Computer Programming Fundamentals

CS 152
Professor: Leah Buechley
TAs: Melody Horn, Noah Garcia, Andrew Geyko, Juan Ormaza
Time: MWF 10:00-10:50am
https://handandmachine.cs.unm.edu/classes/CS152_Fall2021/

ASSIGNMENT 6
DUE FRIDAY 11/19

Extra credit can count for both extra credit and 2nd CA.

TALK TODAY:
MACHINE LEARNING & CS EDUCAITON

TUTORING

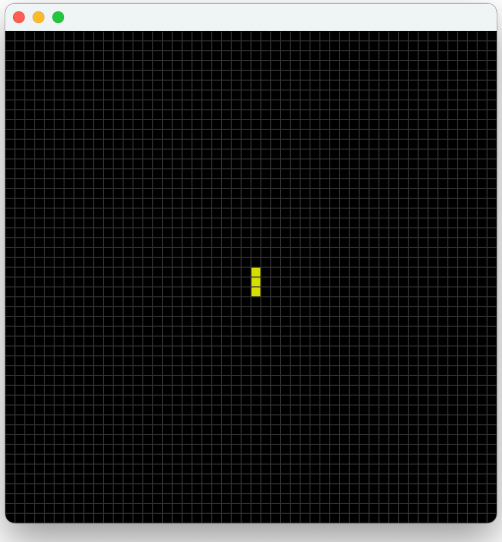
https://www.cs.unm.edu/

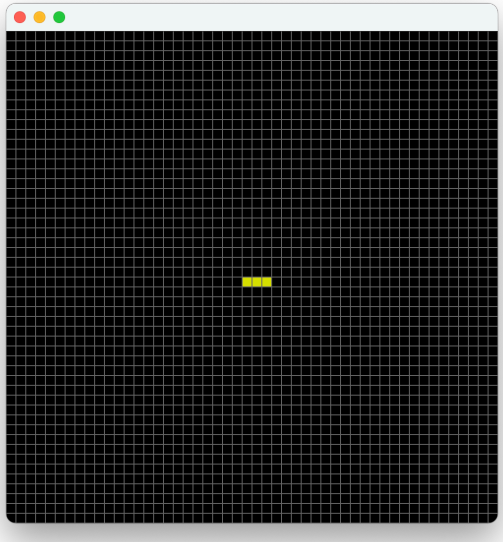
—> students —> peer tutoring

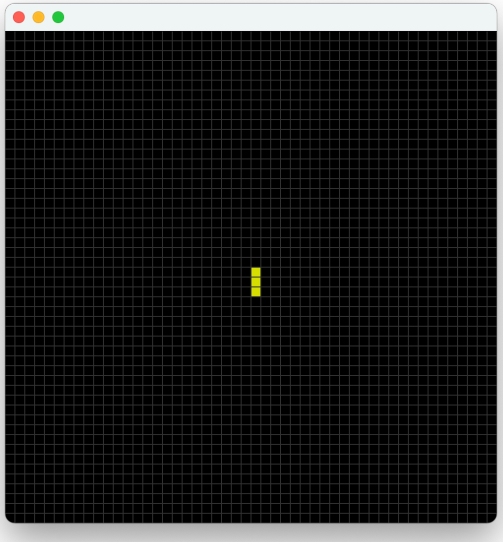
WHERE WE ARE

ANIMATE IN main

```
public static void main(String[] args) {
    CellularAutomata2DTest CA2D = new CellularAutomata2DTest();
    MyFrame frame = new MyFrame(CA2D);
    CA2D.animate(1);
}
```





