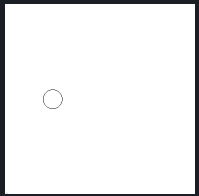
Announcements

- Homework
  - Homework 10 due tonight!
  - Homework 9 was graded and results pushed back to your repositories
- We'll talk projects the second half of class today
- Polling: `rembold-class.ddns.net`

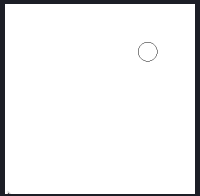
Review Question

The below code is run. Which image best depicts the state of the screen at the end of the program?

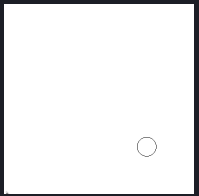
```
w = g.GraphWin('Example', 400, 400)
c = g.Circle(g.Point(100, 200), 20)
c.draw(w)
c.move(200, -100)
c.move(0, 200)
w.getMouse()
```



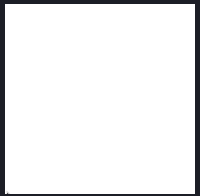
A



B



C



D

Understanding the Event Loop

- When you create a `GraphWin` object, `graphics.py` starts running its own event loop
- Runs alongside your code and controls the timing and actions that take place in the window
- Actions you make in your code get scheduled in the windows loop to get executed when the program has a chance
  - If not much is going on, this is likely immediately
  - If there is a LOT going on in your window, there might be a delay
- `update()` generally brings the event loop and your code into agreement before moving on in each
- `.after()` schedules a task to take place at some future time in the event loop (when its schedule is free)

Blocking Events

- Blocking events will hold up your code (not the event loop) until that event happens
- The `.getMouse()` or `.getKey()` methods are blocking
  - Python will not move onto the next line of your code until it either receives a mouse click or a key press!
  - Essentially pauses your code until that happens (which may or may not be desired)
  - Other things scheduled in the event loop may mean things are still happening in the window!

Return to Sender

- `.getMouse()` returns a `Point` object which contains the coordinates of where in the window a left click occured
  - Need to save the returned value as some variable to make use of it!
  - Can access the individual elements of the `Point` object using `.getX()` or `.getY()`
  - Only left click has an affect, and holding it down will not issue multiple clicks
- `.getKey()` returns a string of the key that was pressed
  - Again, need to save it as some variable to use it
  - Don't know what string a certain key corresponds to? Just capture it and then print it out!

Non-Blocking Events

- Non-Blocking events will *not* stop your code when run
- The `.checkMouse()` and `.checkKey()` methods are non-blocking
  - If no key or mouse is pressed when the command is run in your loop, `None` is returned instead
  - Regardless, your loop continues on, no pausing
  - If you are saving the value to be later used, make sure you handle `None` cases, since the majority of the time `None` is what will be returned

Projects

- Mostly groups of 3, which I have created and will be sending out
- Largely open to whatever you want or are interested in! I can help with ideas if you like.
- Will do a 10 min + questions presentation on the project on the last day of classes.
  - Should include:
    - What your project was
    - How you approached it and divided up the problem
    - Who worked on what parts
    - How it came together and works
    - Maybe a demonstration
- Code will be submitted to Github, but besides the presentation and code no other "output" necessary.

Timeline

- Campuswire groups made for each grouping
- Monday's class will focus on planning a project and collaborative coding
- Wednesday we'll do some fun things with a program I wrote
- Before break, your group should have arrived at strong idea of what you want to do, how you'll go about doing that, and who is responsible for what.
- After break, probably both the Monday and the Wednesday class and lab time will be devoted to time you can work on the project and ask questions.