Byron Heads
802 Lowell St. Apt 2
Ypsilanti MI

Stardock Entertainment
Applicant Questionnaire

1. Are there any indie or personal projects you've worked on in your spare time that you've found interesting? Please provide some technical details.

   I have worked on several personal projects over the last few years. My most recently completed project was done for my Computer Game Programming class at Eastern Michigan University. We had a semester to design and implement a game from scratch. We were assigned to groups of three based on programming experience. Many groups chose to use Unity to develop their games, but we decided to write the engine ourselves using C++ and SFML.

   The game we create is called Office Zombie. It is a 2D isometric puzzle game. We created a simple puzzle game mechanic that involved moving office furniture around to trap office workers, to eat their brains, of course. The game offered many design challenges and interesting bugs. When creating the code that allowed the player to grab furniture to push and pull, I created a bug where you could move the furniture behind walls. Fixing the bug was solved by adding a more complex collision test.

   To handle all of the games resources (images, sounds, maps, game objects), I wrote a conversion and packing tool. This allowed us to edit the game data in a text editor and game maps in spread sheet. These files would be then packed into a single file that the game engine would read. Image resources and sound files were packed into a second pack file. The game engine could load multiple pack files allowing for mods to be loaded.

2. What's your favorite programming language and why?

   Currently I am torn between D and C++, but I would have to say C++. I like many of the language features of D, but the language is still academic. I find C++ a useful and powerful tool in game and application programming.

   C++ has new powerful and difficult features to use and master. I often do not use all of these features but its nice to know they are there when the situation arises. I have been programming in C++ for the last 8 years. I have used it on many personal projects and one small commercial project.

3. What's your favorite game (video, board or otherwise) and why?

   My favorite game is the original Civilization by Sid Meier. This was one of the first computer games I had ever played. I had an Nintendo system but Civilization was different. The game was not about timing when to jump or how quickly I could dodge something. Instead I had to plan and create to win the game.

   It was this game that got me interested in games and make my own. My family began giving me books on programming at a young age. My first game was an ascii adventure game written in QBasic. It included over 20,000 lines full of goto statements as I had yet to learn about subroutines and functions.

4. What is your favorite book (technical, fiction or otherwise) and why?

   One of my favorite books was the first programming book I ever read. It was a QBasic book that my father bought for me when I was ten. I had been trying to figure out how to use QBasic on my own at the time, but this book changed everything. This book opened up the world of programming for me. It also showed me that gotos were best avoided.

5. Outside of software development, what are some of your other interests?

   Several of my interests include beer brewing, biking, hiking, geocaching, and welding. In the future I would like to travel and explorer the world.

6. Have you ever contributed to an open source project or a game modification (total conversion or otherwise)? If so, which one(s)? What was your role?

   My contributions to open source has been though bug reporting. Most of which have been to the D compiler and the Ubuntu operating system. I have a few older programs available on source forge but they have been replaced by better programs. I have also released a few mods written for The Elder Scrolls Oblivion, and Fallout 3.

7. If you were going to start a small project tomorrow, are there any technologies and/or engineering practices you'd like to implement and/or experiment with?

   Right now I am currently exploring the use of real-time ray-tracing for gaming. I am profiling different data structures and algorithms trying to determine what works. Over the course of the next few months I will be developing a demo using ray-tracing.

   I have only worked on solo and small team projects, I would like to get experience working in a large team environment. Other practices I would like to experiment with are agile programming and pair programming.