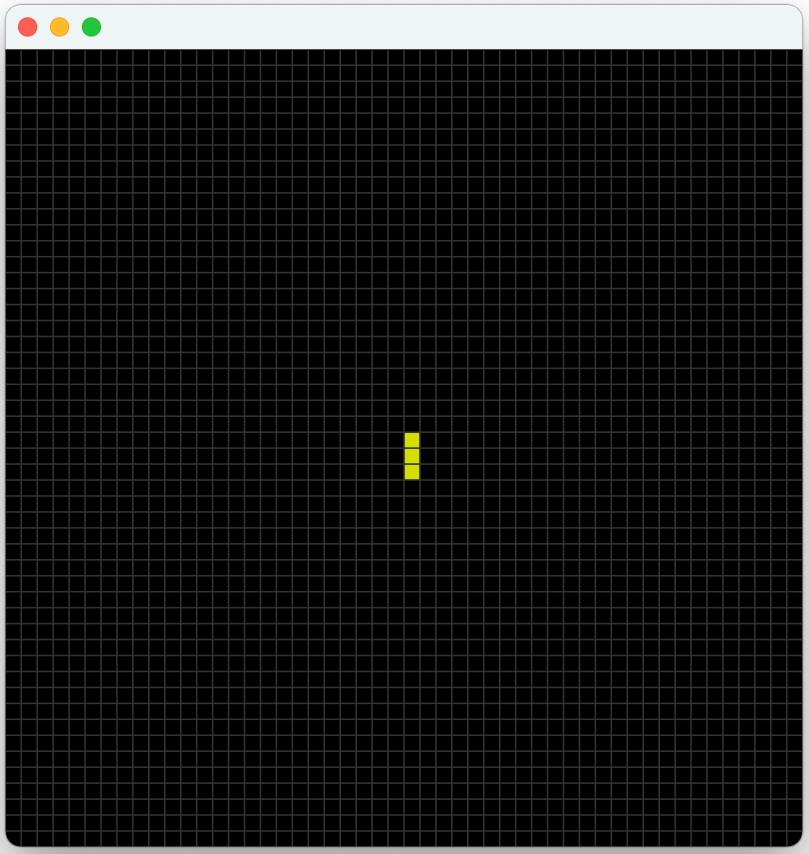


questions?

MORE OF A USER INTERFACE

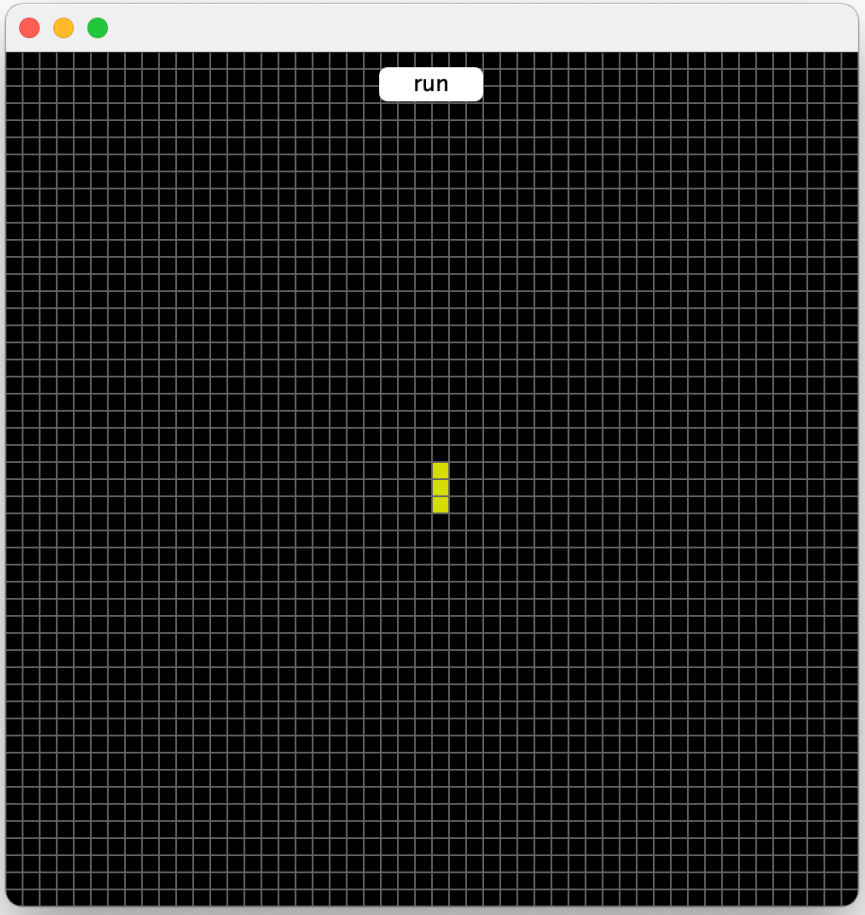
MORE OF A USER INTERFACE

- Ability to stop and start animation
- Ability to click on cells to turn them ALIVE or DEAD



