

# Term Project



***15-112 FALL20***

***TERM PROJECT LIGHTNING ROUND VIDEO***



15-112  
Lecture 2

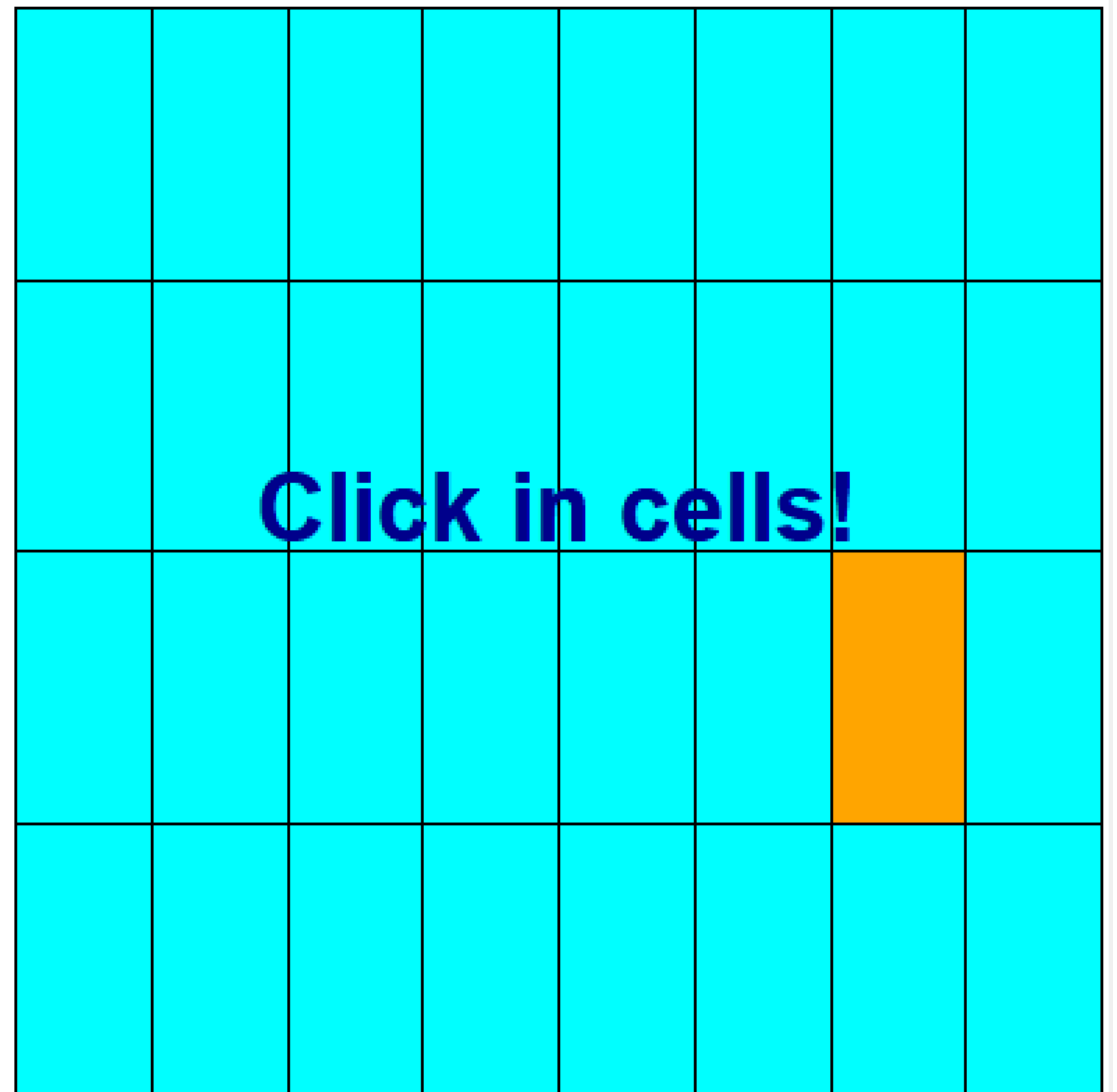
Week 6 Thu  
Anim3: Lists

Instructor: Pat Virtue

# Grid Example

See notes for equivalent example:

[Animations Part 2: Case Studies](#)

