Digital Portfolio

Intuition & Independent Development:

How Can Small-Scale Developers Help Combat the Growing Monopolism In the Software Industry?

By: Solomon Johnson-Rock

Pinnacle Project Research

Introduction:

Historically, innovation has always brought with it monopolistic business practices: Oil, cell services, computer hardware, and now software. With the limitless business opportunities, companies have been quick to put their resources towards innovative product development. While not negative in that regard, the imminent capitalization, consolidation, and stagnation of further innovations in that field are viewed as negative. However, with growing public discontent for large corporations and their platforms, it is now more possible than ever for independent developers to compete in the industry, preventing things such as the stagnation of innovation, decreases in service quality, increased subscription prices, and the development of predatory algorithms to "farm" engagement. By filling the market with more desirable products for consumers, and phasing one another out of the market.

Background Research:

An interesting factor of software development which is not often discussed is the mindset of the developer. Specifically, when facing challenges, independent developers will often turn to using intuition rather than relying on encyclopedic knowledge of coding principles. Several journals studying the process of software development have denoted it as superior for the purposes of making innovative software. Additionally, small teams and independent developers report greater productivity whilst using it.

Methodology:

I opted to do a journal documenting the development of a rudimentary 3D imaging software. At regular intervals, I would periodically put my progress on developing the software up against a rubric meant to measure how intuition is applied whilst facing challenges during development. Additionally, I compared my work to preliminary research I have synthesized in a literature review. All of this was means to measure how exactly intuition can impact independent software development, and how that impact can help independent developers combat monopolism in the tech industry.

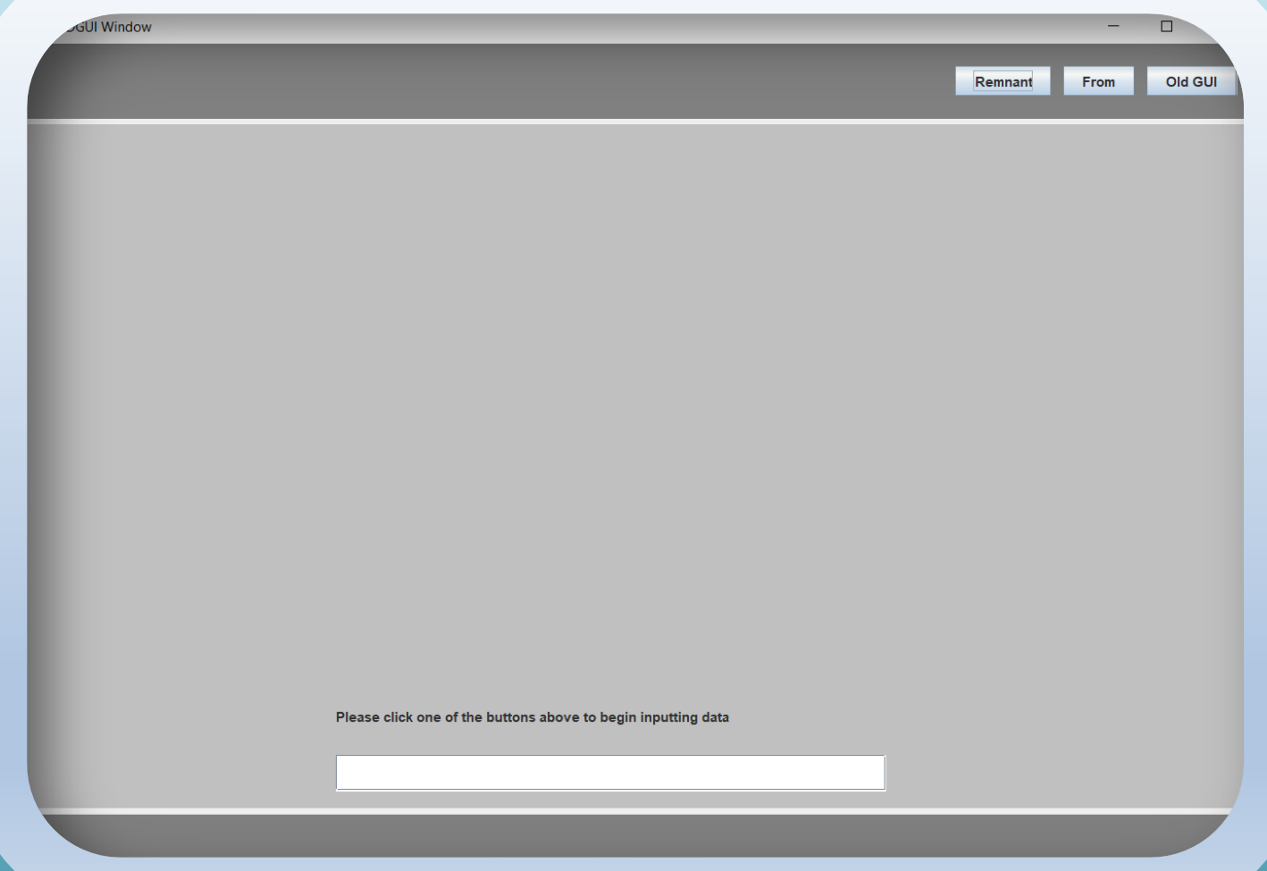
Definitions:

Intuition: or programmability as it's sometimes called, is the ability to read code and solve computational problems in a manner not formally taught. The method in practice is similar to puzzle-solving.

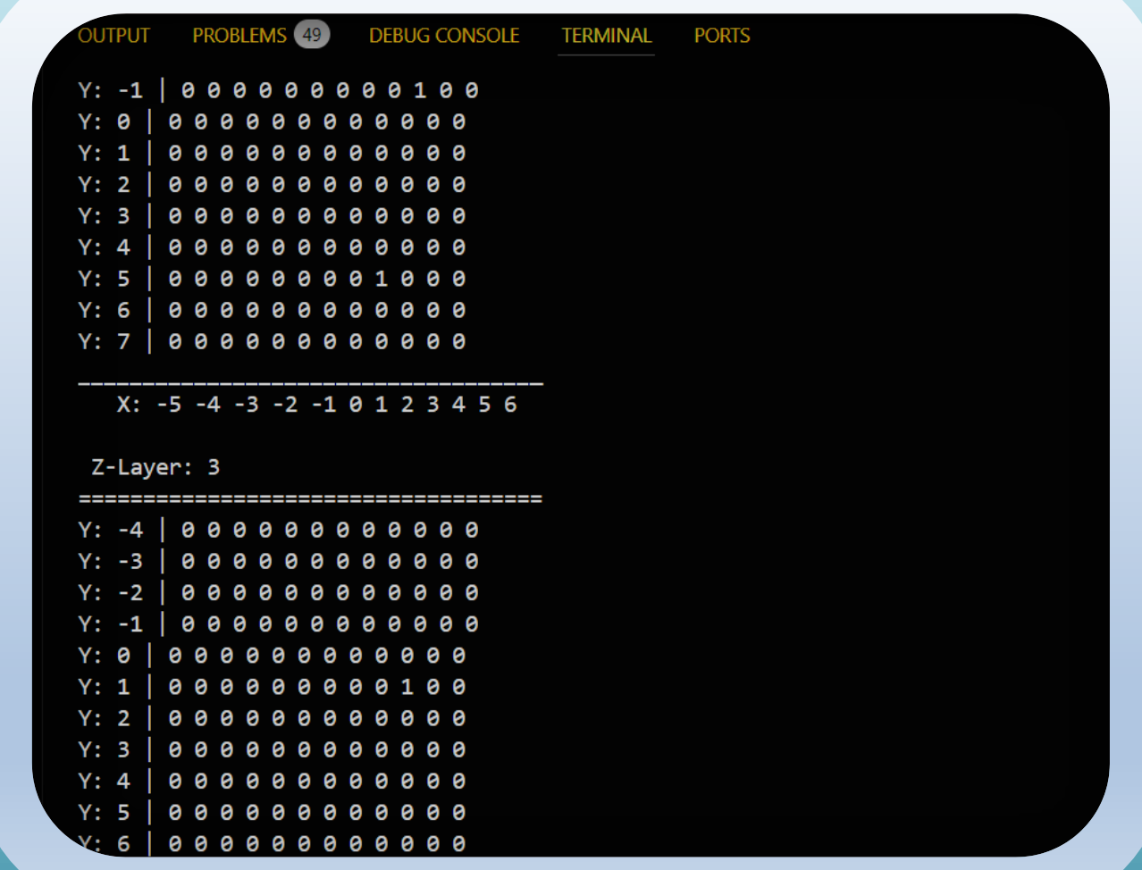
Software: A completely virtual product, which allows its user to perform specific tasks.