STOP AND START ANIMATION

ADD A run/stop BUTTON



ADD A BUTTON VARIABLE

```
public class CellularAutomata2D extends BasicPanel implements ActionListener {
    int size;
    int[][] currentStates;
    int[][] nextStates;
    final int ALIVE = 1;
    final int DEAD = 0;
    final int CELLSIZE = 10;
    final Color ALIVE_COLOR = new Color(219, 224, 4);
    final Color DEAD_COLOR = Color.BLACK;
    final Color GRID_COLOR = new Color(100,100,100);
    JButton runButton;
```

Button will listen for actions

Type JButton

https://docs.oracle.com/en/java/javase/16/docs/api/java.desktop/javax/swing/JButton.html

https://www.javatpoint.com/java-jbutton

INITIALIZE IT IN CONSTRUCTOR

```
CellularAutomata2DTest() {
    size = 50;
    setSize(size*CELLSIZE,size*CELLSIZE);
    runButton = new JButton("run");
```

text “run” is what button will display

INITIALIZE IT IN CONSTRUCTOR

```
CellularAutomata2DTest() {
    size = 50;
    setSize(size*CELLSIZE,size*CELLSIZE);
    runButton = new JButton("run");
    this.add(runButton);
```

add the button to the panel (this)

INITIALIZE IT IN CONSTRUCTOR

```
CellularAutomata2DTest() {
    size = 50;
    setSize(size*CELLSIZE,size*CELLSIZE);
    runButton = new JButton("run");
    this.add(runButton);
    runButton.addActionListener(this);
```

add a listener to the button & panel

DEFAULT LOCATION:
TOP CENTER OF WINDOW

questions?

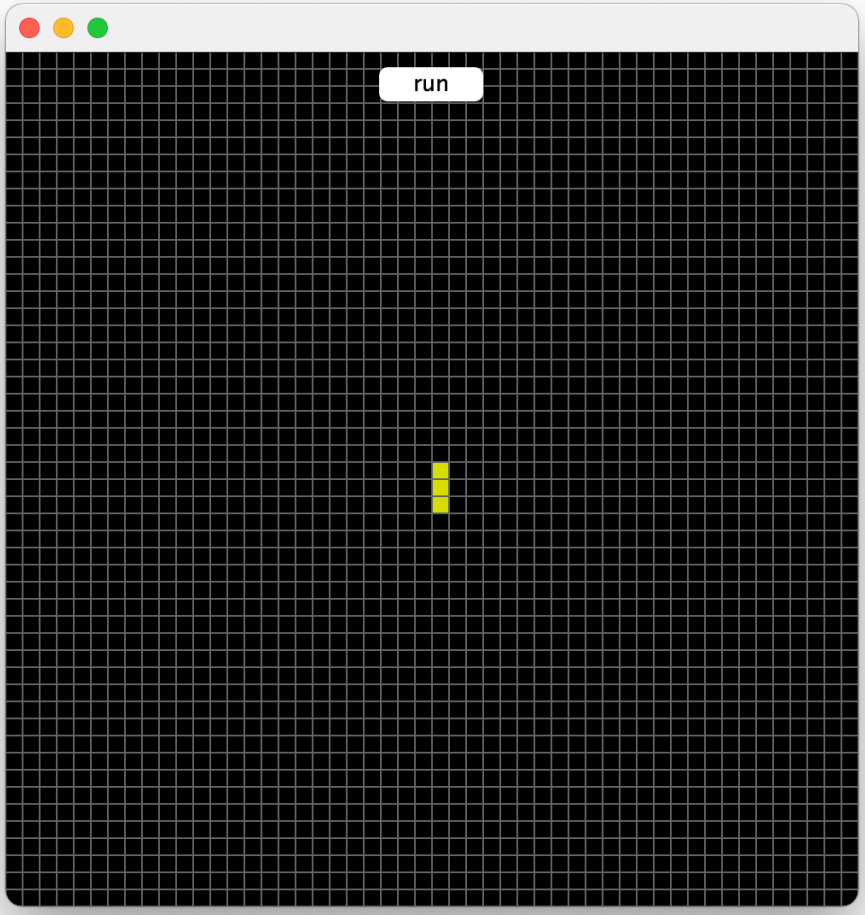
ADD actionPerformed METHOD

```
public void actionPerformed(ActionEvent e) {
    System.out.println("button pressed");
}
```