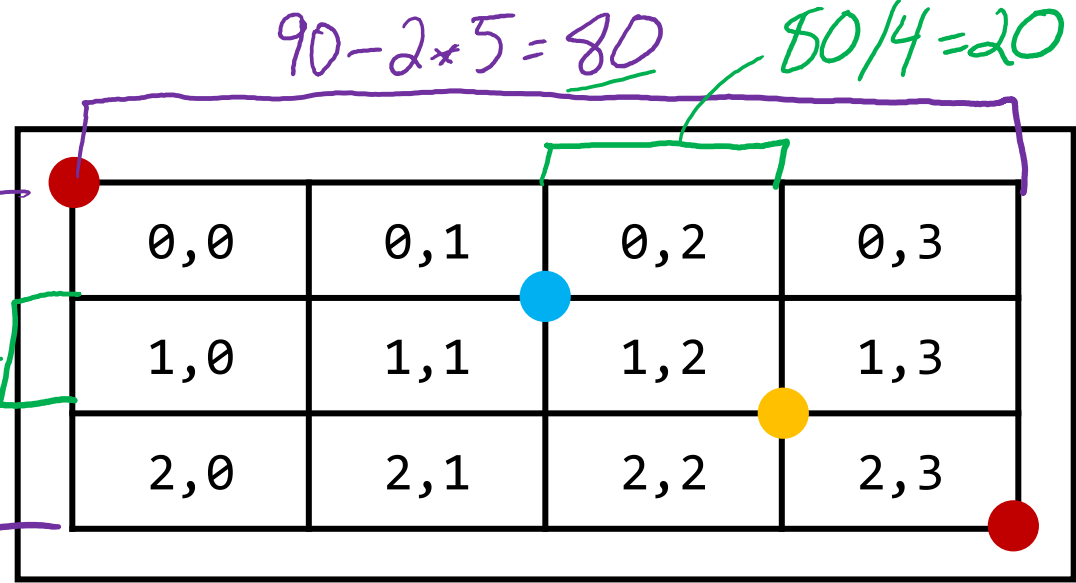


# Last time: Grid Worksheet

Given:

app.width = 90  
 app.height = 55  
 app.margin = 5

$gridHeight = 55 - 2 * 5 = 45$   
 $cellHeight = 45 / 3 = 15$



Fill in the following grid with the value return for each call to:

$x0, y0, x1, y1 = getCellBounds(app, r, c)$

$getCellBounds(app, 0, 0)$ 5, 5, 25, 20	$getCellBounds(app, 0, 1)$ 25, 5, 45, 20	$getCellBounds(app, 0, 2)$ 45, 5, 65, 5	$getCellBounds(app, 0, 3)$ 65, 5, 85, 5
$getCellBounds(app, 1, 0)$ 5, 20, 25, 35	$getCellBounds(app, 1, 1)$ 25, 20, 45, 35	$getCellBounds(app, 1, 2)$ 45, 20, 65, 35	$getCellBounds(app, 1, 3)$ 65, 20, 85, 35
$getCellBounds(app, 2, 0)$ 5, 35, 25, 50	$getCellBounds(app, 2, 1)$ 25, 35, 45, 50	$getCellBounds(app, 2, 2)$ 45, 35, 65, 50	$getCellBounds(app, 2, 3)$ 65, 35, 85, 50

# Last time: Model to View

```
def getCellBounds(app, r, c):  
    gridWidth = app.width - 2*app.margin  
    gridHeight = app.height - 2*app.margin  
  
    cellWidth = gridWidth / app.numCols  
    cellHeight = gridHeight / app.numRows  
  
    x0 = app.margin + c*cellWidth  
    x1 = app.margin + (c+1)*cellWidth  
  
    y0 = app.margin + r*cellHeight  
    y1 = app.margin + (r+1)*cellHeight  
  
    return x0, y0, x1, y1
```

# Last time: Draw Grid

```
def drawCell(app, canvas, r, c):
    x0, y0, x1, y1 = getCellBounds(app, r, c)

    canvas.create_rectangle(x0, y0, x1, y1,
        fill='', outline='purple', width=5)

def drawGrid(app, canvas):
    for r in range(app.numRows):
        for c in range(app.numCols):
            drawCell(app, canvas, r, c)

def redrawAll(app, canvas):
    drawGrid(app, canvas)

    drawCharacter(app, canvas)
```

# Last time: Move Character

### Conroler functions

```
def appStarted(app):  
    ...  
  
    app.charRow = app.numRows//2  
    app.charCol = app.numCols//2  
  
def keyPressed(app, event):  
    if event.key == 'Down':  
        app.charRow += 1  
    elif event.key == 'Up':  
        app.charRow -= 1  
    elif event.key == 'Right':  
        app.charCol += 1  
    elif event.key == 'Left':  
        app.charCol -= 1
```

