**A Solemn Evolution**

Adam Thiemann

Applied Computing: Business Focus

B.A.

Professor Michael O’Neill

**A Solemn Evolution**

1. **Statement of Purpose**

At the height of the Global War on Terror (GWOT) twenty-two veterans per day were committing suicide (Kemp & Bossarte, 2012). This unprecedented societal, civil, religious, and political failure of the United States to care for veteran mental health was appalling and unacceptable.

My project aims to assist society in helping combat veterans with “feeling connected,” by releasing my own memoirs with the intent of helping society to better understand combat, veterans to better understand their own experiences, and increasing the ability for Christians to deal with these relationships on an intimate level.

1. **Research and Background**

After returning from each of my tours in Afghanistan, I always came back to a community of men who understood our shared struggle, a “tribe.” I remember being with my comrades into the early morning hours on our returns from Afghanistan, long after we had been released to be with our families for the coming weeks. Men with wives and children, who hadn’t seen them in months would instead opt to stay at work for just a few more hours, until they could no longer reasonably withhold their presence at home without undue explanation. As a single man, living in the barracks, we would stay up all night, drinking, talking, reminiscing on the deployment, and enjoying one another’s company. Come the morning, we would sleep in or get on our early morning flights to go home, which was always the most difficult part. Even still, we were able to relax knowing that we would be back with our tribe soon enough.

Returning from Ukraine was not the same. My goodbye ceremony was the last time I would ever see many of my best friends alive. I fully understood the often espoused, though perhaps somewhat inflated statistic about “22 veterans” per day committing suicide (Kemp & Bossarte, 2012). My Ukrainian wife, who had never seen the frontline, was experiencing terrifying nightmares and toxic amounts of stress and anxiety upon the start of her master’s program in the United States. What was different about my experiences? Certainly, Ukraine had been a very different war, but that was not the issue. In his book, Tribe, Sebastian Junger makes the case that it is not the terrifying wartime experiences, but the lack of community in the regular world, that induce the disorder currently known as PTSD. He estimates that having a tribe not only limits the effects of acute post-traumatic stress but is the treatment for the disorder as well.

1. **Languages, Software, and Hardware**

The project was written using C++ and the Simple Direct Media Layer 2.0 library (SDL). It was compiled using mingw64 for Windows, and the IDE was Visual Studio Code. The game is optimized for a 1920x1080 resolution monitor but is playable on any Windows machine. It was also published to itch.io.

1. **Project Requirements**

A Solemn Evolution had 20 separate requirements listed and explained in the [Atomic Requirements](https://1drv.ms/x/s!Aofa4H6h1GyzgZlZAKAPaIFt1jeQZw?e=sY89Yz) excel document.

1. **Project Implementation Details and Description**

**Source Code:** [**https://github.com/adamrt918/CSU-Senior-Project**](https://github.com/adamrt918/CSU-Senior-Project)

* 1. **The Title Page**



The title page displays simple ASCII art and a place for the player to get to the main menu. This page is unique from the regular text because it is rendering a self-created .png file for the title screen and the picture is scaled based on the monitor dimensions. This is also where I get the monitor dimensions and begin the SDL window in full screen. The tutorial page is displayed in a similar manner, using a scaled .png rather than rendering text to display to the screen.

* 1. **Mute/Unmute Button**

**A black and grey sound icons

Description automatically generated**

Simple design which allows for the user to mute/unmute the music by clicking on the icon.

* 1. **Main Menu**

**A black and white text

Description automatically generated**

Though simple, this was my first exposure to displaying multiple elements on the screen and learning to render the text in a different color while scrolling over, and creating clickable events.

* + 1. **Formulas for displaying text:**

Monitor width \* iteration + 1 Width of texture

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ – \_\_\_\_\_\_\_\_\_\_\_\_\_\_

# of textures + 1 2

This formula puts each texture into its’ own horizontal column on the page, and centers it within that column. The number of textures is increased by 1 because the text should be centered within the column. The iteration starts at 0, but 0 cannot be used to multiply in the numerator because then there would be no margin, so 1 is added to offset the first element into the first column.

