Teachers Teaching with Technology™
Professional Development from Texas Instruments

Kicking a Field Goal for the Win!

TI Webinar Series

9/20/2022

Texas Instruments

About us:

Todd Graba

T^3 Regional Instructor

STEM Division Leader

Crystal Lake, IL

tgraba@gmail.com

Twitter: @tgrabes17

Brad Posnanski

T^3 National Instructor

H.S Math and STEM Teacher

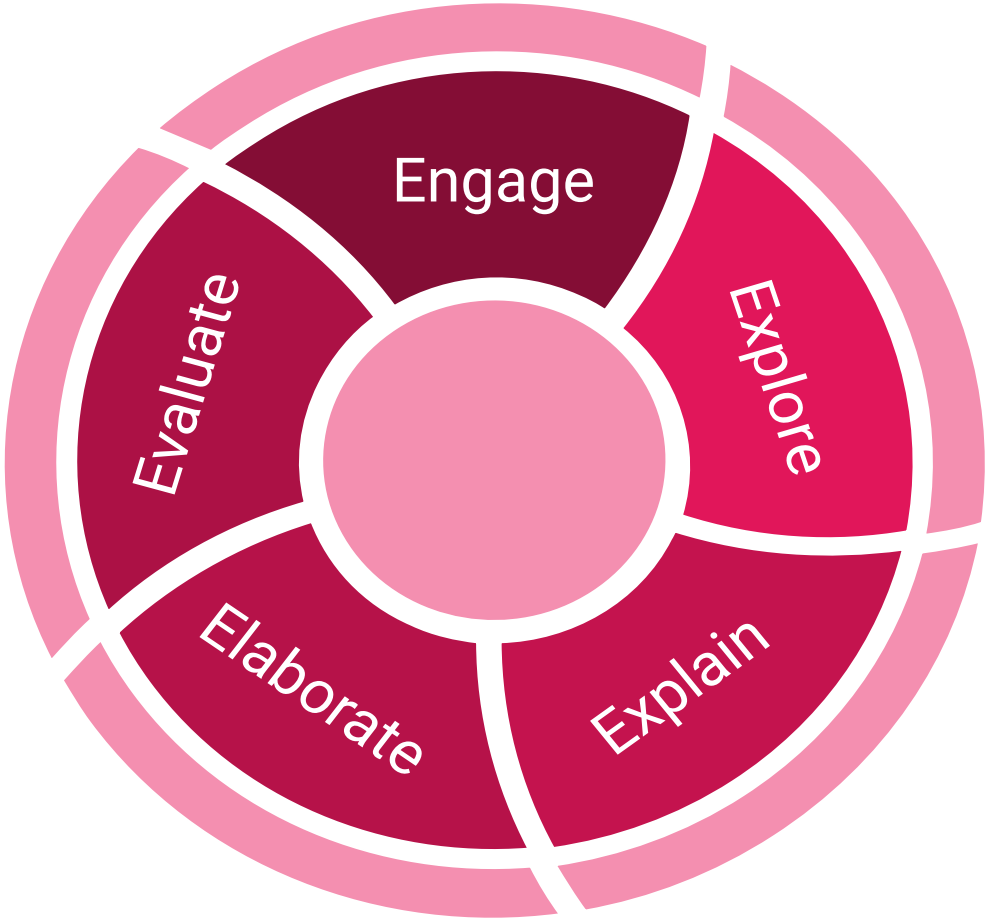
Port Jefferson Station, NY

pos22@hotmail.com

Twitter: @POSitive_MATH

Texas Instruments

5 E's Model, Cross Curricular, Multiple Entry Points



Phenomenon (Engagement)

Good or Not Good - What makes the difference on a Field Goal?

- What path does a football make when kicked off the ground and through the air?
- What kind of a function will model this behavior?
- What are the factors at impact that matter (excluding weather conditions) for a field goal kicker to be successful?
- How much does the velocity need to change for a 30 yard field goal vs. a 50 yard field goal?
- Is the speed or angle more important for success?

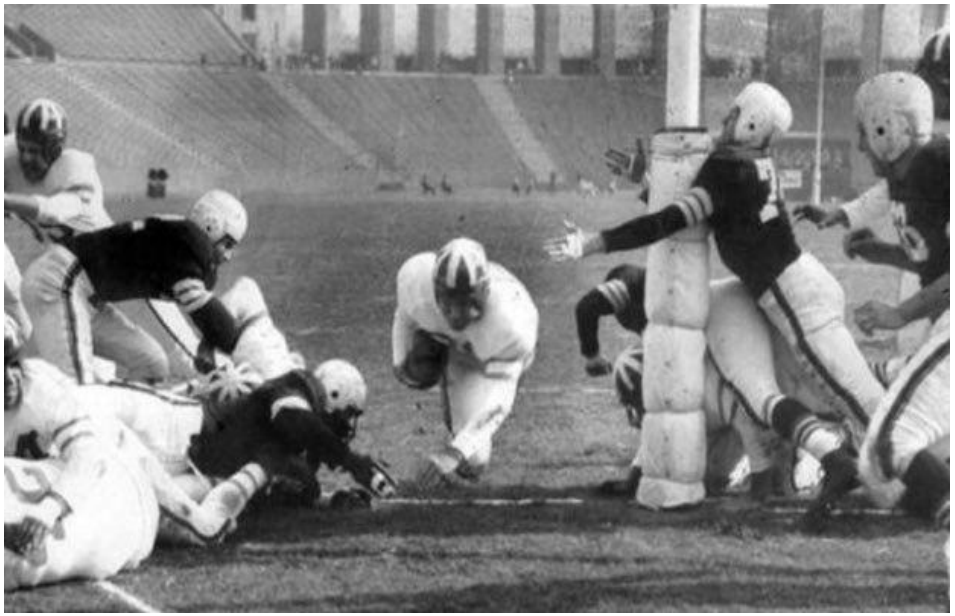
Texas Instruments

Platform 1 - 84CE (Explore)

