

Nigel Baillie

(734) 678-5328  
bnigel@umich.edu  
github.com/Resonious

Born April 11<sup>th</sup> 1995  
2287 S. Main St.  
Ann Arbor, MI

Mr. Zak McClendon  
Lead Game Designer  
Double Fine Productions, Inc.

Dear Zak McClendon:

I write to express interest in joining Double Fine as a gameplay programmer, as per the listing on your website. I'm currently a web developer at a local small business, and also a college student studying Computer Science at the University of Michigan. In addition to my professional and academic profile, I have been practicing game programming for fun since I was sixteen and have been playing games for as long as I can remember. Between my job, schooling, and hobby, I have developed a diverse set of general programming and game development skills.

To bring to the Double Fine table, I have solid hobbyist experience Unity, Unreal Engine 4, and SDL2, in addition to some 2D JavaScript libraries. Most notably Unreal Engine 4, of course, as I believe that is the backbone of Psychonauts 2. The project I used UE4 in was a Ludum Dare game jam entry, a side scrolling puzzle platformer in which the player arranges mirrors to redirect beams of light with special abilities. For instance, the purple rays propel the player in the direction they point. The game was developed mostly using Unreal's C++ interface and later I integrated the C++ functionality with Blueprint for quick prototyping. I found the Blueprint system is surprisingly fun to work with.

I'm no mathematician but I am plenty capable at mathematically involved tasks. I taught myself OpenGL and the Rust programming language by creating a 2D batching sprite renderer that supports scale, rotation about a point, and hue adjustment on the shader level. In addition to my hobby projects, my profession although not specifically games related, has provided me 3 years of solid software engineering experience in areas such as databases, deployment, user interface, and error reporting. I even wrote an image compositor for custom t-shirt mockups that outperforms ImageMagick by about 1.5 $\times$ .

While a games company probably doesn't often write their own CPU compositor, I think the under-the-hood knowledge does bear some meaning. More specifically, a step towards a more thorough understanding of the systems we work with. On the off chance that you'd like a more thorough understanding of the systems *I've* worked with, see the enclosed resume. I like what you guys do at Double Fine, and I'd be honored to join you. If you don't mind, I'll give you guys a call in a couple weeks when you've had time to think things through.

Sincerely,  
Nigel

# Objective

WANT 2 DO GAMES

---

## Education

---

Present  
Fall 2016

University of Michigan

- **Major:** Computer Science Engineering.

Winter  
2016  
Fall 2012

Washtenaw Community College

- Attended from second half of highschool through first half of college.

## Employment

---

Present  
Winter  
2014

Intermediate Developer at Ann Arbor T-Shirt Company

- I develop features and bug fixes for our quote building software, production tracker, online store, and main website using **Ruby on Rails**.
- Wrote mobile app for quick pricing using **React Native**.
- **Proposed** and implemented unified authentication system to provide a single sign-in for all in-house apps.
- Built domain specific image composition tool that out-performs ImageMagick at generating t-shirt mockups by about  $1.5\times$  using **C++** and libpng.
- Set up **continuous integration** for automatic testing and deployment of our Rails apps.

## Hobby

---

Present  
2011

Miscellaneous Game Development

- Participated in 3 **Ludum Dare** competitions (solo) and 1 jam (team.)
- Built a pixel-perfect platformer **collision detection** system based on the old Sonic the Hedgehog games for Sega Genesis.
- Implemented a lockstep-like **networking** protocol for said platformer using raw UDP (and TCP) sockets.
- Built a small 2D sprite renderer in the **Rust** programming language using **OpenGL** (“gl” crate.)
- Wrote a runtime **JVM bytecode** injection routine for a Minecraft mod to disable collision.