The waterfall approach to app design is really interesting to me. It’s really easy to see how it would’ve been developed. The waterfall approach is top-down – all research and design work is done before any actual implementation is done. In situations where the app can only be used as a complete application, Waterfall makes the most sense.

In Game Development, it kind of feels like many of the situations where Waterfall may have been necessary have been phased out. Regular updates and patches have become regular in game development, which kind of goes against the all-in-one idea of waterfall. I feel as though Waterfall is mostly seen in game design nowadays when a game is being prepared by a third-party developer for a contractor.

I think one scenario in which Waterfall would be used today is in arcade machine work. While arcade machines nowadays can connect to the internet, they’re more susceptible to bugs or programming issues due to many of them being more involved. With a game like Dance Dance Revolution, for example, it makes a lot more sense to have the arcade machine’s controls and programming planned out in advance to make sure all planned features are feasible. Arcade games that use the attached guns would also benefit for similar reasons.

Another scenario where Waterfall would be useful in game development would be if you’re making a point-and-click adventure game like the Leisure Suit Larry series. Knowing what to plan for well in advance would help with foreshadowing and making sure none of the events are unfeasible.