### View function

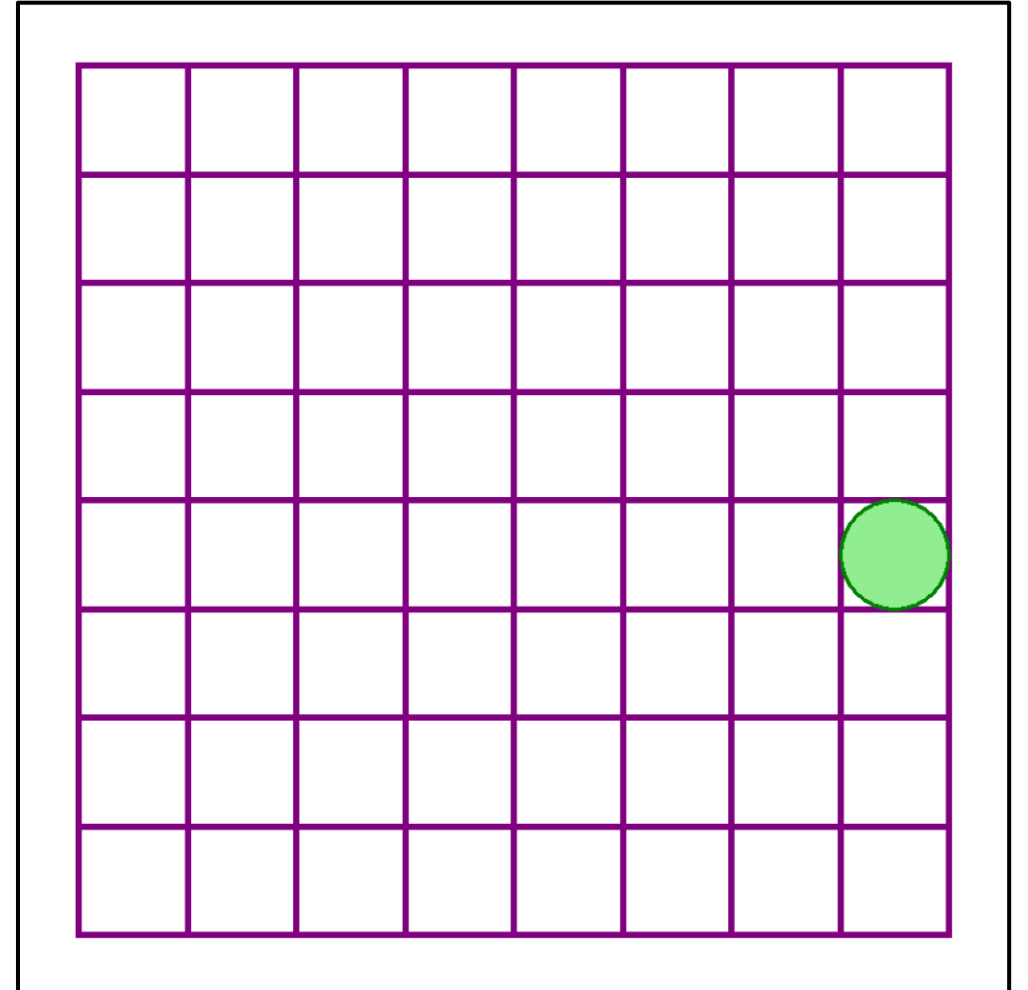
```
def drawCharacter(app, canvas):  
    x0, y0, x1, y1 = getCellBounds(app,  
                                    app.charRow,  
                                    app.charCol)  
    canvas.create_oval(x0, y0, x1, y1,  
                       fill='lightgreen',  
                       outline='green', width=3)
```

## Poll 2

Given existing code (in slides above):

What happens when we click the 'Right' key?

- A. No change (character stays)
- B. Steps one cell out of grid
- C. Wraps to far-left column
- D. Index out of bounds error
- E. Tkinter error





# Character on grid → Select a grid

## Additions

getCell(x, y) # Returns r, c

- View to model
- Analogous to getCellBounds(r, c) # Returns x0, y0, x1, y1 (model to view)

isPointInGrid(x, y) # Returns True or False

mousePressed

- To move character

timerFired

- With new **random** location for mole

# Select a grid → Whack-a-mole

## Additions

app.moleRow, app.moleCol

app.moleMaxTime

app.moleTimeLeft

app.score

drawMole (copy drawChar)

timerFired

- Decrement app.moleTimeLeft
- With new **random** location for mole
- Slow down with app.timerDelay

# MVC

model

Control

## Event Loop

View

runApp()

appStarted(app)

redrawAll(app, canvas)

clearCanvas(canvas)

mousePressed(app, event)

keyPressed(app, event)

timerFired(app)

# MVC

## Violations

- Never call view or control functions

```
NO: def createAThing(app):  
    app.thing = ['o w o']  
    redrawAll(app, canvas)
```

- Controllers never update the view

```
NO: def mousePressed(app, event):  
    app.r += 1  
    redrawAll(app, canvas)
```

