If I am to be totally honest reading Bogost's writings was one of the most difficult readings that I have ever had to do. The in class lecture on what he was attempting to say helped my understanding a little bit, but I don't think I can agree with Bogost at all. His argument that the processes and functions that make up something, specifically video games, are what define it and give it meaning seems… strange? I feel like if I was to go and make a pinball machine, the solenoids, linear actuators, springs, microcontroller, etc. would not be what gives it value and meaning. If you asked anybody what they think of when they think pinball, they would probably mention a fun game or a nice memory associated with it. Maybe a select few would talk about how the mechatronics are really cool on a pinball machine, but they would be referencing it being cool in regard to the overall idea of the pinball machine and what it does and how it works. The value would come from what the pinball machine does and if it does it well, and if it was broken the sadness over that breakage would be over the machine not working rather than pin 7 on the Arduino burning out or a motor seizing. I would feel the same way about a video game. You could argue that any program that uses a for loop is basically the same process, but when the impacts of each program is so different I feel like that claim just doesn't hold up. I like the way that Sicart views the value within games and literature. It is less about what is going on "under the hood" and more important what these basic functions combine to do and accomplish.

As for my Inky effort, I think I got a little caught up in making a cute and fun little story. I played around with the syntax and figured out how to use knots and threads and the like. I played with some conditionals as well. I think in the future for any kind of branching path game it would be a better idea to plan everything out before hand in depth. I ended up writing myself into a corner in a portion of the script where a loop didn't quite work seamlessly in the story, and I wasn't able to efficiently fix it without doing some massive code changes. I think I should really create a knot or thread anytime I want a diverging path, even if I can just do branching decisions. That way it is a lot easier to come back around to.