CS 5 Vantablack
The colour was exclusively licensed to Kapoor’s studio for artistic use, causing outrage among some other artists such as Christian Furr and Stuart Semple.[16][17] During a talk at the Denver Art Museum, Semple casually responded that his release of an ultra-fluorescent pink he had developed would not be allowed to be purchased by Kapoor. He later released a strong shade of pink with a non-binding disclaimer that Anish Kapoor was not allowed to purchase.[18][19] He later stated that the move was itself like performance art and that he did not anticipate the amount of attention it received.[20] In December 2016, Kapoor posted an Instagram post of his middle finger dipped in Semple’s pink.[21] Semple has also produced a colour-changing paint and cherry-scented deep black coloured paint that Kapoor is barred from purchasing in a similar fashion.[22][23] The company Nanolab partnered with Boston artist Jason Chase and released a
Vantablack stuff

• Connect 4 is due next Tuesday at 11:59 PM
• No print statements please in the Board or Game classes (from your “friendly” Autograder)
• This Friday, office hours end at 5:30
• Tuesday’s class
  – Project showcase
  – Work time on Connect 4 (bring your laptop)
Pretty Pictures!

Willow
Kyle
My Totally Excellent Trip to Beijing!
What’s this?
Von Neumann Architecture

CPU

central processing unit *registers*

RAM

random access memory locations

Von Neumann bottleneck

IR

Current - Instruction Register

r1

General-purpose register, r1

Assembly language is *human-readable* machine language

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>read r1</td>
</tr>
<tr>
<td>1</td>
<td>mul r2 r1 r1</td>
</tr>
<tr>
<td>2</td>
<td>add r2 r2 r1</td>
</tr>
<tr>
<td>3</td>
<td>write r2</td>
</tr>
<tr>
<td>4</td>
<td>halt</td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>&quot;mnemonics&quot; instead of bits</td>
</tr>
</tbody>
</table>

Human readable? I doubt it!
Von Neumann Architecture

Programs are stored in memory in machine language
A Computer

CPU (Processor)

RAM (Memory)

Instruction Pointer (Program Counter)

Instruction Register

Other Registers

Total snapshot of computer’s state!
A Computer

CPU (Processor)
- Instruction Pointer (Program Counter)
- Instruction Register
- Other Registers

RAM (Memory)

“States”

Time
- External input

00000000
01001000
10010011

01001000
11001000
Finite State Machines (in “Engineering”)  
Deterministic Finite Automata (in “CS”)  

- Finite number of states 
- Unique start state 
- One or more states are designated “accepting” 
- At each state, one transition per possible input 
- At end of input, “accept” if in an accepting state
Finite State Machines (in “Engineering”)
Deterministic Finite Automata (in “CS”)

- Finite number of states
- Unique start state
- One or more states are designated “accepting”
- At each state, one transition per possible input
- At end of input, “accept” if in an accepting state
Finite State Machines (in “Engineering”)  
Deterministic Finite Automata (in “CS”)  

- Finite number of states  
- Unique start state  
- One or more states are designated “accepting”  
- At each state, one transition per possible input  
- At end of input, “accept” if in an accepting state
Finite State Machines (in “Engineering”)  
Deterministic Finite Automata (in “CS”)

**Input:** 10110

- Finite number of states
- Unique start state
- One or more states are designated “accepting”
- At each state, one transition per possible input
- At end of input, “accept” if in an accepting state
Finite State Machines (in “Engineering”)  
Deterministic Finite Automata (in “CS”)  

Input: 10110  

- Finite number of states  
- Unique start state  
- One or more states are designated “accepting”  
- At each state, one transition per possible input  
- At end of input, “accept” if in an accepting state
Finite State Machines (in “Engineering”)
Deterministic Finite Automata (in “CS”)

- Finite number of states
- Unique start state
- One or more states are designated “accepting”
- At each state, one transition per possible input
- At end of input, “accept” if in an accepting state

Input: 10110

Accept Input!
Describe the Inputs accepted by these FSMs

“0, 1” means a 0 or a 1
Try These!

You can give the states any names you like!

1. Draw a FSM that accepts any string that begins and ends with a 0.
2. Draw a FSM that accepts any string that begins and ends with the same digit.
3. Draw a FSM that accepts any string that does not contain the pattern “110”.
4. Draw a FSM that accepts exactly those strings in which the number of 0’s is odd and the number of 1’s is odd.
5. Draw a FSM that accepts exactly those strings that represent multiples of 3 in binary (challenging!).

<table>
<thead>
<tr>
<th></th>
<th>16</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

= 3
= 6
= 9
= 12
= 15
This space for rent

(Call 1-800-CS5-Green)
Now, try these!

Worksheet...

• Draw a FSM that accepts its input if and only if the third digit from the *left* is a 1.
  
  Remember, the input is read from left to right!

  1010000: Accept!
  000101: Reject!

• Draw a FSM that accepts its input if and only if the third digit from the *right* is a 1.

  Hmmm… That sounds a lot harder! How do I predict that the digit that I’m seeing is the third digit from the end?!

  1010000: Reject!
  000101: Accept!

Challenge!
Now, try these!

• Draw a FSM that accepts its input if and only if the third digit from the left is a 1.

1010000: Accept!
000101: Reject!
Now, try these!

- Draw a FSM that accepts its input if and only if the third digit from the right is a 1.

Aren’t there some transitions missing here?

Which states are accepting states?
Finite State Machines are Everywhere!

25 cent Cokes? Where is this machine!?

(penney, fifty cent piece, silver dollar, Canadian currency, ....)

(some transitions not shown)
Finite State Machines are Everywhere!

Picobot!

state pattern $\rightarrow$ move  new state

0  $x***$ $\rightarrow$ N  0
0  $N***$ $\rightarrow$ S  1
1  $***x$  $\rightarrow$ S  1
1  $***N$  $\rightarrow$ N  0

$x***$ (move North)

$N***$ (move South)
Ran’s Office Temperature Controller

- Display 73
- Actuate buzzer at F&M
- Display Error message

Arrow directions:
- Up
- Down
AAA versus ATG

```python
>>> python -i lengthToATG.py
mean to ATG with 10000 trials  63.4509
mean to AAA with 10000 trials  83.0144
```
AAA versus ATG

```python
>>> python -i lengthToATG.py
mean to ATG with 10000 trials  64.199
mean to AAA with 10000 trials  84.345
```

Now, build a similar finite state machine for AAA!
AAA versus ATG

```python
>>> python -i lengthToATG.py
mean to ATG with 10000 trials 64.199
mean to AAA with 10000 trials 84.345
```