

# CompSci 94

## Review for Exam1

### September 23, 2021



Prof. Susan Rodger

# Class Today

- Review for Exam 1
- For Thursday Sept 30, next quiz and videos
- Exam 1 is Tuesday, Sept 28
  - Old tests are on Resources tab
  - See them on today's date with problems marked out (we have not done if, loops and written functions yet)
- Checkoff classwork you have done, also ok to check classwork 8 from Sept 21 in next class period with classwork, on Sept 30!

# Exam Logistics

- Exam is on paper
- Tuesday, Sept 28 regular class time
  - More time if you get accommodations
  - Should have gotten email from me
- The exam is your own work
- Do not talk about the exam with anyone until it is handed back
- See the Exam1 reference sheet
  - Alice snapshots of procedure names provided

# Exam Topics - Alice

- Alice Videos on warpwire
  - 2.x, 3.x, 4.1.0-4.2.2
- Setup, camera markers, invisible object markers
- Built-in procedures and functions
- Built-in properties: vehicle, opacity, height, etc
- Do in order, Do together
- Write a procedure with parameters
- Use procedure with arguments
- Random numbers, constant variables

# Best Way to Study for Exam

- Study Lecture notes, watch video again
- Study Classwork
  - Can you write a procedure on paper or type in file?
  - Try to recreate a classwork or write on paper
- Old exams are available on course web page
  - See “Old Tests” link (on resources tab)
  - **Practice writing methods on paper**
- Old Reading Quizzes available today as practice quizzes (for no credit) - part of studying
  - More important – practice writing code

# Old exams

- On resources tab on course web page
- Fall 2020, Fall 2019 and Spring 2019 – most like your exam
- Fall 2018, Spring 2018 – Alice 3 (material in different order)
- All other exams are Alice 2, which is different
- **Ignore HTML, CSS, IF stmts, loops questions**
- **See list of questions to study, ignore other ones**
- No classwork today, just review.
- Practice writing code on paper

# Some Practice questions

# Problem 1

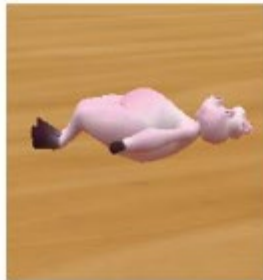
- Consider the following Alice code and the pig is standing straight up as shown with Start in the figure on the left below. Which figure A)-D) is where the pig will be after this line of code is executed?

```
this.pig.getLeftHip.turn FORWARD, 0.25.add detail
```

Start



A)



B)



C)



D)



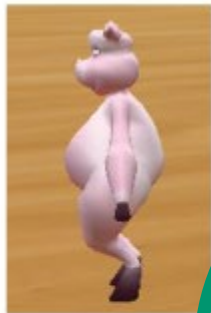


# Problem 1

- Consider the following Alice code and the pig is standing straight up as shown with Start in the figure on the left below. Which figure A)-D) is where the pig will be after this line of code is executed?

```
this.pig.getLeftHip.turn FORWARD, 0.25.add detail
```

Start



A)



B)



C)



D)



# You should practice writing code

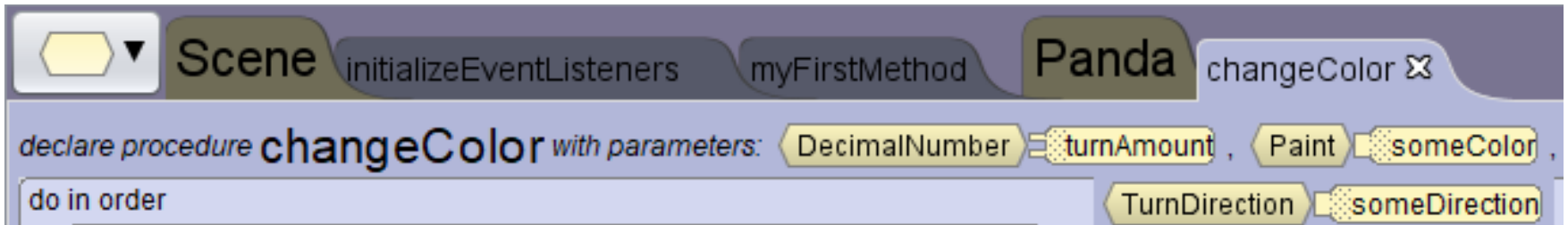
- Practice writing code from classworks and old exams

