

Night run was a Jam Game developed for Lux Jam 01, in 49 hours. It's a platformer that tasks the player with racing through an infinite series of minor platforming challenges, with an emphasis on speed to achieve a higher score before time runs out. I'm relatively happy with the project, despite a few lingering problems I'll be diving into a bit later on. At the end of the day, the game was finished, relatively satisfying to control, and managed to actually have layers of nested loops of gameplay. It's definitely the greatest success in this larger project to date.

The first question is "what went right". The short answer being that I learned my lessons from the first two jams: keep the player-designer loop short. Build playable modules, test them, break them, fix them, then repeat. The old legend about how Miamoto spent the first year developing *Super Mario 64* with nothing but a blank room and a controllable character was actually one of the first things in my mind when sketching out a timeline.

The real priority was "building a character that was engaging to control", and I'd say I got about 80% of the way to getting where I'd want to. The animation suite I'd put together was fairly basic (the walking animation was the only one with more than 2 frames) but did its job communicating and emphasizing what the player was doing. I'm particularly happy with the dash move I put in, which managed to be the most satisfying thing despite it's entire script being "player.AddForce(4000)". Allowing the player to hug walls and jump off of them was definitely a stumbling block (it's definitely too rigid when jumping off, and has some bugs when the player contacts walls), but ultimately it's something I'd wager most players could learn to control well enough with time. It's almost an accidental addition to the skill ceiling.

Really the only real failure was that I got slightly complacent towards the end of the development, and didn't plan for last second bug fixing. Which of course lead to a situation where the engine would apply gravity to the player correctly only *sometimes* when a build compiled and other times it would leave the player hopelessly floating in the air. I had to spend

time that could've been spent tuning other elements desperately trying to patch that, before just randomly building the game a few times until it exported without the bug. Unity's included modules and scripts haven't failed on me like that before, but I'm just chalking that up to Murphy's law.

The more artistic improvement made this iteration was in "flavor", for lack of a better word. The thematic explanation for why the player has a limited time before losing is that they're a vampire, and the sun is coming up fast. Bonus points are earned in consuming the living that are inexplicably standing around, and extra time by knocking over vaguely religiously-themed tombstones. Also the dawning sun in the background is both a good element of the UI but also is a stereotypically good thing coming to doom you, which gets at the fact you're playing the bad guy. It's no great shakes, but I'm proud to have found the time to hide a little narrative inside this one. What really did something extra was the few bits of text I managed to throw in as tutorial text and a game over screen.

For a time investment of around two minutes, I managed to give the game something of a personality, if only in the phrase "slay daywalkers, despise the light". The game over text is a bit over dramatic as well, but it does communicate with the audience. There's an artifact, and it's attempting to put impressions into the minds of its audience; art is definitely happening. Moreover, there are ways to do it that don't need to step on the toes of the more technical elements that have to exist before any purple prose can be added. Just taking the moment to consider what parts of a tutorial, or interface, or little art detail can be employed to reinforce or establish a theme.

Ultimately *Nightrun* has its flaws. The movement scheme definitely feels a few steps less fluid than you want from a precision platformer. There are bugs with collision detection that definitely come up in normal gameplay, affecting gameplay maybe once every 3

runs. The environments are relatively samey. Ultimately the game shows a lot of seams, especially in how it jutters whenever it's reset. But it works, and it has a little bit of charm and thematic intent. It's progress.