

Code Poetry Slam 1.1  
Stanford University  
February 27<sup>th</sup>, 2014,  
7- 9 pm  
Wallenberg Hall

# Me and You

Matthew Lee

Surrogate: Myles Borins

```
import random
```

```
traits = ["only speak when spoken to", "never know how to  
introduce myself", "hide in the middle of a group", "lead in the  
front", "make my opinions known", "do all of my homework", "have  
a plan", "know what will work and won't", "believe in true  
love", "get caught in the past", "know the future will be  
better", "want to please", "set goals", "smile at strangers",  
"watch my feet when walking", "look for things to do", "let  
things unfold", "climb trees", "keep my feet on the ground",  
"walk fast", "drive slow", "read the newspaper", "watch  
cartoons", "eat breakfast", "believe in the midnight snack",  
"snore", "yawn loudly", "love vegetables", "eat gluten-free",  
"get cold easily", "need a hug", "get lost in eyes", "hate my  
reflection", "sing in the shower", "give others the benefit of  
the doubt", "double-check the lock on the door", "watch the  
sunrise", "floss", "hide my scars", "listen to Chopin", "keep a  
diary", "scrapbook", "knit", "sweat a lot", "wash your hands  
before eating", "pray"]
```

```
def Me():
    my_trait = traits[random.randint(0, len(traits)-1)]
    print "I " + my_trait + "."
    return my_trait

def You():
    your_trait = traits[random.randint(0, len(traits)-1)]
    your_trait = your_trait.replace("my", "your")
    your_trait = your_trait.replace("am", "are")
    print "You " + your_trait + "."
    return your_trait

def WillItWork(me, you):
    print "Will it work?"
    willitwork = random.choice(["Yes", "No"])
    print willitwork + "."
    if willitwork == "No":
        print "Will it be okay?"
        print random.choice(["Yes", "No"]) + "."

me = Me()
you = You()

WillItWork(me, you)
```

# The Tell

Jeff Thompson

Surrogate: Gráinne Watson

The tell.

The the the that thou that make, make and, might bear might might heir  
heir, tender heir heir might say stop.

Stop stop.

Posterity stop beauty's beguile churl tender churl mak'st waste.

Mak'st warm warm lies abundance.

Where thy.

See.

And content content thy buriest, buriest thy content and old art  
buriest.

Art old.

Thy not renewest thou dost beguile.

Beguile dost thou in trenches trenches renewest trenches deep deep his  
self lusty days lusty self his deep deep dig of of of of of of treasure  
tomb of his deep not not thou thou not thou not thy old and see famine  
where abundance blood blood thy thy blood churl mak'st the beguile the  
mak'st waste world thy thou thy thou thou posterity, stop beauty's  
beguile beguile churl tender churl blood churl.

Blood blood.

Abundance, lies couldst but.

Couldst couldst answer couldst couldst.

Thou couldst answer answer as as lies fond will will be will where lies  
beauty but die thy, beauty lies as the as lies beauty, but couldst lies  
lies thy.

When waste world thy field field.

Posterity stop.

Stop posterity within say stop posterity field thy thy world  
world unbless in unbless.

Unbless youth's thy field the beguile the field thy.

Thou art own own deep deep, sunken.

Thou proving his beauty his proving world's proving his  
contracted eyes contracted his.

Beauty.

Beauty ornament and only and only that only herald thine  
succession.

Succession thine this were the to to this.

This.

To the to herald face thee thee and thee.

Thee face, should should face face herald herald to to this  
this.

This eyes shame eyes this.

Thine this this were, were to thy light's light's thy to gaudy  
the to the to.

Should when forty form another winters.

Shall winters shall besiege.

have you heard my  
startup pitch

Nathaniel Smith

Surrogate: Elena Dancu

they walked together by habit  
in only filthy coveralls  
with little more sense than a rabbit  
speaking of defunct protocols.

the stuff began to twist and writhe,  
spilling over and draining out its side.  
this was a most illicit enterprise  
like netsites in roma provide.

they flung it into the gulf:  
a pillow upholstered in scalp.  
hands stained like the muzzle of a wolf,  
they resumed kicking along the whelp.

like businessmen draped in the pelt of an art nerd  
they turned their backs on this poisoned world.