# Problem 2

## Write **panda** Procedure **changeColor**

- This procedure has **three parameters**
  - One parameter of type **Decimal** named **turnAmount**
  - One parameter of type **Paint** named **someColor**
  - One parameter of type **TurnDirection** named **someDirection**.
- When called, taking 3 seconds total, the panda turns around the **turnAmount** in the direction **someDirection** while at the same time changing to the color **someColor**.

# Write the procedure changeColor



The screenshot shows a programming environment with a top toolbar containing a yellow hexagon icon, a dropdown menu, and several tabs: 'Scene' (with sub-items 'initializeEventListeners' and 'myFirstMethod'), 'Panda', and 'changeColor' (with a close icon). Below the toolbar, the text 'declare procedure changeColor with parameters:' is followed by three parameter declarations: 'DecimalNumber' with a slider icon and label 'turnAmount', 'Paint' with a color swatch icon and label 'someColor', and 'TurnDirection' with a directional arrow icon and label 'someDirection'. Below this, the text 'do in order' is visible, followed by a large empty space for code.

# Solution

The image shows a Scratch code editor with the following structure:

- Project Name: Scene
- Scripts Area: initializeEventListeners, myFirstMethod
- Sprite Area: Panda
- Costumes Area: changeColor (selected)

The code for the `changeColor` procedure is as follows:

```
declare procedure changeColor with parameters: DecimalNumber turnAmount, Paint someColor, TurnDirection someDirection
do in order
  do together
    this setPaint someColor, duration 3.0 add detail
    this turn someDirection, turnAmount, duration 3.0 add detail
```

# Give the two calls to changeColor

- Give the call that has the panda turn right twice while turning Blue
  
- Give the call that has the panda turn left 1.5 times while turning Green

# Give the two calls to changeColor

- Give the call that has the panda turn right twice while turning Blue

```
this.panda.changeColor(turnAmount: 2.0, someColor: BLUE, someDirection: RIGHT)
```

- Give the call that has the panda turn left 1.5 times while turning Green

```
this.panda.changeColor(turnAmount: 1.5, someColor: GREEN, someDirection: LEFT)
```

# Problem 3:

## Write **Bunny** Procedure **funJumping**

- This procedure has **four parameters**
  - One parameter of type **Decimal** named **opValue**
  - Two parameters of type **Paint** named **color1, color2**
  - One parameter of type **Sdisc** named **someDisc**
- Before called, the bunny is standing on a disc that will be passed as an argument





# funJumping story(cont)

- The disc moves up 1 and back down to the ground carrying the bunny up and down with it. As the disc moves up it changes its color to color1 and the bunny changes its color to color2.
- Next the bunny changes its opacity to opValue
- The disc moves up 1 and back down again with the bunny
- Then instantly, the bunny turns back to its original color, the bunny is no longer faded and the disc disappears.

# Write the procedure funJumping

```
declare procedure funJumping with parameters: DecimalNumber opValue , Paint color1 ,  
do in order  
do in order Paint color2 , SDisc someDisc
```

# A solution

Scene initializeEventListeners myFirstMethod Bunny funJumping ✕

declare procedure **funJumping** with parameters: DecimalNumber opValue, Paint color1, Paint color2, SDisc someDisc

do in order

- do in order
  - this setVehicle someDisc
  - do together
    - someDisc move UP, 1.0 add detail
    - someDisc setPaint color1 add detail
    - this setPaint color2 add detail
  - someDisc move DOWN, 1.0 add detail
  - this setOpacity opValue, duration 0.0 add detail
  - someDisc move UP, 1.0 add detail
  - someDisc move DOWN, 1.0 add detail
  - this setPaint WHITE, duration 0.0 add detail
  - this setOpacity 1.0, duration 0.0 add detail
  - someDisc setOpacity 0.0, duration 0.0 add detail

Now let's look at some old  
exams