what runs when the button is clicked

COMPILE & RUN

run/stop BUTTON



HAVE BUTTON RUN AND STOP ANIMATION

ADD A running VARIABLE

```
public class CellularAutomata2D extends BasicPanel implements ActionListener {
    int size;
    int[][] currentStates;
    int[][] nextStates;
    final int ALIVE = 1;
    final int DEAD = 0;
    final int CELLSIZE = 10;
    final Color ALIVE_COLOR = new Color(219, 224, 4);
    final Color DEAD_COLOR = Color.BLACK;
    final Color GRID_COLOR = new Color(100,100,100);
    JButton runButton;
    boolean running;
```

will keep track of whether the CA is running or stopped

INITIALIZE IN CONSTRUCTOR

```
CellularAutomata2DTest() {
    size = 50;
    setSize(size*CELLSIZE,size*CELLSIZE);
    runButton = new JButton("run");
    this.add(runButton);
    runButton.addActionListener(this);
    running = false;

    currentStates = new int[size][size];
    nextStates = new int[size][size];
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            currentStates[i][j] = DEAD;
            nextStates[i][j] = DEAD;
        }
    }
    currentStates[size/2-1][size/2] = ALIVE;
    currentStates[size/2][size/2] = ALIVE;
    currentStates[size/2+1][size/2] = ALIVE;
}
```

begin CA in stopped state

not running

CHANGE running VARIABLE WHEN BUTTON IS CLICKED

ADD actionPerformed METHOD

```
public void actionPerformed(ActionEvent e) {
    System.out.println("button pressed");
}
```

will run whenever button is clicked

LOGIC

if running is true:

- stop running (set running to be false)
- change button text

if running is false:

- start running (set running to be true)
- change button text

actionPerformed METHOD

```
public void actionPerformed(ActionEvent e) {
    //if the CA is animating, stop it
    if (running) {
        running = false;
        runButton.setText("run");
    }
}
```

actionPerformed METHOD

```
@Override
public void actionPerformed(ActionEvent e) {
    System.out.println("button pressed");
    if (running) {
        running = false;
        runButton.setText("run");
    }
    else {
        running = true;
        runButton.setText("stop");
    }
    //get focus back in main window, off of button
    requestFocus();
}
```

