Jeffer Peng

ICS3U1-01

Final Written Report

May 21st 2016

**Final Report: The Conqueror**

**1. Game Introduction**

Our program is a shooting game programmed through pygame. The purpose of the game is to allow the player to take place of an alien that is invading Earth and trying to take over the world. There are four fictional countries representing four different campaigns that the player has to complete to finally “conquer the world” - Asia Union, Soviet Union, Euro Union and North American Union. The above campaigns are also listed according to their difficulty from easiest to hardest. Each player/Alien begins with a set base value for their Mobility, Health, Attack, Gold and Score. As the game progresses, players can earn gold and score points from completing campaigns, then upgrade their equipment, or enhance their stats, using the gold they have earned.

**2. Reflection on SAD Proposal (See end of report for copy of timeline)**

This game generally meets the requirements that the SAD proposal has laid out. Almost all game features that we decided to implement is successfully coded into the game. From a developer’s standpoint, this game could potentially be called a Minimum Viable Product. It encompasses most of the function, does not include bugs and achieves the goals listed in the SAD proposal.

* **What aspects of the game design is not in the actual game**

While coding, we discovered that many of the functions we originally wished to implement into the game were not possible because of time constraint and technical difficulty. For example, in the SAD proposal, the enemies were split into many different Tiers with different stats (Attack, Health etc.). In addition, there should be bosses that would appear in the end of each campaign that also required a different set of stats. While coding, we discovered that it was extremely time consuming and risky to create this many classes and thus, we changed the overall theme of enemies. Now the game has only soldiers and tanks as the enemies. According to the level of difficulty of the campaign chosen, the number of soldiers and tanks would alter, presenting either more or less challenge to the player. Another aspect of the game that we originally planned to implement but was not able to was the progression of levels. This was not implemented because of time limits rather than technical issues. As a result, players are able to play any map they wish to at anytime, meaning one is able to start from the North American Campaign, which is the hardest. However, because of the level of difficulty, he/she has a very small chance of defeating the level, thus they are indirectly forced to start with easier campaigns, earn gold and upgrade their weapons/defense. If this component were to be coded, the main method that would be used is Boolean Algebra. A variable would be set that can only either be False or True. The variable becomes True after one level is completed and the following level would then be available. One feature of the game that we were not able to perfectly incorporate was the random spawning of enemies. We used random number generators to generate the position of the enemy. However, in doing so, we allowed the enemy to sometimes spawn inside an obstacle. We currently are not able to fix this bug and hope to improve it in the future.

* **Is everything completed according the original timeline?**

The progress of our program only has minor alterations compared to the proposed timeline of progress. The main change occurred in the debugging portion of the entire project. The proposed time for debugging was only four days and proved to be insufficient. The entire coding aspect of the program was actually finished two days ahead of schedule, thus leaving two extra days for debugging. The program turned out to be in need of many minor tweaking. For example, at certain points on obstacles in a game, the bullets would cause the program to crash if it hits those certain points. This is one major glitch in the game that took between one to two days to debug. It turns out that when creating obstacles through drawing rectangles on screen, there were some overlaps between two obstacles, thus the program would count two separate collisions if the bullet strikes the obstacles at the point of overlap and try to delete two bullets from the game, which is impossible since there is only one bullet colliding.

* **What went well/according to plan?**

The part that was coded the best was the data structure. The two main data structure we used are Dictionaries and Lists. For Dictionaries, it is used to store information such as the size and position of obstacles in a level, laboratory items and their stats (Attack Damage etc.) and more. As for lists, it is used to store information such as bullet attributes and position, and player information. Tuples were used for the position of almost everything in the game, such as the mouse position, the position of player and enemies. Also, tuples were used to store colours and are later used in many occasions such as printing text. In addition to data structure, another aspect of the program that was well coded was the functions. For actions that we would have to repeat multiple times such as printing text, testing if two objects have overlapped each other and other calculations, we created a function for it. Thus later on in the code, a four line code would be shortened to only a single line, which makes the program run much more smoothly.