Link to website

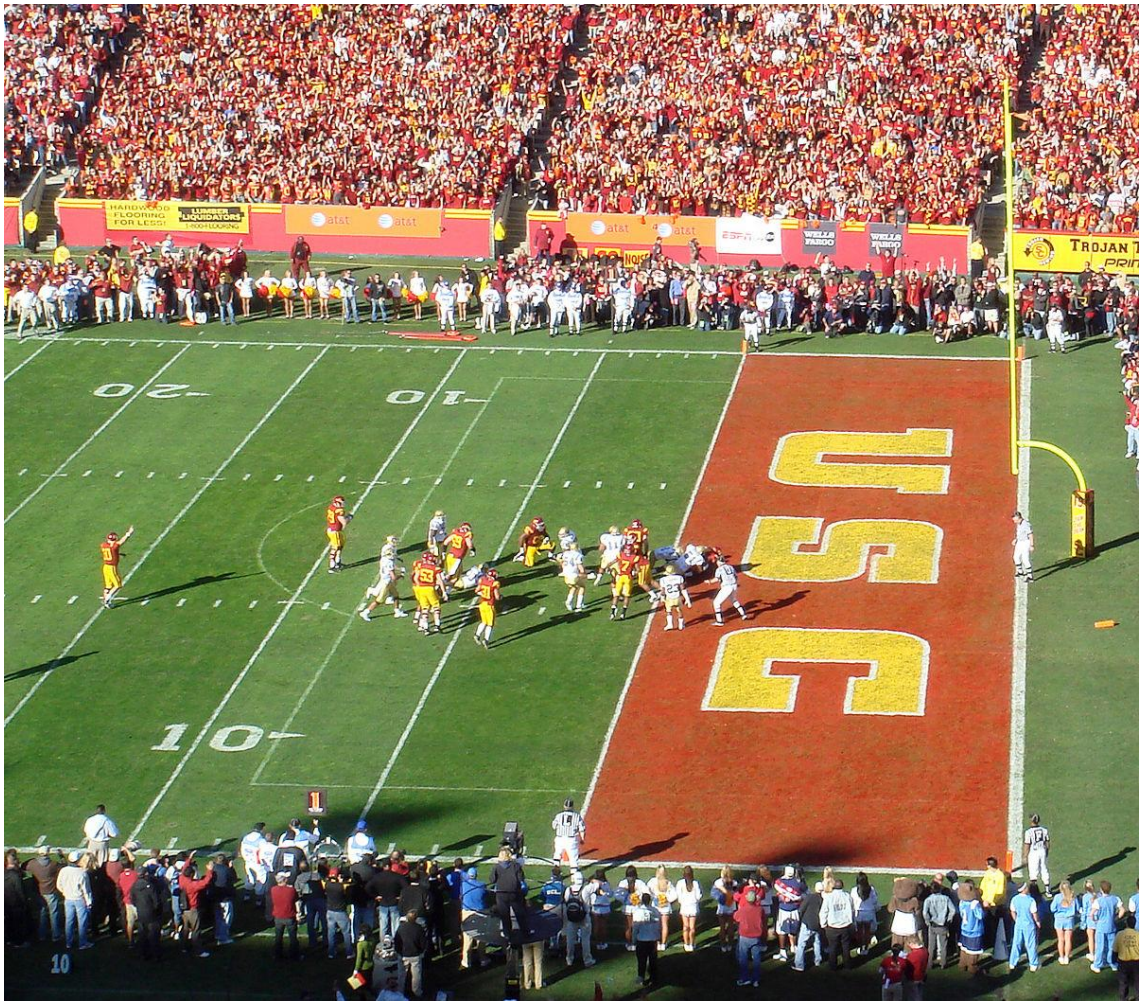
Link to webinar

Placement of the field goal posts



Texas Instruments

Placement of the field goal posts



Field Goal Distance:

= yardline
  + length of endzone
  + 7 yards (for the holder)

Longest NFL Field Goal….

How far?

66 yards

Who kicked it?