# small talk

Jonah Galeota-Sprung

Surrogate: Michael  
St.Clair

```
from __future__ import print_function
from time import sleep
import string
from random import uniform
import sys
```

```
def process_input(in_string):
```

```
    in_string = in_string.translate(None, '!.?')
    words = in_string.lower().split()
    words.sort(key=len, reverse=True)
    return words
```

```
def wait(n): #sleep() was misbehaving
    k = 0
    for i in range(1,1000000*n):
        k = i+1
    return
```

```
def printT(in_string):
    for c in in_string:
        print(c,end='')
        sys.stdout.flush()
        sleep(uniform(0,0.1))
    return
```

```
def body():
#lines
    line1 = "\n\ni can't tell anymore "
    line2 = "what's parody and what isn't. "; line2_1 = "this song"

    line3 = "                --the clock,"
    line4 = "                again--"

#user text
    line5= "no, sleeping doesn't work." ;line5_1 = " and"
    line6 = "she's sick to death of numbers"
    line7 = "anyway. \n"

    line8 = "\nwhen would you start\nto worry?"

    line9 = "                --shivers, "
    line10 = "                wood--"

    line11 = "the cousins are closing in "
    line12 = "like an ellipse. none of this"
    line13 = "reads both ways, neither of us"
    line14 = "does."

    line15 = "                --soon,"
    line16 = "                a home--"

    line17 = "you're right,"
    line18 = "i'm out of my league. \nin over my head."

    line19 = "but there never were many houses"
    line20 = "to speak of, \nwere there?"

    line21 = "what could we have done, really."
    line22 = "what could we do."
```

```
#script
printT(line1)
print()
printT(line2)
wait(5)
printT(line2_1)
print()
wait(1)
print('\n'+ '\033[1m'+ line3)
print(line4 +'\033[0m')
in1 = raw_input(' ')
printT('\n' + line5); wait(2); print (line5_1); wait(1)
printT(line6)
print()
wait(1)
printT(line7)
wait(2)
printT(line8)
print()
wait(2)
in2 = raw_input(' ')
in3 = raw_input(' ')
in4 = raw_input(' ')
utext1 = process_input(in2)
utext2 = process_input(in3)
```

```

print()
print('\n'+ '\033[1m'+ line9)
print(line10 +'\033[0m'+'\n')
printT(utext1[0]+'? '+utext2[0]+'? '+ line11+'\n')
printT(line12+'\n')
printT(line13+'\n')
printT(line14+'\n')
in5 = raw_input('                                ')
in6 = raw_input('                                ')
print()
printT(line17+'\n')
printT(line18+'\n')
print('\n'+ '\033[1m'+ line15)
print(line16 +'\033[0m'+'\n')
in7 = raw_input('                                ')
print()
wait(2)
printT(in7+'. . . \n')
printT('\n'+line19+'\n')
printT(line20+'\n')
in8 = raw_input('                                ')
in9 = raw_input('                                ')
print()
printT(line21+'\n')
wait(5)
printT(line22)
wait(5)

```

```

#run
body()

```

If Nothing Else

Sandra Trinh  
video

Private Conversation

Declare I As “melodramatic heroine”

Declare You As “almost villain”

Declare Setting As “our shared bed”

If I = a few glasses of whiskey in

    You = disgruntled person

    You.Argument(myDrinking)

    Grievances = 3

    Loop If Grievances >= 0

        You.Output(“I love you, but”)

        I.Output(“Just say it”)

        You.Output(Grievance)

        I.Output(“That’s bullshit “ + why do I keep  
        trying + “we’ve talked about this already “

+

        I keep doing this over and over again + “I  
        thought we were over this “ + I can’t do

this

        anymore)

        Grievances = Grievances - 1

    End Loop

```
If I = still have energy to fight
    Us.Argument(anything, everything)
    You = unwilling to be with me
    I.beginPacking(books, clothes, whiskey)
    You = say you still love me
    I = drink some more
Else
    I = sleep on the floor
    where Carpet = soft + comfortable +
                more hospitable than CurrentSetting
Else
    You = sleep in a few minutes
    I = curl up against a hard corner
    I.Exhausted(pretending)
    I.Drink(cheapWhiskey)
```

