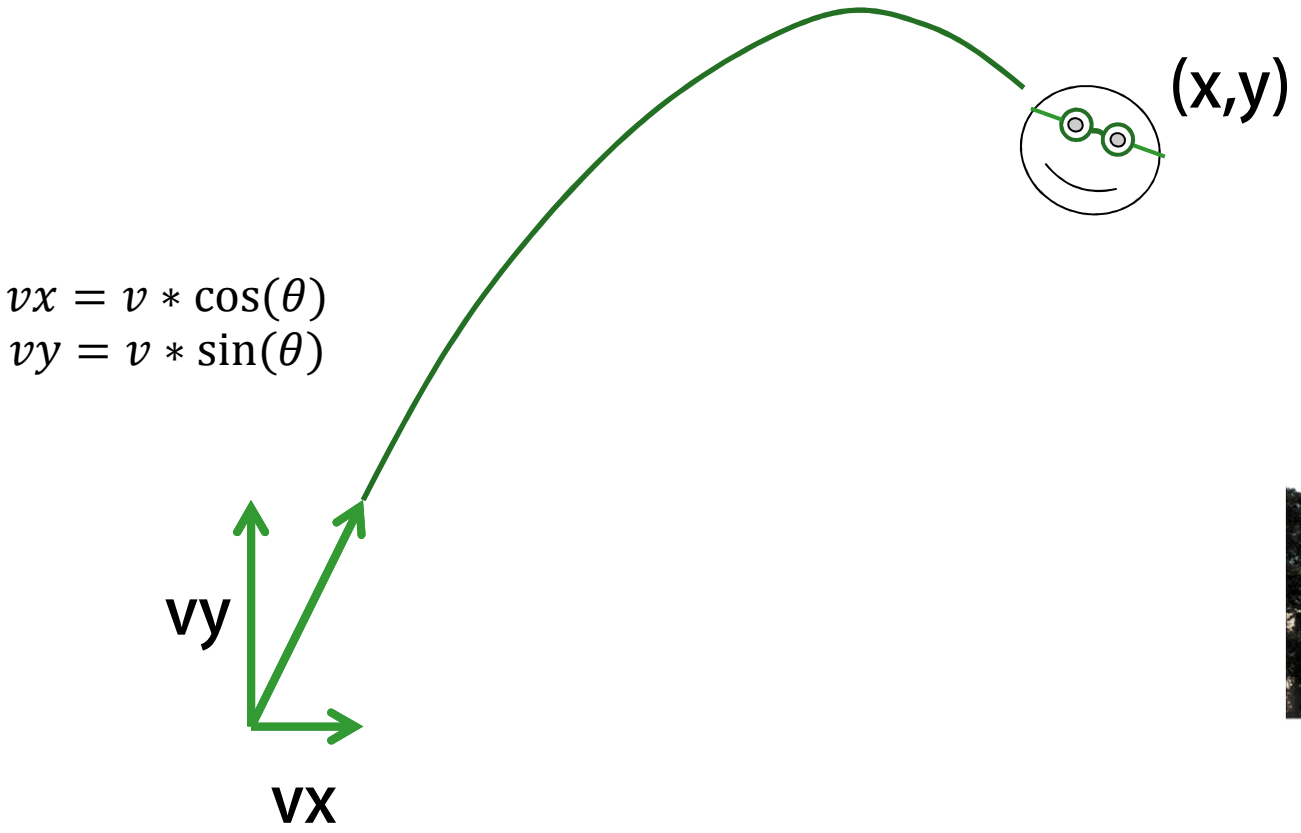
Lecture 3: More control flow, Functions, bisection methods

Armando Solar-Lezama

MIT COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LABORATORY

Angry Nerds

$$\Delta x = vx * \Delta t$$
$$\Delta y = vy * \Delta t + \frac{1}{2} g * \Delta t^2$$
$$\Delta vy = g * \Delta t$$