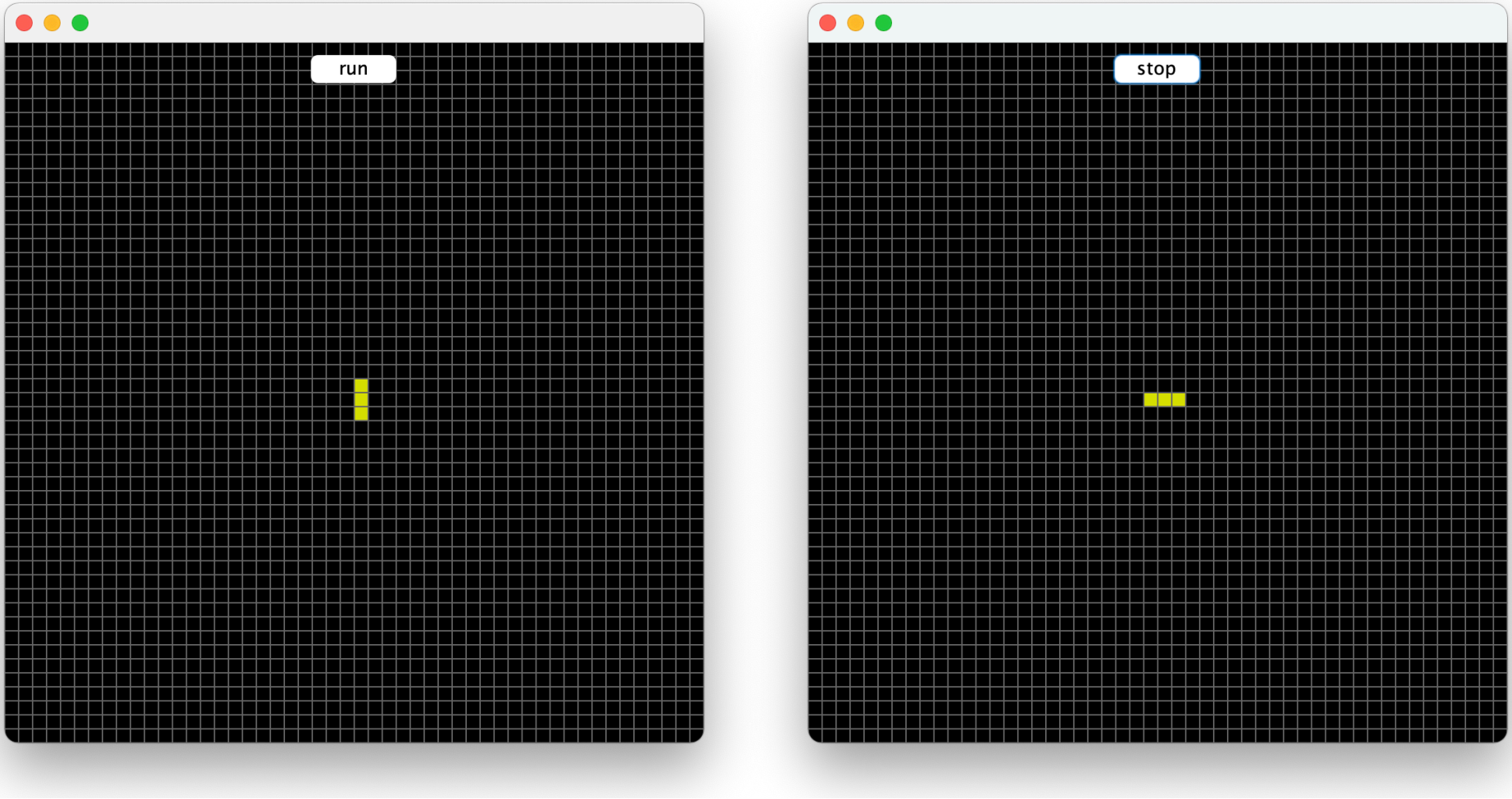
WHAT ELSE DO WE NEED TO DO?

ONLY ITERATE IF running IS TRUE

```
@Override
protected void paintComponent(Graphics g) {
    displayCurrentStates(g);
    if (running)
        iterate();
}
```