Efficiency: In this case, efficiency is a measurement based on how quickly a developer is able to meet a design requirement of their software within a certain span of time.

Prototype Images:

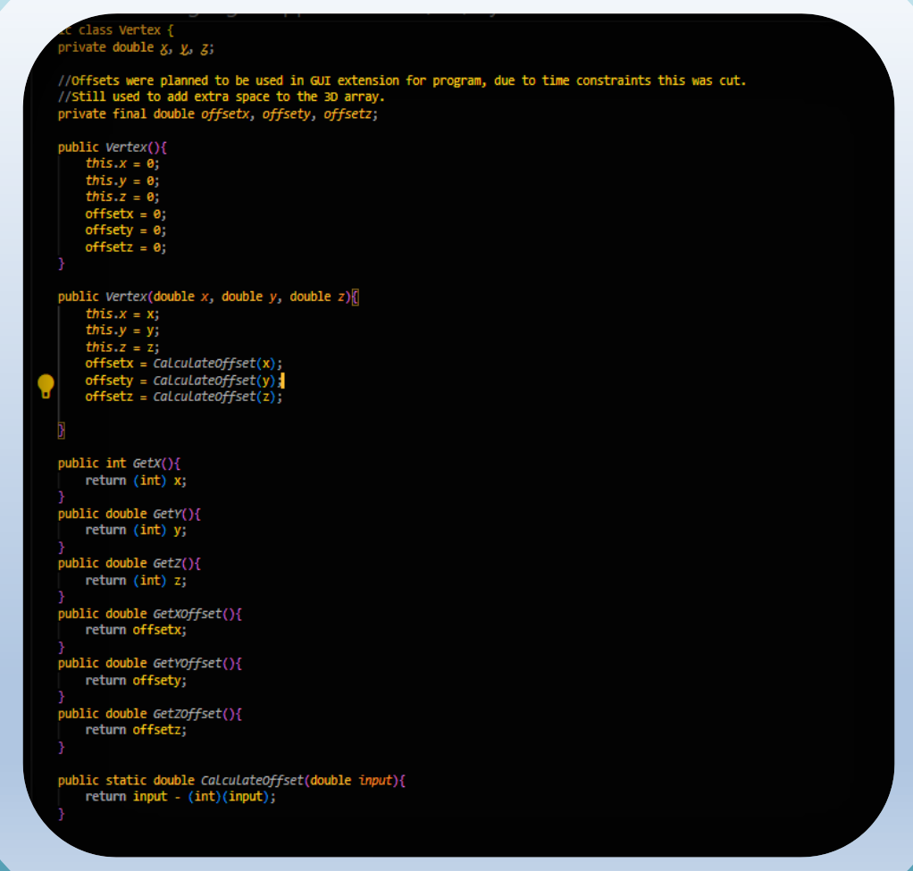


Please click one of the buttons above to begin inputting data



```
Y: -1 | 0 0 0 0 0 0 0 0 0 1 0 0
Y: 0 | 0 0 0 0 0 0 0 0 0 0 0 0
Y: 1 | 0 0 0 0 0 0 0 0 0 0 0 0
Y: 2 | 0 0 0 0 0 0 0 0 0 0 0 0
Y: 3 | 0 0 0 0 0 0 0 0 0 0 0 0
Y: 4 | 0 0 0 0 0 0 0 0 0 0 0 0
Y: 5 | 0 0 0 0 0 0 0 0 1 0 0 0
Y: 6 | 0 0 0 0 0 0 0 0 0 0 0 0
Y: 7 | 0 0 0 0 0 0 0 0 0 0 0 0
------------------------------------
   X: -5 -4 -3 -2 -1 0 1 2 3 4 5 6

 Z-Layer: 3
====================================
Y: -4 | 0 0 0 0 0 0 0 0 0 0 0 0
Y: -3 | 0 0 0 0 0 0 0 0 0 0 0 0
Y: -2 | 0 0 0 0 0 0 0 0 0 0 0 0
Y: -1 | 0 0 0 0 0 0 0 0 0 0 0 0
Y: 0 | 0 0 0 0 0 0 0 0 0 0 0 0
Y: 1 | 0 0 0 0 0 0 0 0 0 1 0 0
Y: 2 | 0 0 0 0 0 0 0 0 0 0 0 0
Y: 3 | 0 0 0 0 0 0 0 0 0 0 0 0
Y: 4 | 0 0 0 0 0 0 0 0 0 0 0 0
Y: 5 | 0 0 0 0 0 0 0 0 0 0 0 0
Y: 6 | 0 0 0 0 0 0 0 0 0 0 0 0
```



```
ic class Vertex {
private double x, y, z;

//Offsets were planned to be used in GUI extension for program, due to time constraints this was cut.
//Still used to add extra space to the 3D array.
private final double offsetx, offsety, offsetz;

public Vertex(){
    this.x = 0;
    this.y = 0;
    this.z = 0;
    offsetx = 0;
    offsety = 0;
    offsetz = 0;
}

public Vertex(double x, double y, double z){
    this.x = x;
    this.y = y;
    this.z = z;
    offsetx = CalculateOffset(x);
    offsety = CalculateOffset(y);
    offsetz = CalculateOffset(z);

}

public int GetX(){
    return (int) x;
}
public double GetY(){
    return (int) y;
}
public double GetZ(){
    return (int) z;
}
public double GetXOffset(){
    return offsetx;
}
public double GetYOffset(){
    return offsety;
}
public double GetZOffset(){
    return offsetz;
}

public static double CalculateOffset(double input){
    return input - (int)(input);