Code

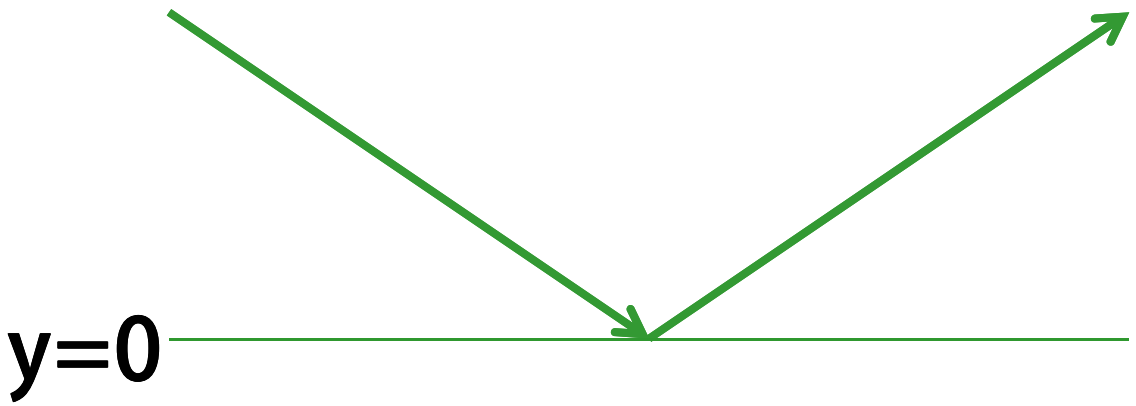
```
import math
import simpleplot as sp

g = -9.8
dt = 0.01;

x = 0.1
y = 0.1
v = 25.0
ang = 30.0
vx = v*math.cos((ang/ 180.0) * math.pi)
vy = v*math.sin((ang/ 180.0) * math.pi)

while y > 0.0:
   x = x + vx*dt
   y = y + vy*dt + g*dt*dt/2
   vy = vy + g*dt
   sp.plotTrajectory((x,y))

print x
sp.doAnimation()
```

Control flow

```
while cond :
    body
after-the-loop
```

```
if cond :
    t-body
else:
    e-body
after-the-loop
```

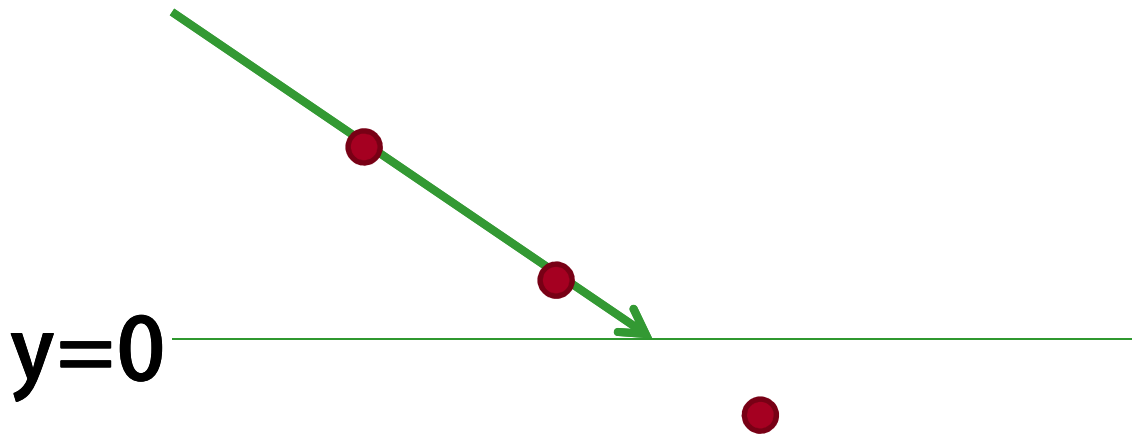
Making the nerd bounce



Almost Correct solutions

```
if y == 0.0:
    vy = -vy*0.5
```

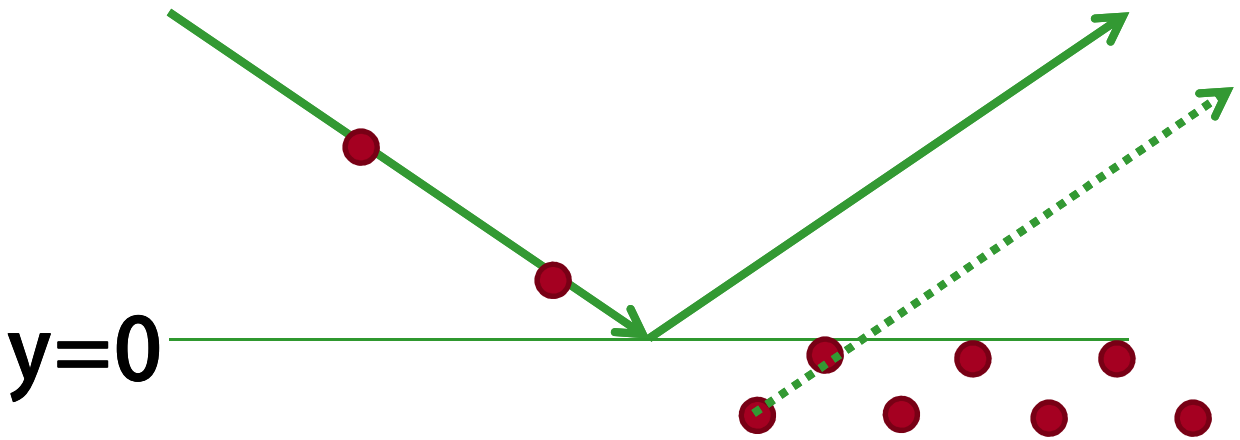
Problem: Unlikely that y will be exactly 0.0



