March of the Northmen was designed for a 2-week long jam, and largely collapsed under its own weight. One of the classic lessons that developers are doomed to relearn until their dying day is scale. How well you manage scale is the metric by which any given project will live or die, and this one died. Though to claim it was entirely a problem of scale is an oversimplification. The failure to achieve a finished product came from three core issues: overambitious design, what I'm going to call 'sequencing issues', and untestability.

The first stumbling block is fairly straightforward. I took a 2 week deadline as license to make a 'real game', which largely meant having a large scheme of a game already planned before any testing or design. The creative process didn't really go "make a game about beating a drum and bolt extra parts onto it as time permits", but an idea of how that mechanic should evolve and be put into a context. Of course, no part of that mechanic existed or had been tested before that point. Extra time was put into building art assets or adapting scripts to be ready for that plan, and ultimately that detracted from refining the core of the project.

Somewhat related to that is what I would call 'sequencing issues'. Basically there were a few points when components were built in the wrong order, or without knowledge of where they'd attach. This is really a basic note, and I'm largely putting this in the report so I can tell my future self "I told you so" the next time I spend a day drawing background art without knowing how parallax is going to work.

The last, and most damning problem with the whole process was untestability. Essentially after putting together the most basic loop of this game, I fell down a hole of developing assets for an imagined game not the next iteration of the thing I was testing. But along with that was the issue that I was spending time adding to aethstetics before there was really a playable version of

the game ready. There wasn't any loop built into development, just building more assets without a clear picture of where they'd fit.

Putting all that aside, there were definitely some positive discoveries. The main one I've walked away from is an approach of singling out particular parts of the design, and sprinting through them. Various flavors of "sprints" are (were?) a hallmark of the sort of "code bro" circles, but Agile or Scrum have their virtues. I'm not a psychologist, but there's probably some aspect of the human neurological landscape that works well having one task to knock down in a handful of hours. So cutting a project into several blocks of 3-6 hours of sprints tends to improve productivity, which I suppose is the whole idea of game jams. Though interestingly enough, being in a jam with less of a time constriction made that production speed up more obvious as you had to consciously engage with it.

In brief, this was a failure and a success. There might not be anything playable, but valuable lessons were learned: likely making this the only instance of a failure to deliver.