Justin Tucker

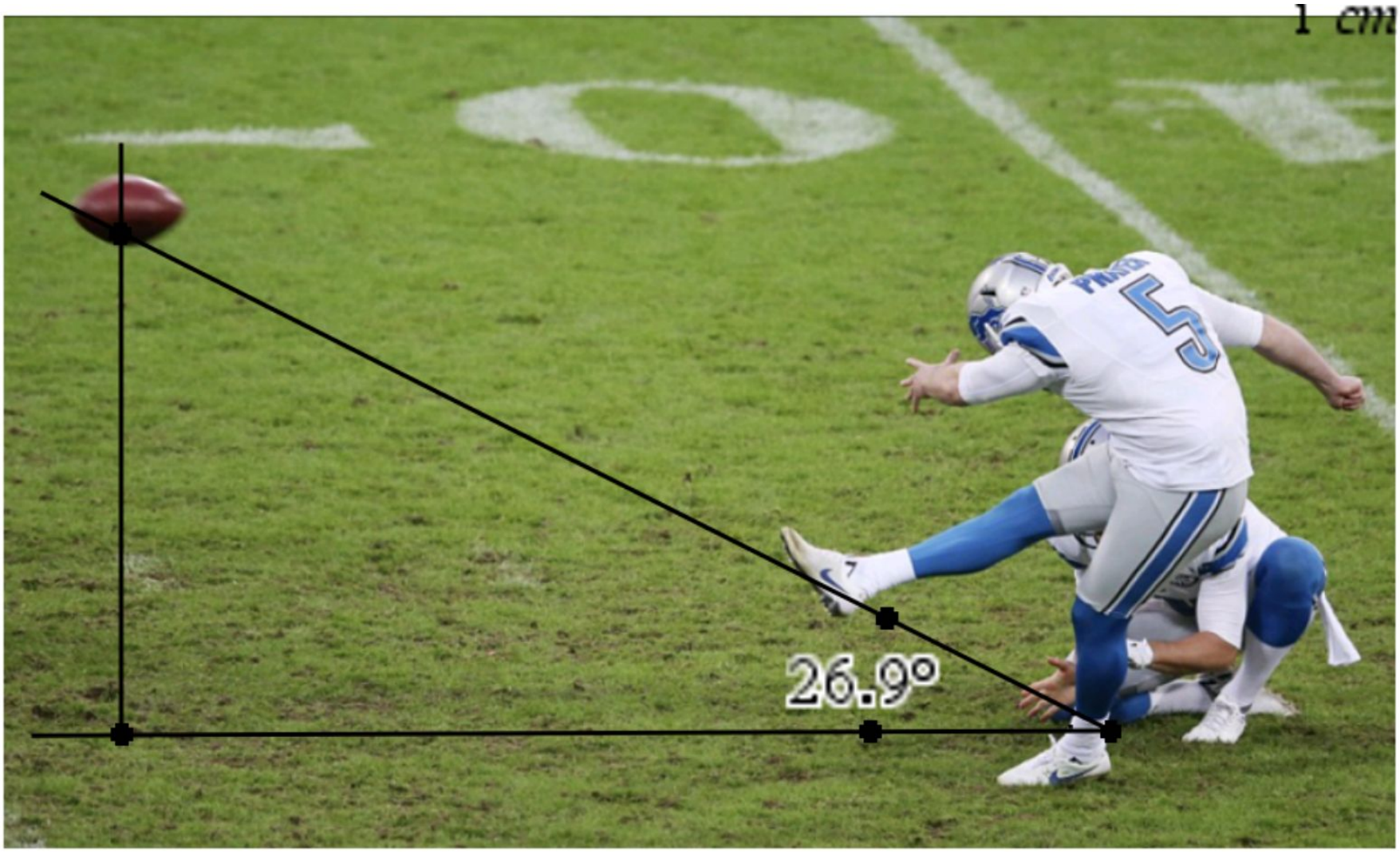
Texas Instruments

NSpire CX and CX-II

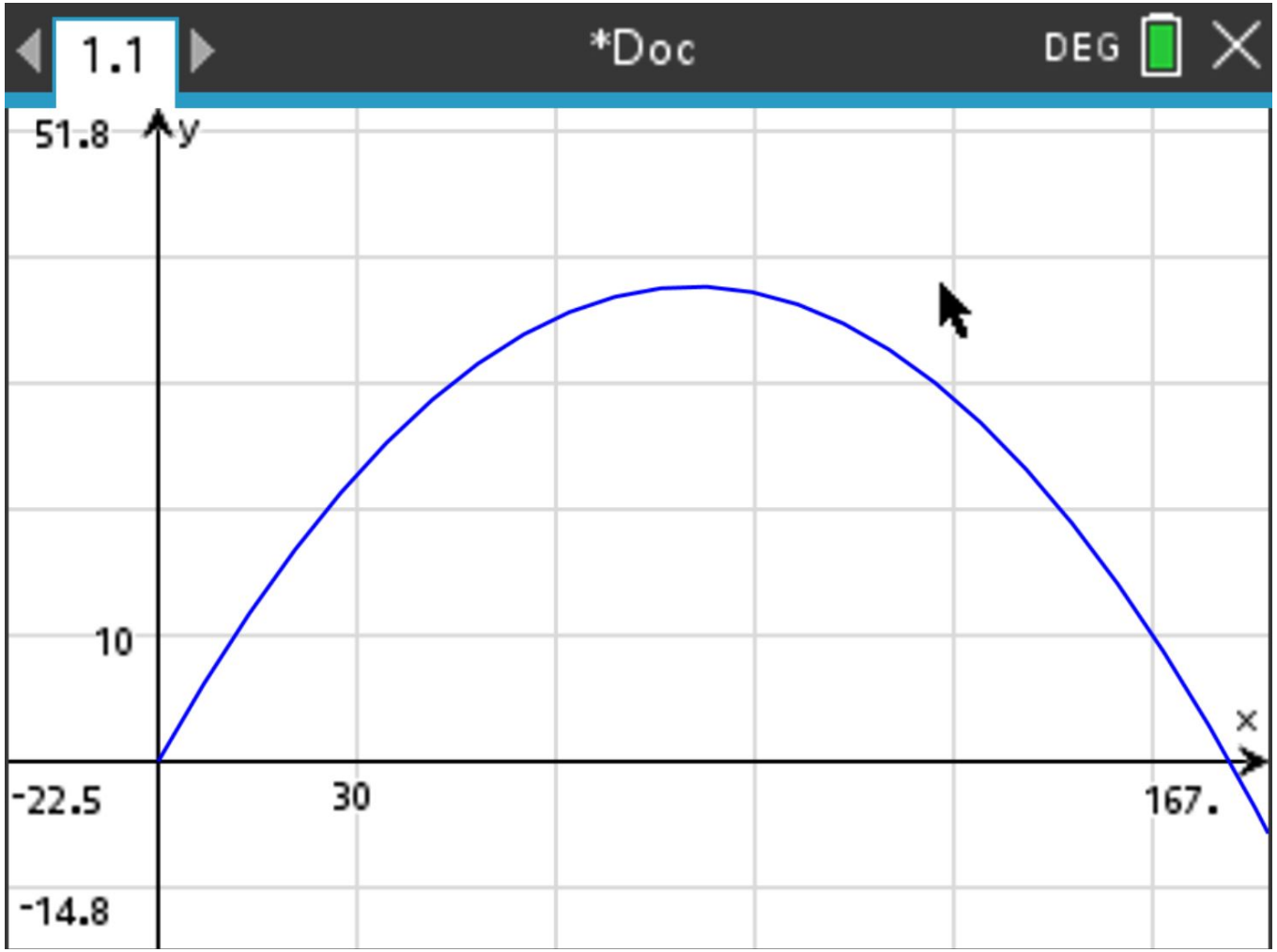
Photo credit

https://www.dailymail.co.uk/sport/nfl/article-2808472/Detroit-Lions-22-21-Atlanta-Falcons-Lions-produce-remarkable-comeback-defeat-Falcons-kick-game.html

Geometry page for angle

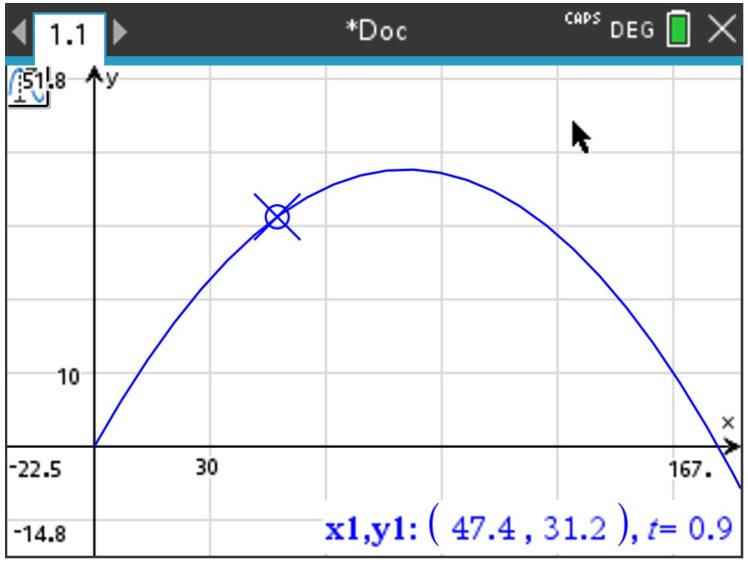


The path of the field goal: (Explain)