Almost correct solutions

```
if y < 0.0:
    vy = -vy*0.5
```

Problem: After the bounce, velocity may not be enough to get past zero



Correct Solution

```
import math
import simpleplot as sp

g = -9.8
dt = 0.01;

x = 0.1
y = 0.1
v = 25.0
ang = 30.0
vx = v*math.cos((ang/ 180.0) * math.pi)
vy = v*math.sin((ang/ 180.0) * math.pi)

bounce = 0
while (bounce < 15):
   x = x + vx*dt
   y = y + vy*dt + g*dt*dt/2
   vy = vy + g*dt
   sp.plotTrajectory((x,y))
   if(y < 0.0 and vy < 0.0):
      bounce = bounce + 1
      vy = -vy*0.5
print x
sp.doAnimation()
```

Functions

Give you a way to package functionality

```
def fname(param1, param2, ...):
    body
    return value
```

Example: function for conversions

```
def degToRad(deg):
    return (deg / 180.0) * math.pi
```

```
import simpleplot as sp
import math
def degToRad(deg):
    return (deg / 180.0) * math.pi

g = -9.8
dt = 0.01

x = 0.1
y = 0.1
v = 25.0
ang = 30.0

vx = v*math.cos(degToRad(ang))
vy = v*math.sin(degToRad(ang))
```

```
bounce = 0
while (bounce < 15):
    x = x + vx*dt
    y = y + vy*dt + g*dt*dt/2
    vy = vy + g*dt
    sp.plotTrajectory((x,y))
    if(y < 0.0 and vy < 0):
        bounce = bounce + 1
        vy = -vy*0.5

sp.doAnimation()
print x
```

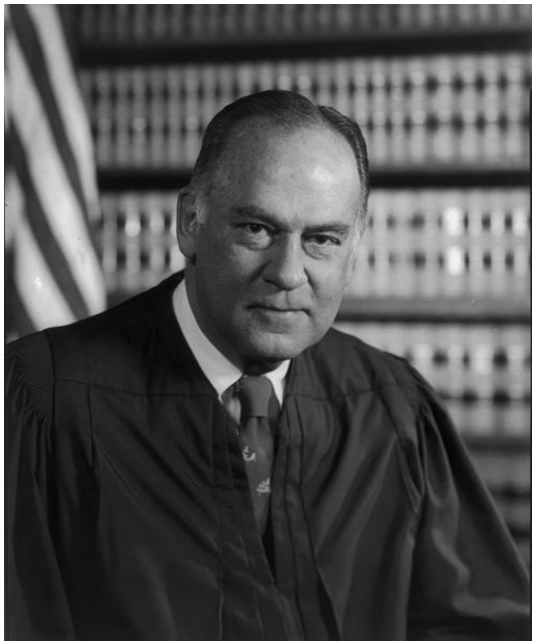
GUESS AND CHECK ALGORITHMS

I know it when I see it

I shall not today attempt further to define the kinds of material I understand to be embraced within that shorthand description ["hard-core pornography"]; and perhaps I could never succeed in intelligibly doing so. But **I know it when I see it**, and the motion picture involved in this case is not that.

—Justice Potter Stewart, *Jacobellis v. Ohio (1964)*

Source: Wikipedia.



Guess and Check Algorithms

For any problem where the answer is easy to check

```
while ( answer is not correct):
    guess a new answer
    check if it is correct
```

Very general and powerful

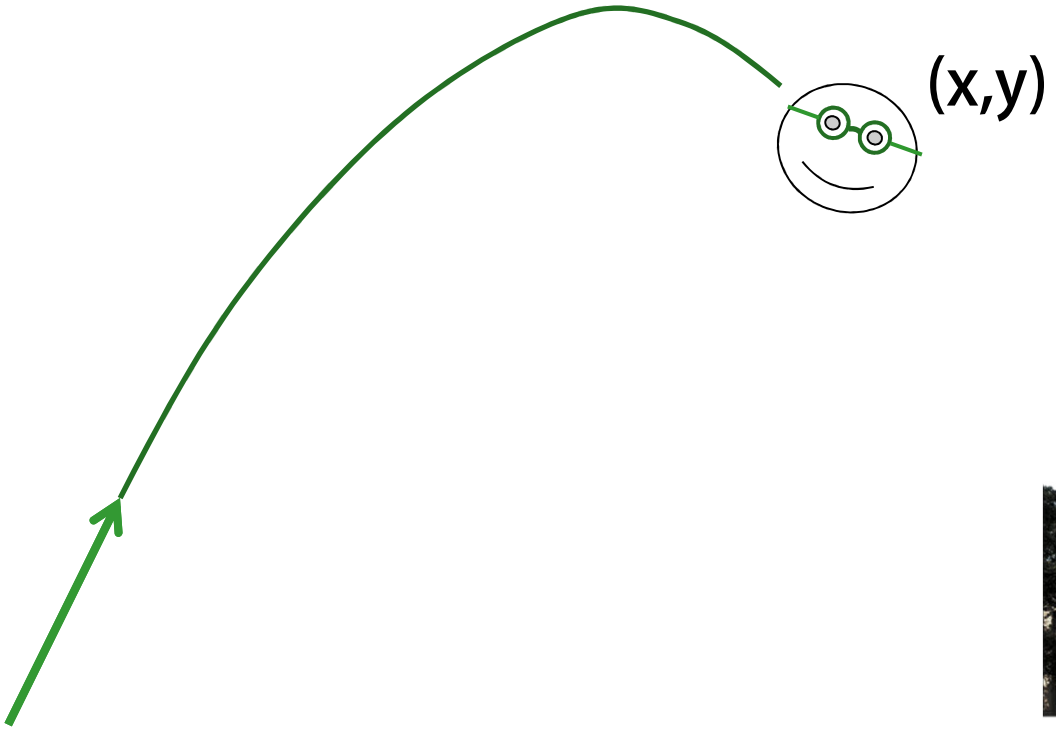
Also very slow!