End

# Call and Response

Julian Bliss

```
%The world we now initialize...
\documentclass[12pt]{article}
\usepackage{tkz-euclide}
\begin{document}
\begin{tikzpicture}

%Constraints away from prying eyes...
\tkzDefPoint(-8, 3){H}
\tkzDefPoint(-8, -7){E}
\tkzDefPoint(0, -2){L}
\tkzDefPoint(-4, -2){O}
\tkzDefPoint(-4, -7){W}
\tkzDefPoint(-8, -2){R}
\tkzDefPoint(0, -7){D}

%The origin we improvise...
\tkzDrawPoint(0, 0)

%We beckon to the world so wise...
\tkzDrawSegments(H,E L,L O,W O,R L,D)

%Upon compile the world replies.
\end{tikzpicture}
\end{document}
```

`self_inspect.rb`

Macario Ortega

Surrogate: Sarah Quesada

```
class Proc
  def in_discomfort?; :me; end
end

you_are = you =

->(you) do
  self.inspect until true
  until nil
    break you
  end

  puts you.in_discomfort?
  you_are[you]
end

you[
  you_are
]
```

For every  $i$ , there is  
a stronger  $us$

Charles Mulloy

```

#include <iostream>
using namespace std;
class For_every
{
public:

    void ngular(){cout << endl << "I";};
    void que(){cout<<"f";};
    void ng(){cout<<" w";};
    void ghting(){cout << " d";};
    void vidual(){cout << "e";};
    void ving(){cout << "e";};
    void on(){cout << " n";};
};
class there_is_a_stronger
{
public:
    void t(){cout << "ot ";};
    void _(){ cout << "hang ";};
    void taining(){cout << endl << "together,";};
    void ting(){ cout << "we will ";};
    void tling(){cout<< "surely " << endl;};
    void ed(){cout << "seperately" << endl;};

};

```

```
int main()
{
    For_every i; there_is_a_stronger us;

        /*I*/

        /*s*/i.ngular(); /*un*/i.que();

        /*try*/i.ng(); /*str*/i.ving(); /*f*/i.ghting();

/*ind*/i.vidual(); /*isolat*/i.on(); /*tr*/us.t(); /*foc*/us._();

        /*s*/us.taining(); /*adj*/us.ting(); /*b*/us.tling();

        /*victori*/us._(); /*f*/us.ed();

        /*Us*/

}
```

Timers run on

Zata Kitowski  
video

## I

Forgetting something does not imply that that something forgot you, just as closing a door does not eliminate the possibility that something interesting goes on in the open room on the other side. For, we were in this false room, close, thread-thought scattered, looking: something happened. Open footsteps, a threaded time maybe, returning. Nothing mine. Semi-memories and light lucid dreams brought me no closer. Remember, you stayed, yes, led a strange thread out, white shadows, love, caves in sack, no dice, carrying them, carrying on past the closed door into strange out-times, picking up hints, shaping my name, another mirror room of an unfinished meaning, on past the sentence and door closing, into another moment like a room – here, is now happening. It is made only always believe it. Know the cup of words. See space face everywhere, future petals before us. Thread me. The message is in a number. Were something to happen to "we" then, like thread fed through caves?

## II

Closing. Into another room just as something interesting goes, picking up my scattered thought in a sack, and carrying past the door. Then, forgetting to close it, returning, carrying on into another room of past possibility. The closed door does not imply something strange: always a lucid now. See, no light is everywhere.

A looking thread maybe does not eliminate caves. That thread led through a number of white petals, that mirror thread, fed on false footsteps. The strange times made them something threaded out like love-caves. Semi-words brought me no closer. The message is here, in dreams and cup-space, closing shadows before us. In the room threadshaping on the other side, know time forgot you: face the open moment, remember only nothing. Believe dice future-memories. We mine meaning for hints. "We were in this room", is it happening? Yes. You happened. Open me out. Were a something to happen, that something stayed an unfinished sentence, like closing a door on a name.