- View (including helper functions) can never update the model

```
NO: def redrawAll(app, canvas):  
    app.time -= 1
```

# Debug tip: Pause and Step

```
def appStarted(app):  
    ...  
  
def keyPressed(app, event):  
    ...  
  
def timerFired(app):  
    # Change stuff in model
```

```
def appStarted(app):  
    app.paused = False  
  
def keyPressed(app, event):  
    if event.key == 'p':  
        app.paused = not app.paused  
    elif event.key == 's':  
        takeStep(app)  
  
def timerFired(app):  
    if not app.paused:  
        takeStep(app)  
  
def takeStep(app):  
    # Change stuff in model
```

# Course Feedback

<https://forms.gle/MKNokFdE1jhkDANF7>

Link also on pinned Piazza post

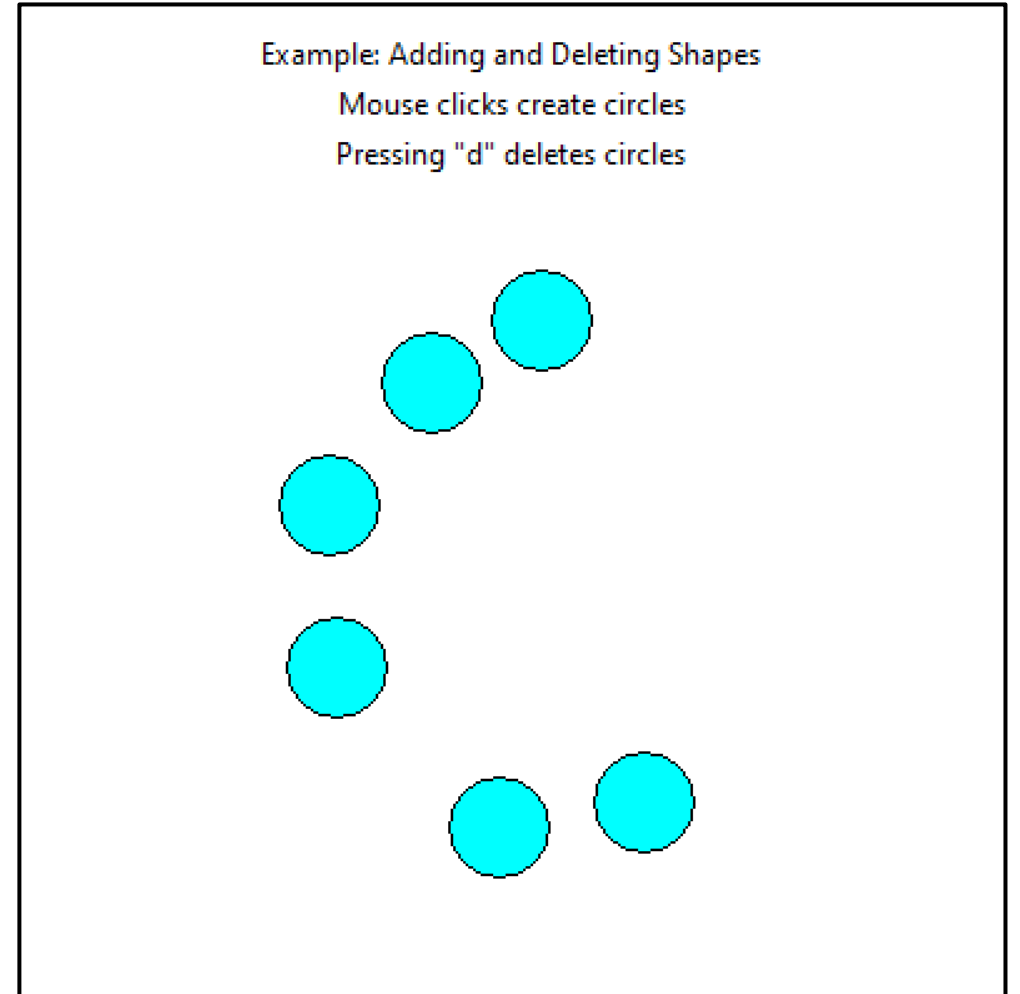


# Polygon Creator Example

See notes for equivalent example:

[Animations Part 3: Animations with Lists](#)

[Example: Adding and Deleting Shapes](#)





# Polygon Creator Example

See notes for equivalent example:

[Animations Part 3: Animations with Lists](#)

[Example: Adding and Deleting Shapes](#)

# Snake game!

<https://kfor.com/video/watch-snake-looks-like-video-game-tetris-as-it-climbs-brick-wall/8046787/>

# Snake Example

See notes for equivalent example:

[Animations Part 3: Animations with Lists](#)