- Unless you can guess answers in a smart way

Hitting the target

45^o is the most efficient angle

- But how fast should we launch to hit a target?



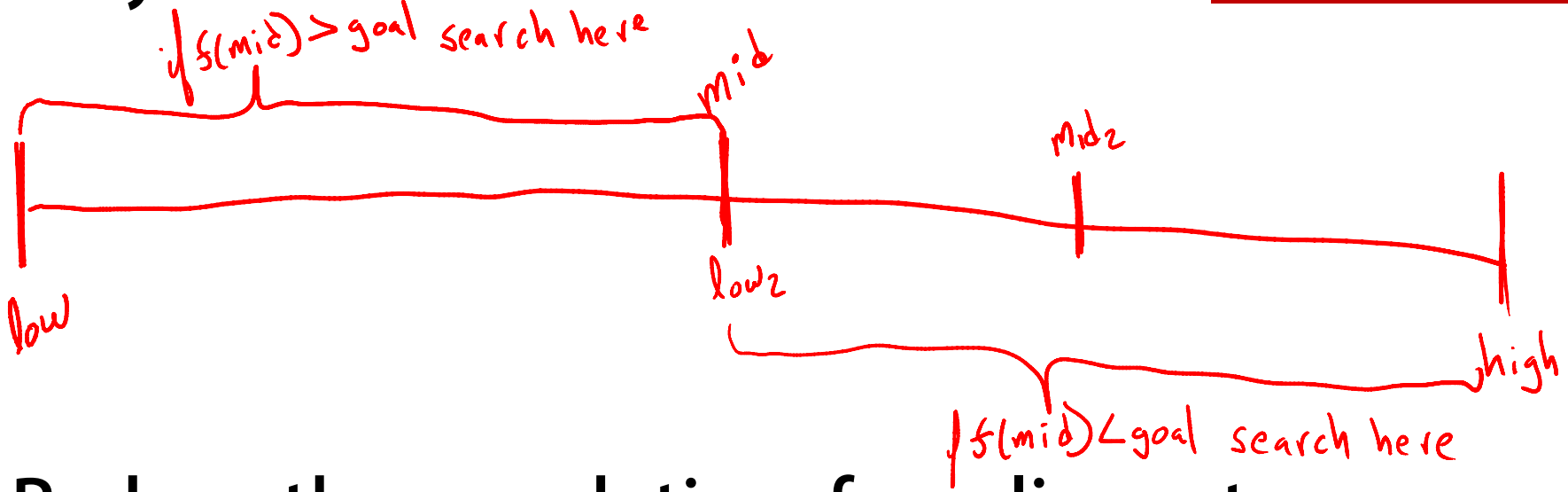
Code for hitting the target

```
goal = 100.0
testV = 0.0
land = 0.0
count = 0
while land < goal:
    testV = testV + 1.0
    land = nerdFinalPos(45, testV)
    count = count + 1

print "testV = " + str(testV) + " land=" + str(land)
print count
```

Code: http://bit.ly/12JcNcV

Bisection Search

Very effective when the function is monotonic



Reduce the search time from linear to logarithmic

Angry Nerds with bisection search

```
low = 0.0
high = 500.0

mid = (low+high)/2.0
landing = nerdFinalPos(45.0, mid)

while abs(landing - goal)>0.01:
    if(landing < goal):
        low = mid
    else:
        high = mid
    mid = (low+high)/2.0
    landing = nerdFinalPos(45.0, mid)
    print "low=" + str(low) + " h=" + str(high) + " m= " + str(mid) + " land = " + str(landing)
```

Code: http://bit.ly/VStfka