}
```

Links & References:

Literature Review

Research Journal

Findings

I've observed that the use of intuition during my software development resulted in increased efficiency whilst pivoting. The improvised coding solutions used to replace industry standards were easier to implement, in addition to opening new possibilities for innovation within the new solution. This should, according to the preliminary research from the literature review, be a suitable formula to compete with big name software, if applied to something more complex.

Works Cited:

Beynon, M., Boyatt, R., & Chan, Z. E. (2012, May 8). *Intuition in Software Development Revisited*. ResearchGate. https://www.researchgate.net/publication/228681862_Intuition_in_Software_Development_Revisited

Borowska, K. (2015, December 15). *The Monopoly On Technology And How To Defeat It*. Forbes. https://www.forbes.com/sites/kasiaborowska/2020/12/15/the-monopoly-on-technology-and-how-to-defeat-it/

Bourne, R. (2022, November 3). *A Monopoly of What? Big Tech in Today's Context*. Cato.org. https://www.cato.org/commentary/monopoly-what-big-tech-todays-context

Bourne, R., & Chiu, R. (2019, June 18). *Is This Time Different? Schumpeter, the Tech Giants, and Monopoly Fatalism*. Cato Institute. https://www.cato.org/publications/policy-analysis/time-different-schumpeter-tech-giants-monopoly-fatalism#introduction

Graziotin, D. (2013). The Dynamics of Creativity in Software Development. *Free University of Bozen-Bolzano*. https://doi.org/10.6084/m9.figshare.703568

M.H. Aabidi, Mahi, B. E., Chafik Baidada, Abdeslam Jakimi, & Ammar, H. (2017). Benefits of reverse engineering technologies in software development makerspace. *ITM Web of Conferences*, *13*(5), 01028. https://doi.org/10.1051/itmconf/20171301028

Scherer, F. M. (2015, June 25). *Antitrust Division | Technological Innovation And Monopolization*. Www.justice.gov. https://www.justice.gov/archives/atr/technological-innovation-and-monopolization