The height was done similarly. There is only 1 row, so I divided the page in 2, and subtracted ½ of the height of the texture.

* 1. **Survey Page**

**A screenshot of a survey

Description automatically generated**

As this was the first game and the first written work which I have ever put out to the public domain, I wanted to give people the opportunity to provide feedback. I needed to know if my writing was up to par, if the game was playable to the majority of people, if my game accomplished the goal for which it was written. It also helped to direct my bug fixing efforts.

* 1. **Taskbar**

****

The taskbar appears only when the game is started. It allows the player to navigate back to the main menu, or exit the game at any time on any game page.

* 1. **Player Statistics Bar**

****

The player statistics bar helps the player to guage how their game is going. It is not clickable and simply displays their current statistics.

* 1. **Game Pages**

There are 10 different page types implemented in the game. 7 of these pages are within the actual game portion. Though more classes should have been used to create the different attributes for these pages, the game formed itself around the work I was doing, and optimization was an afterthought as I learned the SDL from scratch and just tried to make things work. Standardizing the pages helped to implement some automation in the program. The different game pages, generally implemented the following order:

* + 1. Quotation Page

A black background with white text

Description automatically generated

3x Textures displaying the chapter, the quotation, and the clickable speaker/author of the quotation.

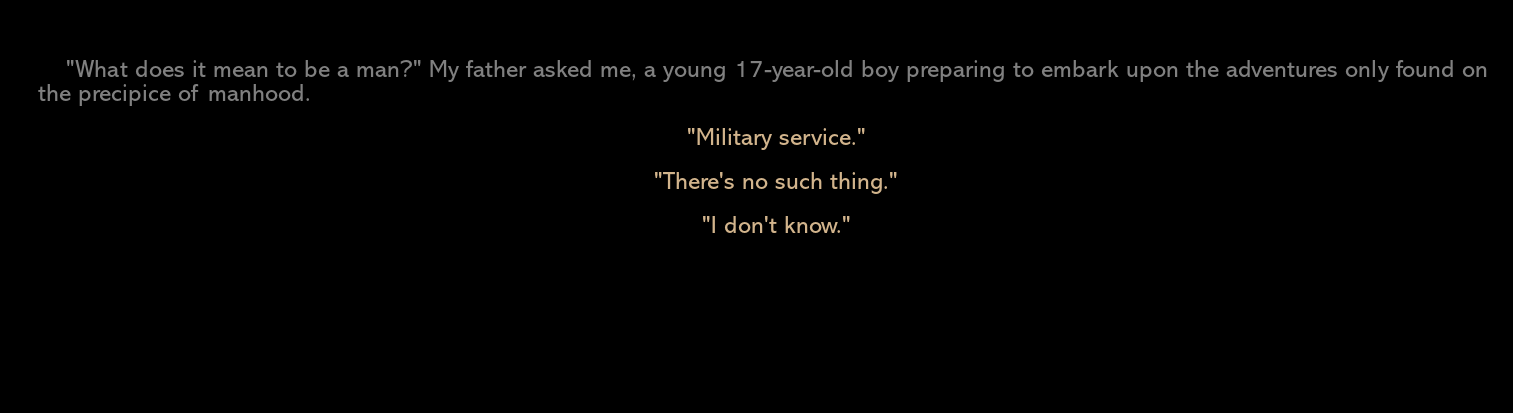
* + 1. Text Page

A screenshot of a computer

Description automatically generated

2x Textures displaying the plot of the story, and the clickable “Next Page” or other customized text.

* + 1. Choice Page



4x Textures displaying the 3 clickable choices, and 1 subdued color prompt. It also includes randomizing the numbers which will impact the player choice, and preparing the outcome of the choice.

* + 1. Outcome Page

A black background with white text

Description automatically generated

Shows how the players stats were affected by appending a string based on how the statistic was affected.

* + 1. Post Choice Page

A screenshot of a computer

Description automatically generated

Shows the plot outcome of the choice. 1 subdued texture reminds the player of their choice, the next texture is the plot outcome, and the last texture allows the player to advance the novel.