The path of the field goal:

Introduction to Parametric Equations:

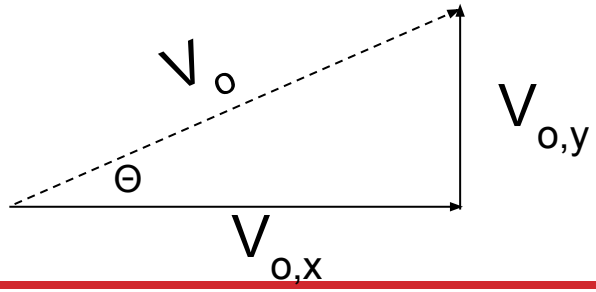


$$x(t) = V_0 \cdot \cos(A) \cdot t$$

$$y(t) = V_0 \sin(A) \cdot t - 16 \cdot t^2$$

V_0 is the initial velocity

A is the angle of the kick

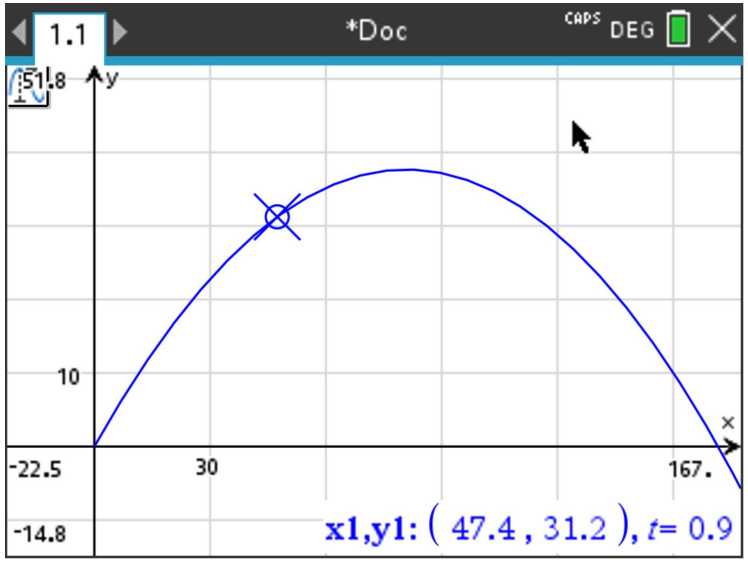


The path of the field goal:

Introduction to Parametric Equations:

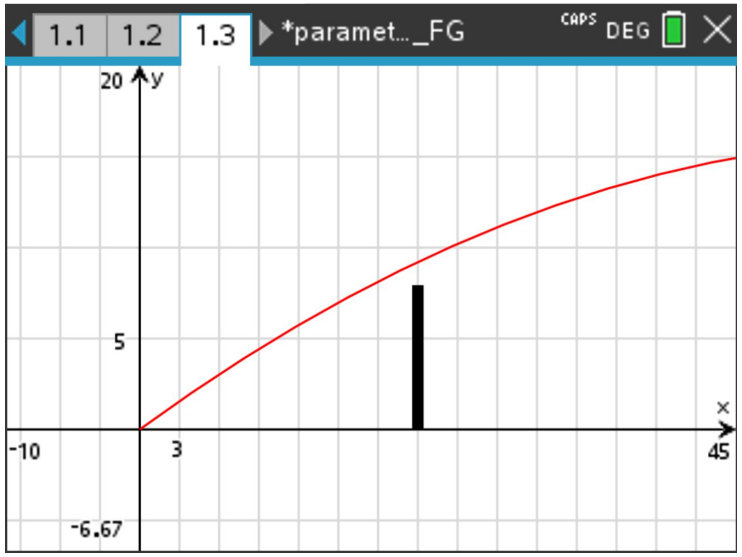
$$x(t) = V_0 \cdot \cos(A) \cdot t$$

$$y(t) = V_0 \sin(A) \cdot t - 16 \cdot t^2$$

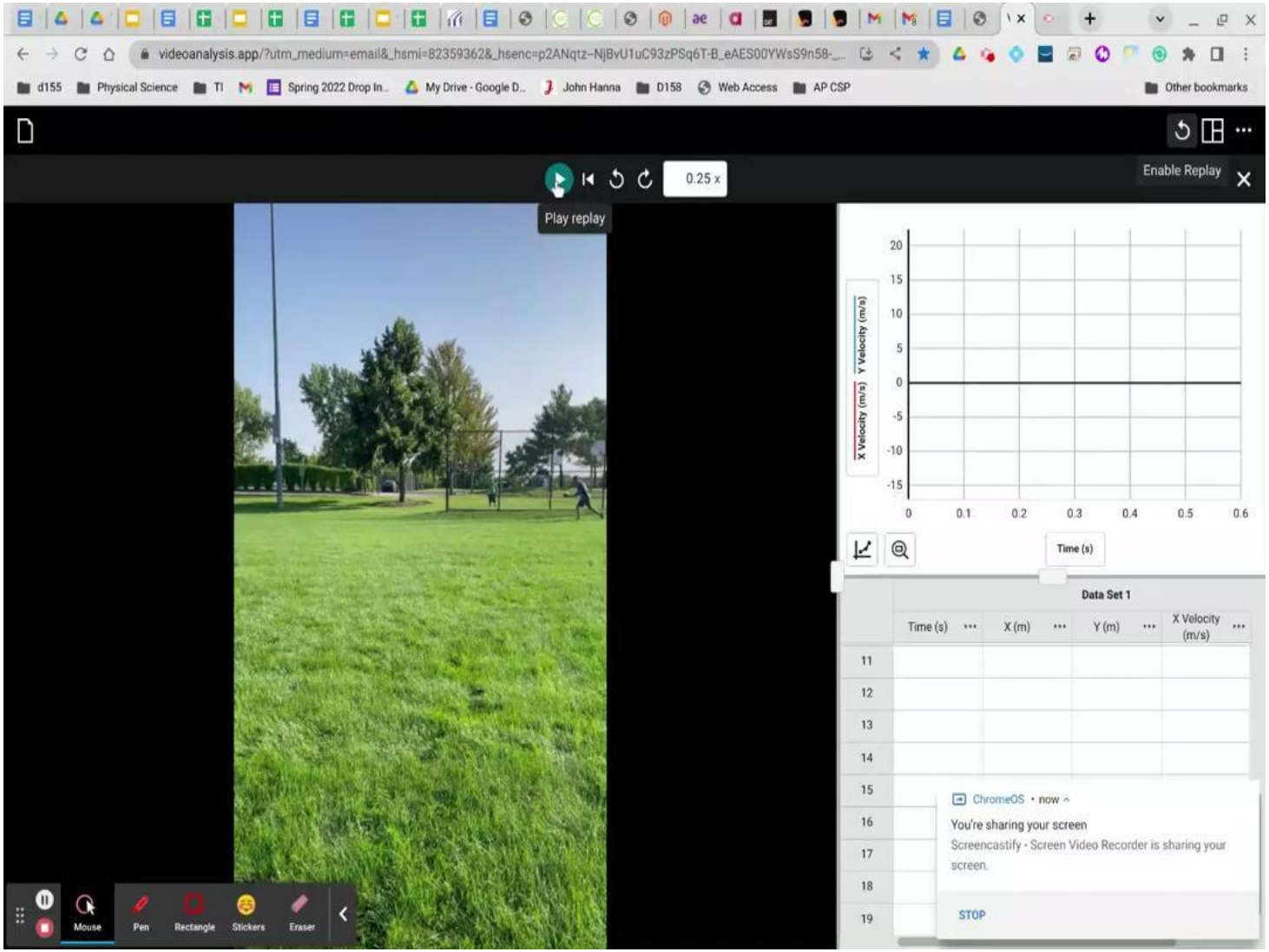


Typical kicker foot speed: 18.6 to 26.83 m/s OR 61 to 88 ft/s. -(16) value is from half of acceleration due to gravity at -32 ft/s/s
Typical kick angle: 27º to 43º

The path of the field goal:



Lab with coding - requirements Vernier Video Analysis® and Python using TI-NSpire CXII:

