Created by Joy Moglia
Contact: jmoglia@ischool.berkeley.edu

Project #1 (Spring 2021 term)
W200 – Section 12 (Kleeman)

Notes on Pygame

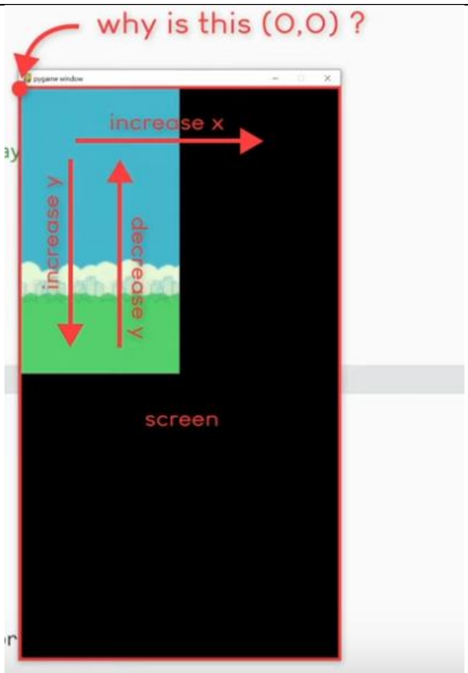
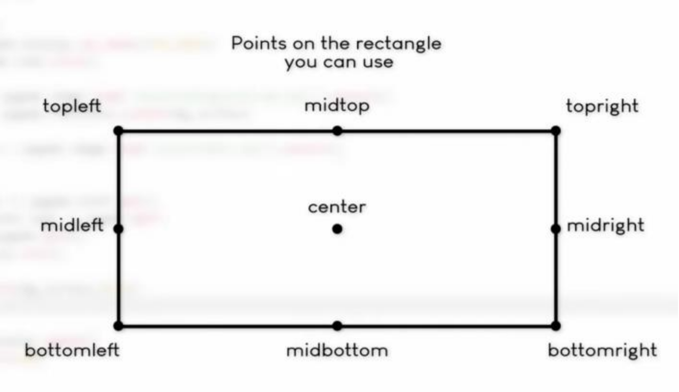
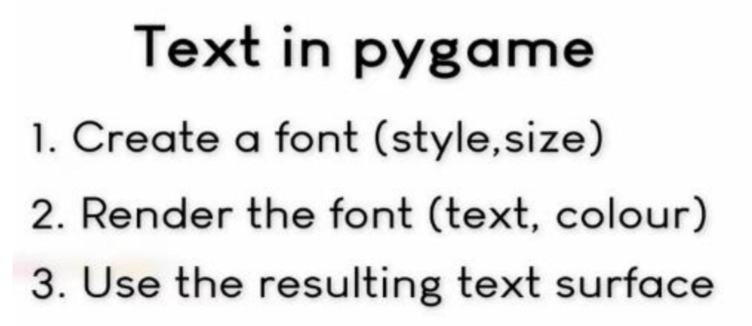
Pygame documentation: https://www.pygame.org/docs/

- Includes links to examples of code on Github
- Additional resources like where to find music here: https://www.pygame.org/wiki/resources
- Includes Pygame examples https://www.pygame.org/docs/ref/examples.html

Clock()	This controls the frame rate, or how many pictures/frames a game draws in a second. For example, 100 is faster than 10. It is good practice to limit the frame rate (ex: 120 is good limit)
Display Surface vs. [Regular] Surface	• The Display Surface object is the screen that opens when the user starts the game. There is only one. This is what the user sees. • Additional Regular Surfaces, containing images, can be added to the Display Surface but they need to be attached • Ex: a new image can be created on a Surface, which can be attached to the Display Surface and thus seen by the user • Surfaces cannot be moved; you need to put a rectangle or 'rect' around them first. • You cannot check collisions between Surfaces
Blit()	This method puts one Surface (which contains an image) on another surface (such as the Display Surface, or 'screen') using a pixel coordinate system that begins at (0,0) in the upper left-hand corner. Blit() is used to 'draw' the images onto the Displace Surface.

Convert()
Convert_alpha()
	Makes an imported image more compatible with pygame and run at a consistent speed. Convert_alpha() helps remove white backgrounds.

Transform	Used to scale an image to specified dimension. May reduce image quality.
Rectangle or 'rect'	In this game, Rectangles are placed around Surfaces that contain images (ex: skier, tree) so we can move them and check for collisions
rect1.colliderect(rect2)	Checks for collisions between rectangles and returns True or False
Event module	The event module is helpful for monitoring user input, like left or right arrows, as well as user-defined events.
USEREVENT	Can be used as a timer. For example, I use this to create trees, flags, and snowballs every couple of seconds. To create a new USEREVENT, just add +1
set_timer	second argument is time in milliseconds. 1200 ms = 1.2 seconds
Render	Render creates text on a Surface with a specified color. (0,0,0) is black. (255,255,255) is white.
Sprites	a class in Pygame that combines a Surface, a Rectangle and other features like animations and sounds. Sprites are a good representation of a 'real object' in a videogame. To implement, we can create classes that inherit from the Sprite class. Oddly, Sprites can't be drawn individually, they must be grouped first.
.kill()	Removes a sprite from a group

pygame.font.SysFont(‘font name’, size#)	to find names of fonts, go to: Control Panel\All Control Panel Items\Fonts
Coordinate system *Pygame’s coordinate system is used to place rectangles, which contain surfaces/images.*	 why is this (0,0) ? pygame window increase x increase y decrease y screen
Rectangle points	 Points on the rectangle you can use topleft midtop topright center midleft midright bottomleft midbottom bottomright
Creating text	**Text in pygame** 1. Create a font (style,size) 2. Render the font (text, colour) 3. Use the resulting text surface

-