COMPILE & RUN

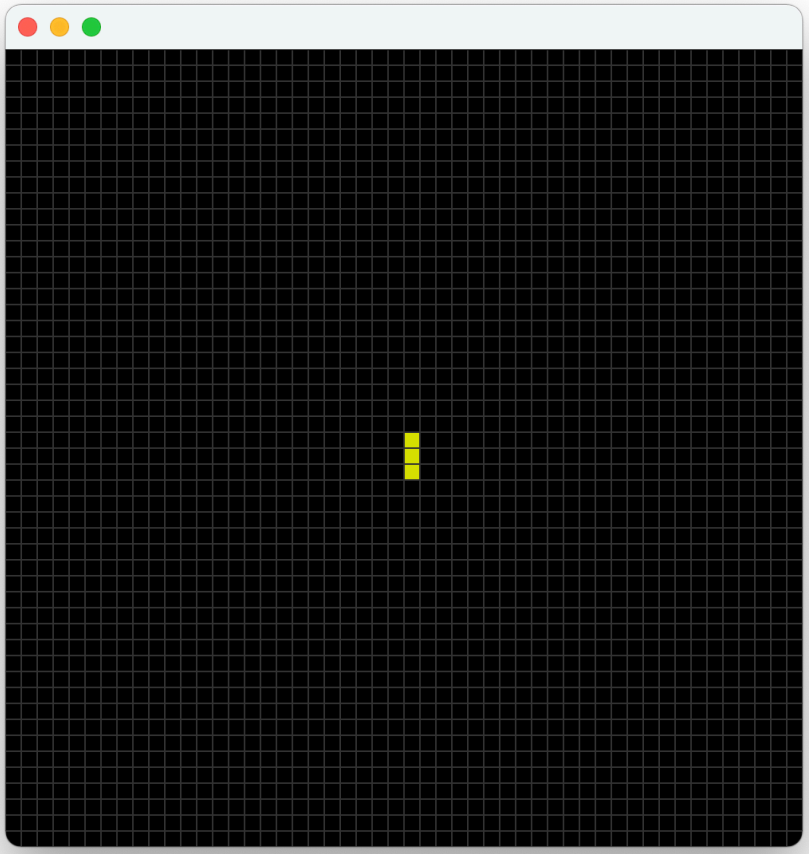
run/stop BUTTON



questions?

MORE OF A USER INTERFACE

- ~~Ability to stop and start animation~~
- Ability to click on cells to turn them ALIVE or DEAD



USE MOUSE CLICKS TO SET CELL STATE

ADD A mouseClicked METHOD

```
@Override
public void mouseClicked(MouseEvent e) {

}
```

will run whenever mouse is clicked

LOGIC
get click location
loop through all cells
if mouse was clicked inside that cell
    change state

ADD A mouseClicked METHOD

```
@Override
public void mouseClicked(MouseEvent e) {
    int x = e.getX();
    int y = e.getY();
}
```

LOGIC

get click location

ADD A mouseClicked METHOD

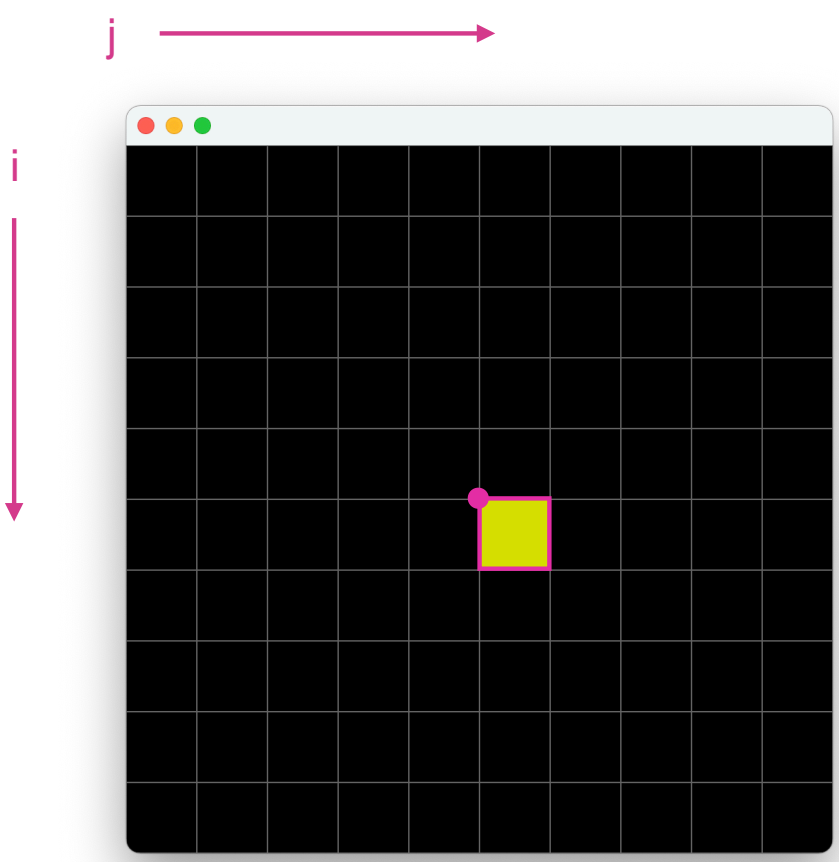
```
@Override
public void mouseClicked(MouseEvent e) {
    int x = e.getX();
    int y = e.getY();
    for (int i=1;i<size-1;i++) {
        for (int j=1;j<size-1;j++) {

      }
    }
}
```