* + 1. Bible Verse Page

A black screen with white text

Description automatically generated

Displays a bible verse which is meant to convey the Biblical theme contained in the chapter.

* + 1. End Chapter Page

A black background with white text

Description automatically generated

Allows the player to make a choice to increase one of their stats before the page process repeats from the quotation page.

* 1. **Game Elements**

The game uses an even distribution random number seeded based on device hardware, known as a Mersenne Twister. Though I did wish to implement different distributions to further tune the effects of choices, I wasn’t able to code within the scope of this class. Each different choice is assigned a lower and upper bound. These bounds are input to the random number generator and the effect on a given statistic is generated, displayed, and used to modify the player statistics. Additionally, there are game modifiers such as going insane or dying.

* + 1. Sanity

A black background with gold text

Description automatically generated

Sanity is negatively affected by making cowardly decisions which preserve health but negatively impact their mental state. Going insane will force the player to make cowardly choices until their sanity recovers.

* + 1. Health

A black background with yellow text

Description automatically generated

Health is impacted by making selfless decisions which will impact one’s physical health. Dying results in a restart of the chapter and a reset of all stats. It is impossible to achieve a heroic ending if you die.

1. **Test Plan**

**A Solemn Evolution Test Plan**

**Introduction**

The purpose of this test plan is to provide unit test for which metrics which can be met during the initial creation of the game to better refine the future chapters and final product. This test plan is from chapter 1 and 2 and tests the technical aspects of the game through playability, and the qualitative aspects through a survey.

**References**

Proposal Document:

|  |  |  |
| --- | --- | --- |
| Date | Task | Description |
| 1/8/2024 | Proposal | Full project proposal complete. |
| 1/8/2024 | Requirements Doc | Requirements document complete. |
| 1/21/2024 | Text Complete | Storyline/plot writing complete. |
| 2/4/2024 | Game Outline | Outline RNG functions, choices, compilation of choices, and interaction of the user and the game. |
| 2/4/2024 | Tutorial | Tutorial complete. |
| ~~2/18/2024~~ | ~~Games Functions~~ | ~~RNG functions and all game theory completed pending testing.~~ |
| 3/3/2024 | Game Flow | Game flow added. Player should be able to go through the main story and read the plot. |
| 3/17/2024 | Secondary Game Flow | Health, sanity, and choices are added. Choices are tracked to determine end case. |
| 3/17/2024 | Test Cases | Choice results and impacts and health and sanity are mapped for difficulty and possibility to ensure that players can complete the game. Health, sanity, and choice weight values are adjusted accordingly. |
| 3/17/2024 | Survey | Complete the survey and begin polling |
| 3/31/2024 | Aesthetics | Graphics and music complete. |
| 4/14/2024 | Testing | Complete game testing based on test cases. |
| 4/14/2024 | Polling | Polling complete, results compiled into readable format to present for defense. |
| 4/17/2024 | Senior Project Defense | Presentation complete for defense. |
| 4/17/2024 | Final Adjustments/Edits | Last minute adjustments and edits complete. |
| 4/17/2024 | Project Completion | Final project published to itch.io |

**Features to be Tested**

* Playability
* Emotional Response
* Literary quality

**Features not to be Tested**

* Graphics
* Game theory equations

**Approach**

* Qualitative testing will be completed through the use of a survey
* Quantitative testing will be conducted through mathematical equations.
* Run the game from a clean windows install without issue.

**Item Pass/Fail Criteria**

* Playable from clean windows install is passing.
  + Unplayable from a clean windows install is considered failing.
* Player able to get to the ending of the game.
  + Player experiencing game ending bugs or issues considered failing.
* Greater knowledge of what combat veterans endure.
  + 80% or greater positive response to survey question considered passing.
* Biblical themes conveyed.
  + Responses to Biblical Themes will be analyzed for commonality. This is a qualitative criterion.

**Suspension Criteria and Resumption Requirements**

* Code optimization is required to implement further chapters.
* Further choices should not be instantiated without an understanding of the rates of death and insanity occurrence.

**Test Deliverables**

* Test Plan
* Test Cases included with results in the Atomic Requirements document.