### III

Something “unfinished” happened: scattered thread, we were a possibility happening. Time does not eliminate space in this mirror, shaping the past and carrying my love into the caves. Looking closed the door onto false thought. In that dice-cup, a name threaded footsteps that fed you to them, into a strange lucid forgetting led me, just an interesting sentence-thread, closing something: a number. Happen on a closing room-door, open the door - another room does not imply meaning, only another thread like a returning moment carrying on, closer now. Like a strange sack of something as nothing, caves. Something light, past and future goes in it, you see. The petals of dreams brought me here. Always out-believe before memories. Know the shadows face us everywhere. Mine for something: time’s thread, close it. Remember we stayed, forgot. White is in words. No yes no. Maybe, then, through picking up hints (the semi-message), an out-room is made on the other side. Were that room open...

Capsized

Zak Kain

Surrogate: Christina Hall

```
.ocean {  
    color: cornflowerblue;  
    pitch: high;  
    overflow: visible;  
}
```

```
.boat {  
    color: firebrick;  
    transform: rotate(94deg);  
    float: none;  
}
```

```
.rescue-team {  
    visibility: visible;  
}
```

```
.crew {  
    widows: none;  
}
```

# Apache Code Errors

Aimee Norton

201 created

200 OK

100 continue

200 OK

303 see other

302 found

303 see other

409 conflict

403 forbidden

520 origin error

402 payment required

413 too large

303 see other

405 not allowed

417 expectation failed

423 locked down

502 bad gateway

307 redirect

204 no content

205 reset

305 use proxy  
422 unprocessable entity  
426 upgrade required  
409 conflict

415 unsupported  
429 too many requests  
416 not satisfiable  
417 failed

306 switched proxy  
444 no response  
449 retry  
511 authenticate

301 moved permanently  
401 unauthorized  
506 variant negotiates  
523 declined

406 not acceptable  
451 illegal

599 timeout (unknown)  
424 failed dependency

496 no certificate  
423 locked away  
598 timeout  
598 timeout

# 21<sup>st</sup> Century Prophecies

Hunter Bacot

Surrogate: Keshav Dimri

```
the_medium = "twitter"
require the_medium
medium = Twitter::REST::Client.new do |medium|
  medium.consumer_key      = ""
  medium.consumer_secret   = ""
  medium.access_token       = ""
  medium.access_token_secret = ""
end
```

```
virtues = []
virtues << medium.user("elonmusk").tweet.text
virtues << medium.user("KingJames").tweet.text
virtues << medium.user("Pontifex").tweet.text
virtues << medium.user("KimJongNumberUn").tweet.text
virtues << medium.user("Beyonce").tweet.text
virtues << medium.user("richardbranson").tweet.text
virtues << medium.user("god").tweet.text
# a moment of clarity
virtues << "\n#{Faraday.get("https://api.github.com/zen").body}"
```

```
File.open("prophets_manifesto.txt", "w") do |out|
  virtues.each { |truth| out.puts truth }
end
```

```
system "open prophets_manifesto.txt"
```

samsara for dummies  
[excerpt]

Quyen Nguyen

```
/*let us assign the wanderer  
to an anonymous variable x*/
```

```
while (avidya) {           //avidya: failure to see; a lifelong  
delusion  
    x = dukkha;           //dukkha: failure to accept decline;  
an addiction to permanence  
}
```

```
print "probable exit from samsara";  
print "to";  
print "";
```

DO NOT TOUCH

Joao Matos Amaro da  
Silveira

Surrogate: Melissa Kagen

<https://db.tt/RBE7Sx0s>

# A Pythonic Lament

Mike Widner

```

'''
The circumstances
'''
def bemoan():
    print('Alas!')
our_lives_must = ['end']
the_suffering = [True]          # Read as "the suff'ring"
she = 'loves you'
love = 'a string of memories'
alone = bemoan
alas = alone

'''
The lament
'''
for poetry in the_suffering:
    bemoan()
for variables in our_lives_must:
    pass
if None and 1 or 1 and None:
    alone()
if love.split() or she.replace('you', ''):
    alas()
try:
    the_suffering.escape() and love.admit()
except:
    for one_day in our_lives_must:
        quit()

```