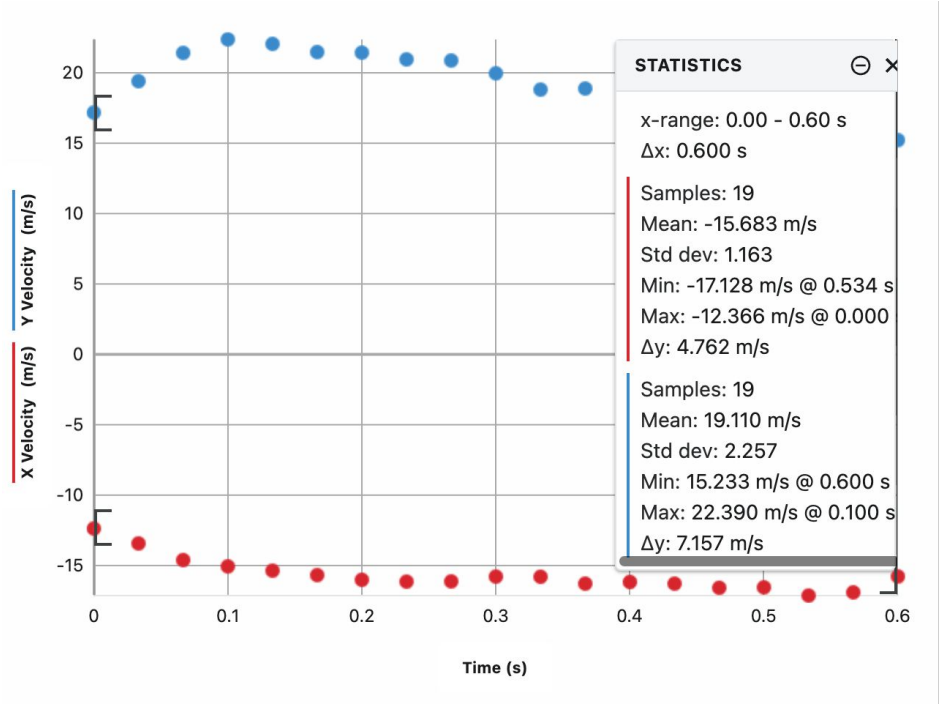


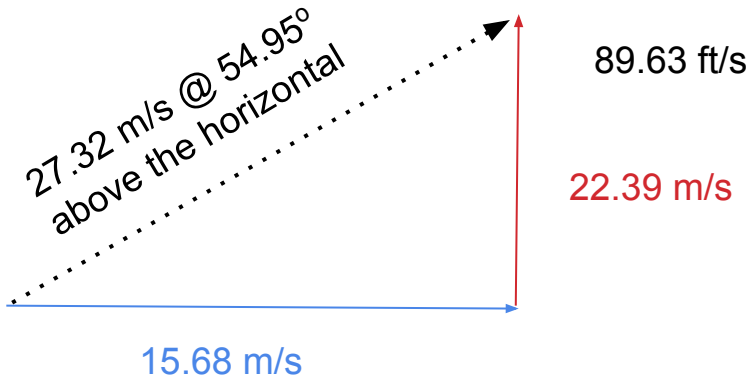
Data we need

X velocity - mean as x-velocity should not change excluding external forces

Y-velocity - max as this should be highest at launch

Calculate resultant velocity

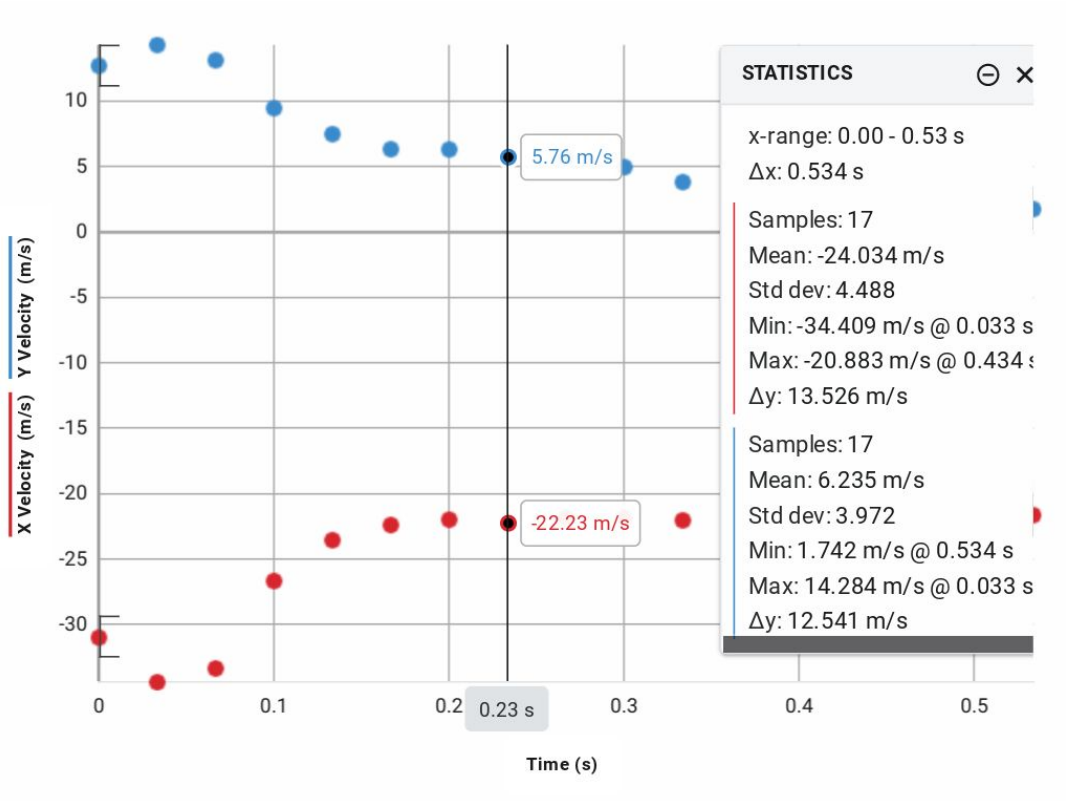


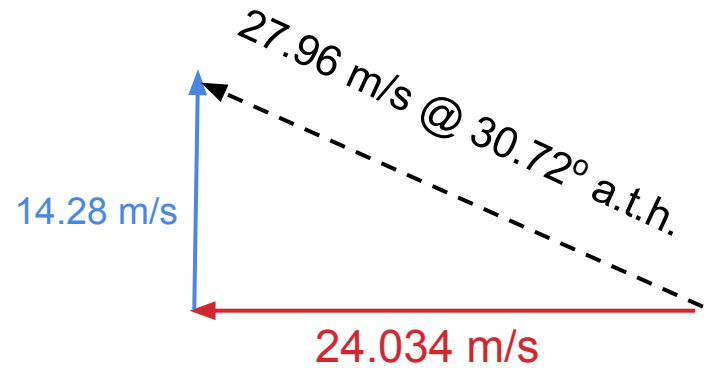


Lab with coding - requirements Vernier Video Analysis® and Python using TI-NSpire CXII:

Texas Instruments

Field Goal Data





78.61 ft/s

Simulate the Kick with Python!

*its_good.py 1/46

```
#Will model and determine if FG was good
import ti_plotlib as plt
from time import*
from math import*
from ti_system import *
clear_history()

d = int(input("How far ? (yards) "))
angle = float(input("angle ? (degrees) "))
velo = float(input("velo ? (ft/sec) "))

d=d*3 #converts yards to feet
angle_rad = angle*pi/180
#playing field as a function of field goal dist
plt.window(0,d+30,0,60)
plt.grid(30,10,"dashed")

#goal post that moves to the set distance
plt.color(250,250,0)
plt.pen("thick","solid")
plt.line(d+6,0,d+6,10,"default")
plt.line(d,10,d+6,10,"default")
plt.line(d,10,d,30,"default")
plt.color(0,154,23)
plt.line(0,1,d+30,1,"default")
plt.color(137,87,54)
```

User inputs

distance and angle conversion

creating the window and scale

Creating the goal posts

Simulate the Kick with Python!

*its_good.py 53/53

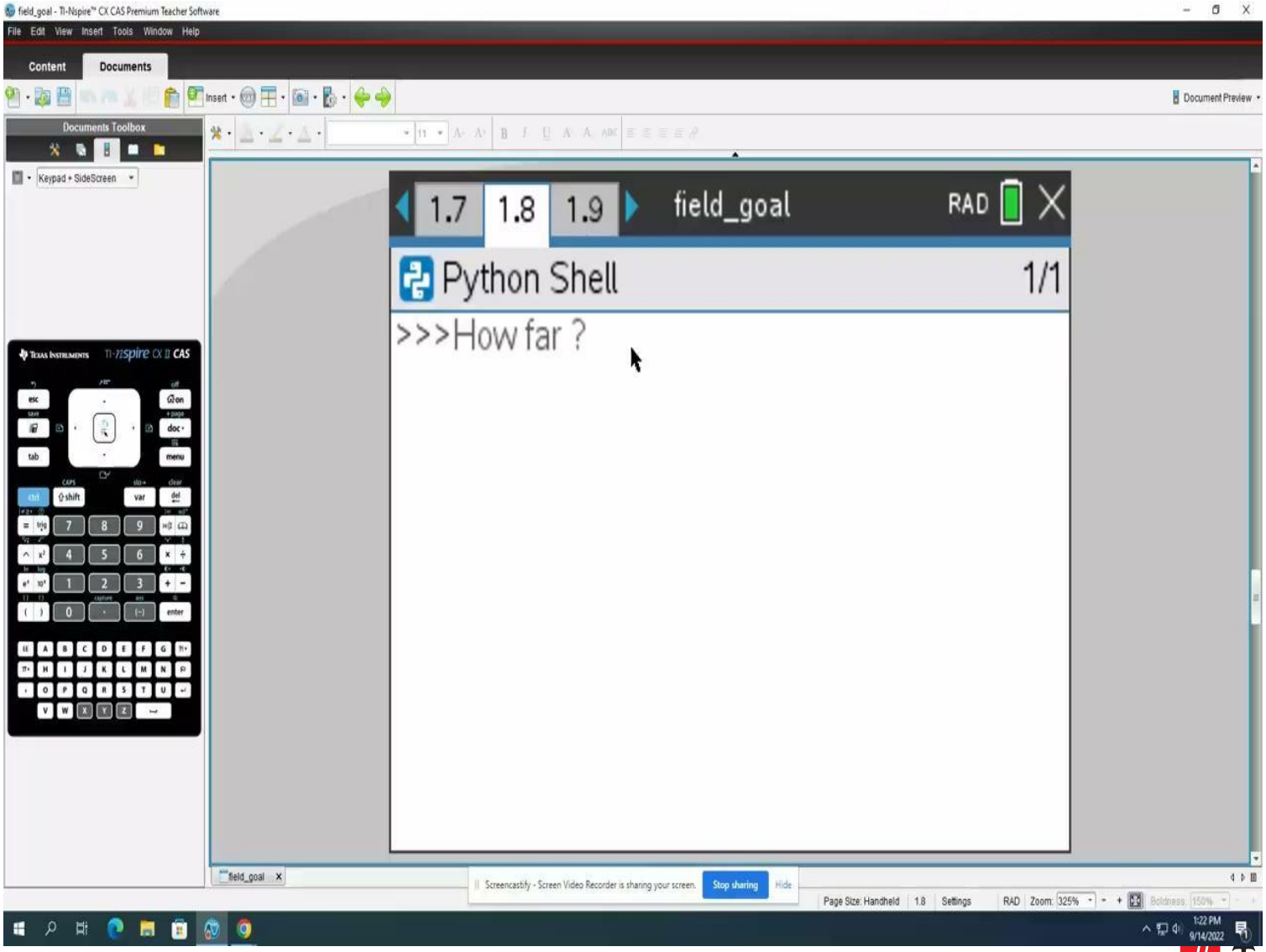
```
t=0 #time
for i in range(0,60):
    a = velo*cos(angle_rad)*t
    b = velo*sin(angle_rad)*t-16*t**2
    t+=.05
    plt.plot(a,b,"o")
    sleep(.1)

#determine and display result
t=float(d/(velo*cos(angle_rad)))
y = float(-16*(t)**2+velo*t*sin(angle_rad))

plt.color(255,0,54) #color of text
if y>10:
    plt.text_at(3,"IT'S GOOD!!","center")
else:
    plt.text_at(3,"NO GOOD  :-(","center")
```