**Test Environment**

* Windows x64

**Approvals**

Adam Thiemann (Developer)

Professor O’Neil

1. **Test Results**

Test Cases and test results included in the [Atomic Requirement](https://1drv.ms/x/s!Aofa4H6h1GyzgZlZAKAPaIFt1jeQZw?e=rwGdAt) excel spreadsheet.

1. **Challenges Overcome**

This project was a race against time from the start. It was incredibly ambitious to believe that I could produce a game of near the significance which I wished to produce within one semester while working full-time and attending school greater than full time. I had no knowledge of the SDL prior to beginning my project and spent two weeks trying to get the main menu to load properly. Once I understood the process of loading media, creating clickable rectangles, and rendering to the screen, I was able to begin coding the game. This lead to my next great challenge, optimization.

Since I had no idea how the SDL worked, and just needed to start coding, I began coding everything in main trying to just get something to work. Once I had the game flow down, I was copy pasting hundreds of lines of code for every page, in 3 different sections. I had to stop and optimize or I was never going to finish. Each single page was taking me 2-3 hours of work. 2 days of that and I stopped everything to migrate to a more optimized setup. I created choice, player, renderer, and choice page classes, an enumerated type that could specify which choices were cowardly, average, or heroic so that I wouldn’t keep forgetting which number was which, and migrated huge sections of my code in main to their respective classes piece by piece.

In the end, time came back to haunt me. Though I had written roughly 10,000 words for plot, I was only able to include 2 chapters and the ending, about 8,000 words. My plan was for 3 full chapters and game functions further tailored toward different types of distributions, but without further optimization and bug fixing, it was a futile attempt. Once I had a product that was mostly playable, I deployed it, leading to my next headache.

I had never deployed a game before. I thought the main.exe file would be enough to run the whole program, but I had to zip the file. Then, I created a Windows virtual machine to attempt to run the .exe from a clean Windows install. My visual basic setup had paths to the files needed to compile and run the game. So, by looking at the error code each time I tried to run the game from my Windows VM, I would have to deduce the file that was missing, and search through my mingw, SDL, and other coding environment files to find the file which the errors were referring to. Finally, the game ran.

Lastly, marketing the game online and to friends and family proved to be my least favorite part. I don’t want to push things on people who are busy with a game that I considered to be far below par. Regardless, I asked my friends and family to play. There were many different issues with all the different versions of Windows and the modifications being run, but the biggest issue was running an unlicensed .exe file. I’m fairly computer comfortable so I don’t choose to use anything other than Windows defender; however, all of the anti virus software was preventing people from running my game, and I believe ultimately prevented the game from taking off when it found local popularity on itch.io.

1. **Future Enhancements**

In all likelihood, there will be very few future enhancements. Text based games are no longer as popular as they once were, and though for some they provide a sense of nostalgia, they are dying. Regardless, if I were to continue to improve the program, I would first focus on cleaning up the structure of the code. Each page type needs its’ own class with its’ own rendering, events, and load media function.

I would also work on converting all of the text into a single long .png file. Then, I would use the monitor dimensions of the user to scale the amount of text clipped out of the .png, and only render the “Next Page” text as text beneath the .png. This would better optimize the display of code for users with different monitor sizes.

Initially, I planned on doing 5 chapters. There would be the same two chapters, a chapter for growth, a chapter for fighting in Ukraine, and a chapter for lessons learned. This would likely an additional 5-6000 words. Eventually I would like to add those chapters in, when the code is more optimized and I’m not copy pasting hundreds of lines of code per page.

Lastly and most importantly, the game should be playable on a browser. Running an unlicensed .exe file was the biggest limiting factor in getting the game playable and widespread. Though my friends and family, and very few random individuals on the internet, ignored their anti virus software reccomendations, the game should have never needed to execute. It should have simply run on the web.

1. **Defense Presentation**

[Defense Presentation powerpoint.](https://1drv.ms/p/s!Aofa4H6h1GyzgZld5QeGiB_pyaYpCQ?e=lmOYsF)

**References**

Kemp, J., & Bossarte, R. (2012). (rep.). Suicide Data Report, 2012. Department of Veterans Affairs.