**3. Overall Game analysis**

The game is overall coded quite well for a first version/prototype. From the playing experience, we think that this game would be more adaptive in the mobile game market. We believe that the game would be a huge success in the market as there are already games of this sort and were proven to be successful.

* **It is easy to play**

This game does not have sophisticated operating mechanics. The game guide appears to be two pages but the essence of it is just move, shoot and score points by killing enemies. The upgrading system is also straight-forward. Each attribute of the character can be enhanced through buying respective items that would power up the character. The objective of the game is also extremely simple - you shoot, you get to the objective point, you win the game. All aspects of the game are easy concepts for the player to grasp. Thus this game is simple but entertaining at the same time, making it appealing to players.

* **It has challenges**

This game progresses through the storyline the form of multiple campaigns. As the game develops, the levels and campaigns become harder, thus presenting challenges to the player. The only way that a player can get pass a level is through upgrading his/her equipment and have high stats, thus improving his/her performance. A game that is too easy does not interest the players but on the other hand, a game that is too hard will discourage many users from playing the game. Therefore, our game has obstacles that will be troublesome for the player, but not yet insurmountable.

* **Future Improvements**

Of course, this version of the game is far less from enough to be successful in the competitive gaming market. It would need major changes to make it appealing. Firstly, the graphics of the game should be improved. The player and enemies should display walking animations and shooting animations when them shoot and move. Huge portions of gamers purchase or play games because of the quality of the graphics in the game. In addition, this game will need a much larger map for players to move around. Having larger space to move gives the player more opportunities to outplay their enemy (dodging bullets etc.). The players will be much more willing to play on a larger map than to be crowded in a small, confined space. Another major improvement that is needed is the addition of more tiers of enemies. Having a diverse enemy pool keeps the game fresh and enticing. Not many people would be willing to play a game with only two different types of characters. Also, the attack speed of player and speed of the bullet turns out to be quite affecting the overall power of the player. Therefore, there should be items to upgrade the attackspeed and bullet speed in the labratory. Last but not least, the game should have more levels/campaigns. A rather lengthy storyline keeps the players interested and more motivated to play on.

* **Testing the game**

During the testing stage of the game, we first tried the program ourselves and found several bugs. We analyzed what caused the bug and tested our solutions. An example would be that the bullet cause the program to crush when it hits several obstacles during the battle. We analyzed the problem and found out the obstacles overlaps each other and causes the program to try to delete the bullet twice. So we fixed this problem by redefining the obstacles and make sure they do not overlap.

Then we invited multiple third parties to test our game. One suggestion we got from the third party is that we should regenerate the player’s health after the player lost a battle. Initially, we considered this as a lose and the player must quit and restart. The suggestion from the third party changed our mind. Another suggestion we got is to make the bullets not penerate the walls in a battle. We accepeted the suggestion. Overall, most third parties seem entertained in our game.

|  |  |
| --- | --- |
| **Due Dates/Timeline** | **Task** |
| April 10-20th | -Create rough designs for game  -Experiment necessary methods of creating game in pygame  - Create, edit, finishes SAD |
| **Due Date-** April 21st | Present SAD to Mr.Cope |
| April 22nd - April 27th | (Fix SAD if there are problems)  **Work on these for Progress Report:**  -Create all graphics that will be used in the game.  -A basic frame work of the game that includes the intro screen, menu,world map, Labratory, and help guide.  -Find sounds to be added to the game (for shooting, character moving etc)  -Finish Planning what to put in all four levels in the game (landscapes, how many enemies) |
| **Due Date:** April 28th | Present Oral Progress report |
| April 29th-May 10th | **Work on these for the next Progress Report:**  -Finish the basic coding for all 4 levels in the game.  -Sounds added to the game  -Score/gold system added to the game |
| **Due Date:** May 11th | Present the second progress report |
| May 12th - May 16th | **Work On:**  -Finishes coding the levels  -Test and debug errors in the program |
| May 16th - May 18th | -Create a User Documentation that describes the contents of software packages and detailed instruction relating to how the program is used.  - Get third party to test the game. |
| **Due Dates:** May 20th 11:30 pm. | Written Documentation and Program and documents submit to dropbox |