Plots the path of the football using the parametric equations. Time is increasing .05 seconds / plot

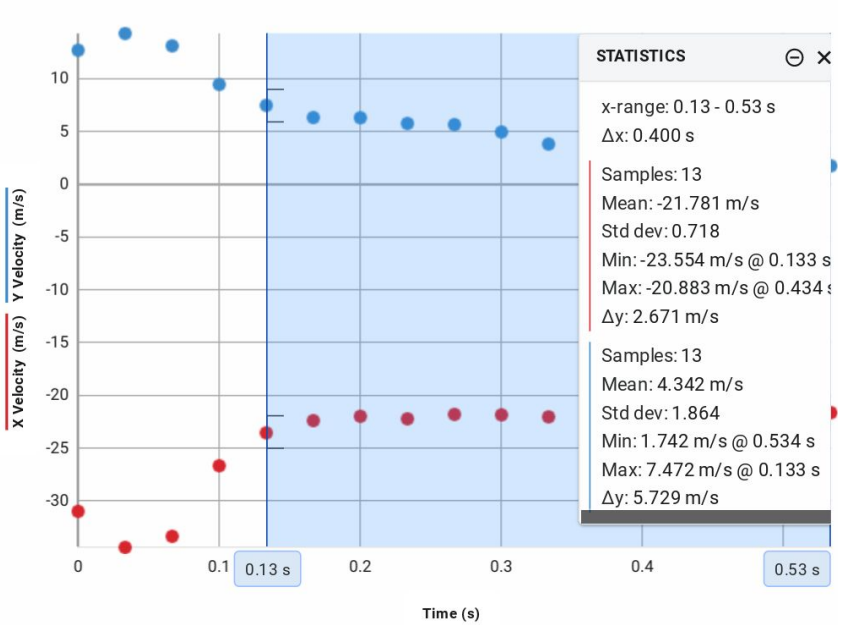
Calculates the height of the football when it reaches the length of the attempt

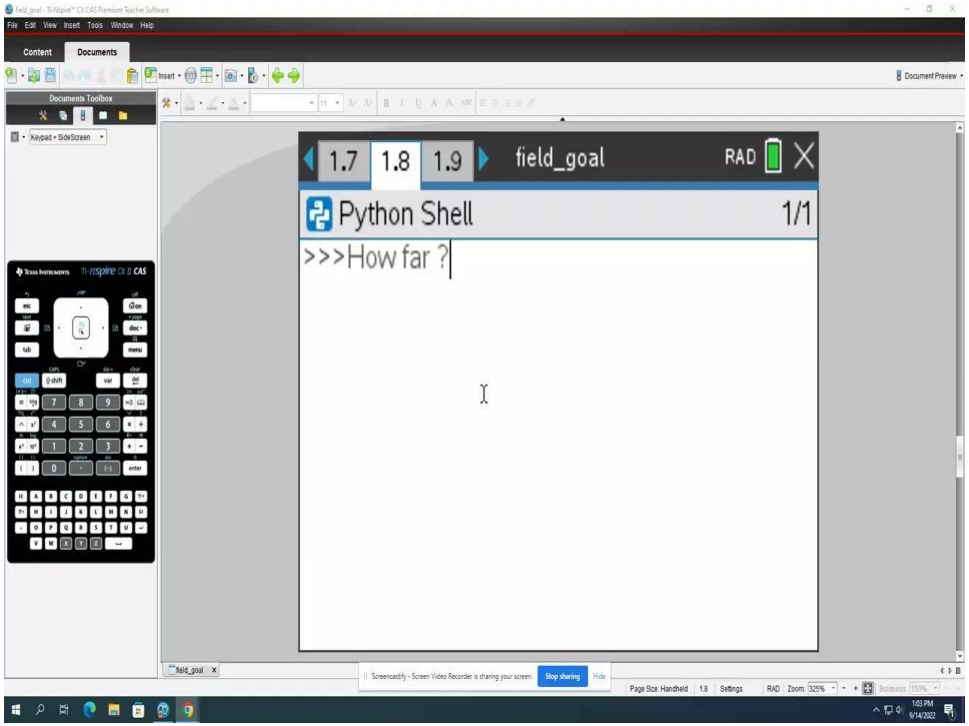
Displays the result of the kick

The Kick is UP… (Elaborate)