LOGIC

get click location

loop through all cells

IS THE CLICK INSIDE A CELL?



- What is x coordinate for each cell?
- Using variables and constants?
- j*CELLSIZE

- What is y coordinate for each cell?
- Using variables and constants?
- i*CELLSIZE

- Conditions?
- (x > j*CELLSIZE && x < (j+1)*CELLSIZE)
- (y > i*CELLSIZE && y < (i+1)*CELLSIZE)

ADD A mouseClicked METHOD

```
@Override
public void mouseClicked(MouseEvent e) {
    int x = e.getX();
    int y = e.getY();
    for (int i=1;i<size-1;i++) {
        for (int j=1;j<size-1;j++) {
            if ((x>i*CELLSIZE && x<(i+1)*CELLSIZE) &&
                (y>j*CELLSIZE && y<(j+1)*CELLSIZE)) {

            }
        }
    }
}
```

LOGIC

get click location

loop through all cells

if mouse was clicked inside that cell

ADD A mouseClicked METHOD

```
@Override
public void mouseClicked(MouseEvent e) {
    int x = e.getX();
    int y = e.getY();
    for (int i=1;i<size-1;i++) {
        for (int j=1;j<size-1;j++) {
            if ((x>i*CELLSIZE && x<(i+1)*CELLSIZE) &&
                (y>j*CELLSIZE && y<(j+1)*CELLSIZE)) {
                if (currentStates[i][j]==DEAD)
                    currentStates[i][j] = ALIVE;
                else
                    currentStates[i][j] = DEAD;
            }
        }
    }
    repaint();
}
```

LOGIC

get click location

loop through all cells

if mouse was clicked inside that cell

    change state

ADD A mouseClicked METHOD

```
@Override
public void mouseClicked(MouseEvent e) {
    int x = e.getX();
    int y = e.getY();
    for (int i=1;i<size-1;i++) {
        for (int j=1;j<size-1;j++) {
            if ((x>i*CELLSIZE && x<(i+1)*CELLSIZE) &&
                (y>j*CELLSIZE && y<(j+1)*CELLSIZE)) {
                currentStates[i][j] = 1 - currentStates[i][j];
            }
        }
    }
    repaint();
}
```

LOGIC

get click location

loop through all cells

if mouse was clicked inside that cell

    change state

questions?

ADD A mouseDragged METHOD

```
@Override
public void mouseDragged(MouseEvent e) {
    int x = e.getX();
    int y = e.getY();
    for (int i=0; i<size; i++) {
        for (int j=0; j<size; j++) {
            //if at current cell, change state
            if     ((x>j*CELLSIZE && x<(j+1)*CELLSIZE) &&
                    (y>i*CELLSIZE && y<(i+1)*CELLSIZE)) {
                currentStates[i][j]=ALIVE;
            }
        }
    }
    repaint();
}
```