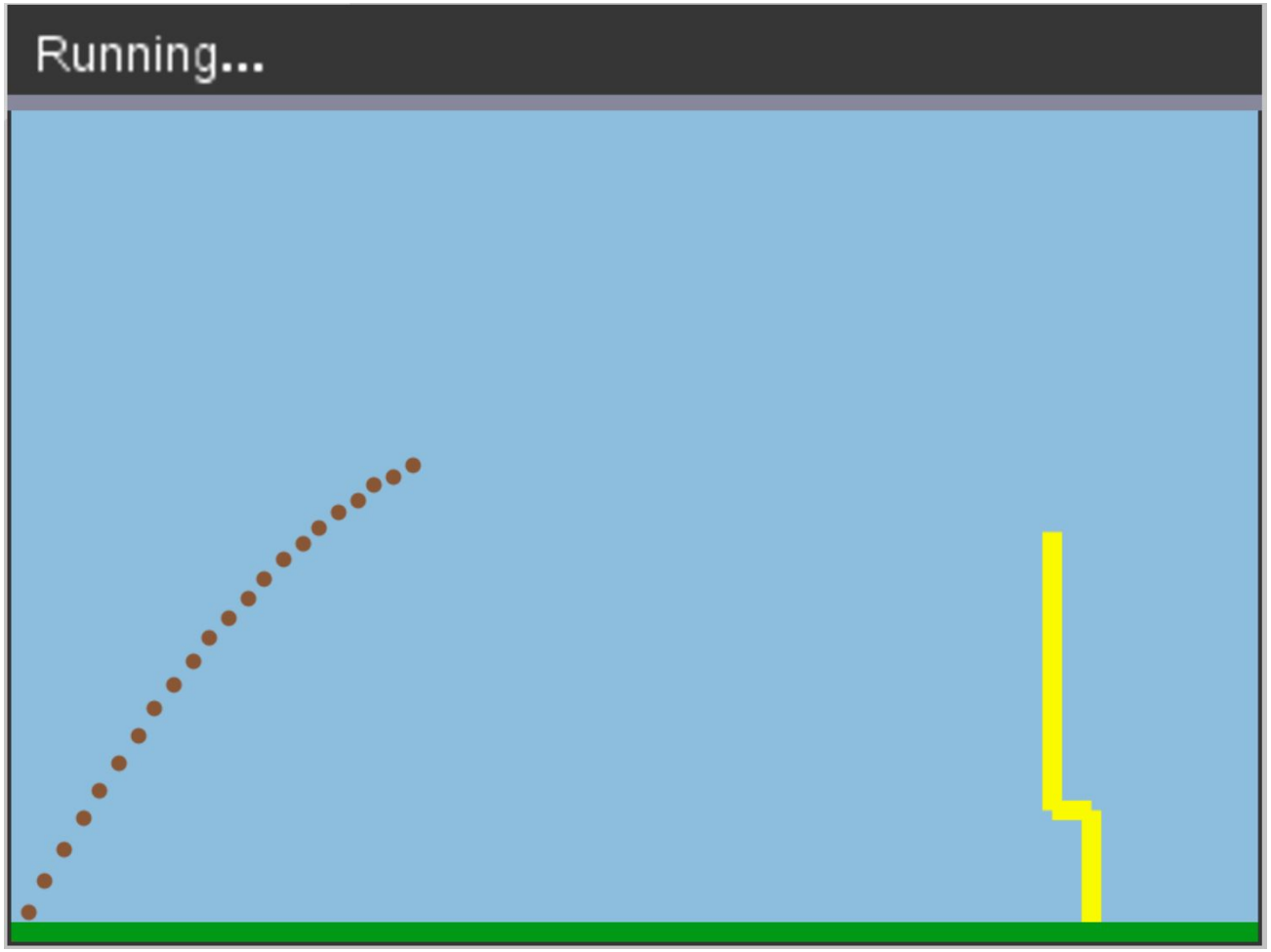


Or really…(Evaluate)





Program development & Participant Field Goal Challenge



About us:

Todd Graba

T^3 Regional Instructor

STEM Division Leader

Crystal Lake, IL

tgraba@gmail.com

Twitter: @tgrabes17

Brad Posnanski

T^3 National Instructor

H.S Math and STEM Teacher

Port Jefferson Station, NY

pos22@hotmail.com

Twitter: @POSitive_MATH

All content from tonight's webinar will be up On Demand on education.ti.com within 48 hours

Texas Instruments

Getting Started
Mood Ring
with TI Nspire CX II

```
from ti_hub import*
```

RED GREEN BLUE

- → LED

```
color.rgb(0,0,255)
```

- → Output

```
text_at(8,"Dreamy",”left”)
```

- → Connecting Temperature Sensor

```
my_temp=temperature(“IN 1”)
t=my_temp.measurement()
text_at(6,”Temperature=“,+str(t)+“℃”,”left”)
```

Defining your digital mood

Temperature level definitions	Mood Message	Color Name or Description	Color RGB Code	Other Output (Optional) Sound? Vibration?...

Temperature level definitions	Mood Message	Color Name or Description	Color RGB Code	Other Output (Optional) Sound? Vibration?...
t<24	"You are Stressed"	off	0 0 0	
t≥24 and t<26	"You are Nervous"	red	255 0 0	
t≥26	"You are Calm"	magenta	255 0 255	

Digital Mood Ring for TI-84 Plus CE Simple Example Program

Setup

Inputs

While Loop

Outputs

Clean up

```
from ti_hub import *
import color
from temperat import *
from ti_system import *

temp=temperature("IN 1")

while not escape():
**t=temperature("IN 1")
**disp_at(6,"temp= ", "left")
**print(t)

**if t<20:
****disp_at(8,"happy",left")
****color.rgb(250,0,0)
**if t>=20 and t<24:
****disp_at(8,"sad",left")
****color.rgb(0,250,0)
**if t,25:
****disp_at(8,"meh",left")
****color.rgb(0,0,250)


sleep(1)
```


INSTRUMENTS

Digital Mood Ring for TI Nspire CX II Simple Example Program

Setup

Inputs

While Loop

Outputs

```
from ti_hub import *

my_temp=temperature("IN 1")

while get_key() != "esc":
**t = my_temp.measurement()
**text_at(5,"Temperature = " + str(t)+"°C","left")

**if t<20:
****text_at(8,"I feel happy","left")
****color.rgb(255,0,0)
**if t>=20 and t<22:
****text_at(8,"I feel hopeful","left")
****color.rgb(0,255,0)
**if t>=22 and t<24:
****text_at(8,"I feel meh","left")
****color.rgb(0,0,255)
**if t>=24 and t<26:
****text_at(8,"I feel calm","left")
****color.rgb(255,255,0)
**if t>=26:
****text_at(8,"I feel chill","left")
****color.rgb(255,0,255)
**sleep(1)
```


INSTRUMENTS