CLEAR THE SCREEN WITH keyTyped

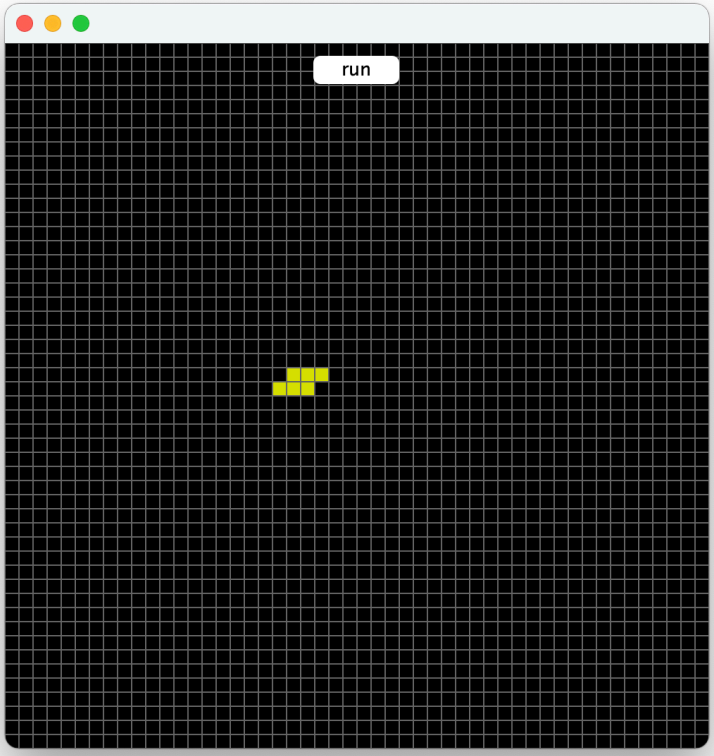
```
@Override
public void keyTyped(KeyEvent e) {
    char c = e.getKeyChar();
    if (c==' ') {
        System.out.println("clearing the CA");
        for (int i=0; i<size; i++) {
            for (int j=0; j<size; j++) {
                currentStates[i][j]=DEAD;
            }
        }
    }
}
```

COMPILE & RUN

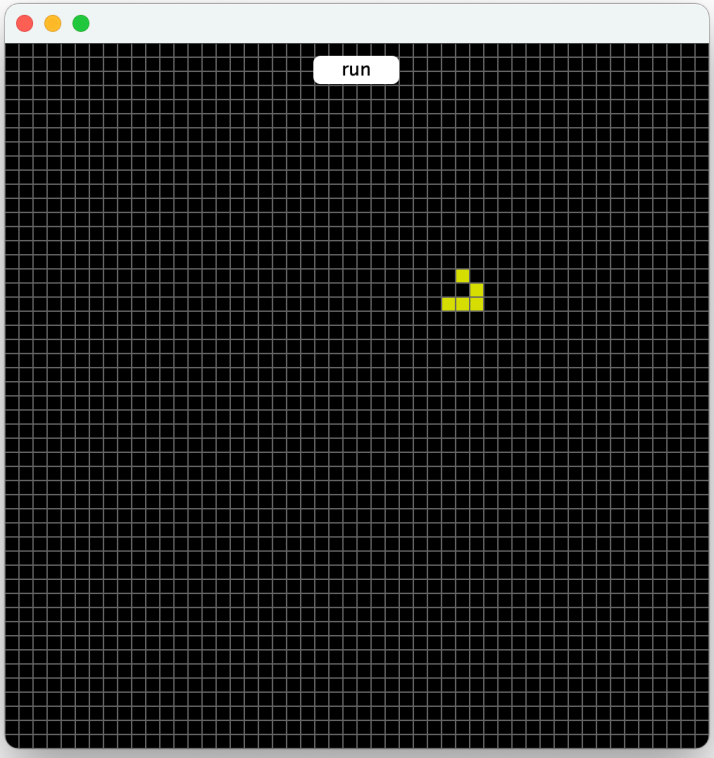
PLAY WITH DIFFERENT PATTERNS & ANIMATION SPEEDS

http://www.math.com/students/wonders/life/life.html

TOAD



GLIDER



Thank you!

CS 152
Professor: Leah Buechley
TAs: Melody Horn, Noah Garcia, Andrew Geyko, Juan Ormaza
Time: MWF 10:00-10:50am
https://handandmachine.cs.unm.edu/classes/CS152_Fall2021/