***Team A – Tank Wars***

***Prioritized Bug/Enhancement Report***

***September 5, 2014***

***John Aromando***

***Mike Bunger***

***Leah Austin***

***Jason Rivera***

**Overview**

After analyzing the data from the Program Review Report and Customer Requirements Report, the team looked at every bug that had been discovered and the recommendations given by the gamers and prioritized what should be worked on in what order. The team accomplished this by creating four tiers; urgent, high, medium and low. Afterwards, the team looked over the list of bugs and enhancements and decided in what order they should be done in and how much estimated time it will take to resolve the problem. Lastly, the team will describe basic strategies to guarantee overall team success and project completion.

**Criteria**

Urgent – Bugs that completely interrupt or disallow game play. These are crucial to producing a fully working game. Enhancements include changes that make the game fully operational.

High – Bugs that interfere with game play. These are bugs that change the game in way’s it should not, sometimes granting advantages to a player. Enhancements include changes that make the game fair and more enjoyable.

Medium – Bugs that inconvenience the player(s). These are minor bugs and pertain to things mostly trying to improve the actual game play. Enhancements include changes that increase user enjoyment and friendliness.

Low – Bugs that are small and may not make a noticeable difference if implemented. These are very minor and may only be changed if the team has the resources and desire too. Enhancements include small changes to the game that improve it slightly.

**Prioritized List ~ Summary**

**Time** – If it is a bug, it includes the time it takes to find the bug and to correct that that particular bug. If it is an enhancement, then it reflects the time it takes to implement the improvement.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Level | Priority | End Goal | Time | Members |
| Tank Moving off Screen | Urgent | 1 | Forcing the tanks to stay on screen or to return the tank to screen if it goes off. | 5 hours | 2 members |
| Power on Restart | Urgent | 2 | When the game restarts, the tanks are able to shot and do damage | 6 hours | 3 members |
| Space Bar | Urgent | 3 | Prevent the space bar from failing to shot and end the turn | 5 hours | 2 members |
| Document Code | High | 4 | Comment code and make it easy to read and work with | 8 hours | 4 members |
| Missed Turns | High | 5 | Turns will always switch between players without skipping | 7 hours | 3 members |
| Frozen Game | High | 6 | Prevent the game from locking up and forcing a restart | 5 hours | 2 members |
| Help/Tutorial | Medium | 7 | Making the help/tutorial more clear and useful to players | 1 hour | 1 member |
| Screen Transition | Medium | 8 | Smooth transitions between screens without going to the desktop | 4 hours | 3 members |
| Music Overlap | Medium | 9 | The music will only play when it should be playing | 2 hours | 1 member |
| Buttons | Low | 10 | Fix buttons that don’t work | 2 hours | 1 member |

**Prioritized List ~ Detail**

1. **Tank moving off screen** - Restarting the game while tank traverses terrain in the “win” screen causes the tank to continue to move in the new game, rolling off screen. Could not be restored by restart or influenced by the move buttons.
	* **Goal:** Develop a better method of keeping the tanks on the screen and have a way to retrieve the tanks should one of them go off screen.
	* **Time:** 2 hours to find the cause of the bug, 3 hours to develop a solution.
2. **Power on Restart -** Multiple instances of the weapons becoming ineffective after restarting the game.
	* **Goal:** Prevent the tanks’ weapons from becoming ineffective after the game restarts by changing how the game restarts
	* **Time:** 3 hours to find the cause of the bug, 3 hours to develop a solution.
3. **Space bar -** On three occasions, the space bar (fire) failed to work, and could not be restored by restart.
	* **Goal:** Guarantee that that space bar never fails to fire and if it does, there is a way to end the turn without firing so the game can continue.
	* **Time:** 2 hours to find the cause of the bug, 3 hours to develop a solution.
4. **Document code -** Lack of comments in the code, making bug identification and fixing more time intensive.
	* **Goal:** Have each member go through the code and add comments and become familiar with the code. Reorganize the code as a team if necessary.
	* **Time:**  2 hours for each member.
5. **Missed turns -** A number of situations led to one player’s turn skipping, or the ability to take two turns in succession.
	* **Goal:** Guarantee that not turns are ever missed by changing how turns are measured in the game.
	* **Time:** 4 hours to find the cause of the bug, 3 hours to develop a solution.
6. **Frozen game -** Game froze two times in testing, requiring forced exit of the program.
	* **Goal:** Guarantee that the game never freezes or stalls.
	* **Time:** 3 hours to find the cause of the bug, 2 hours to develop a solution.
7. **Help/Tutorial -** Customer Requirements Report identified ineffective Help/Tutorial. Customer failed to utilize all functions listed, including crucial aspects of gameplay.
	* **Goal:** Make the help and tutorial useful to player.
	* **Time:** 1 hour to update help/tutorial
8. **Screen Transition -** Screen transitions are jarring from start screen, help screen, or game itself. Program exits to desktop then transitions to next screen after several seconds.
	* **Goal:** Prevent the screen from return to the desktop during transtions.
	* **Time:** 2 hours to find the cause of the bug, 2 hours to develop a solution.
9. **Music Overlap -** Several situations bring overlapping music, particularly when winning/credits roll.
	* **Goal:** Guarantee that the music only plays when is meant to play.
	* **Time:** 1 hour to find the cause of the bug, 1 hour to develop a solution.
10. **Buttons -** Certain buttons do not work. Up and Down arrow keys are designated to the power functionality, which has been identified as “Q” and “E”.
	* **Goal:** Fix buttons that don’t work.
	* **Time:** 1 hour to update buttons

**Plan Moving Forward**

1. Make weekly back ups
	1. Every week, a working back up of the code will be made.
	2. If something happens where the game won’t run there is a way to go back to a working version.
2. Rotate members on certain issues
	1. Based on availability, members will rotate on and off defend parts of the project.
	2. All member will be allowed to work with each other.
	3. Provides a fresh perspective on a problem and brings new ideas for solving them.
3. Address new issues as they arise
	1. New problems are likely to appear throughout the devolvement process.
	2. Problems will documented and categorized based on the criteria above.
	3. Based on priority level, problems will be dealt with immediately or later on.
4. Stay organized
	1. All members are expected to document all work done and time spent on projects.
	2. All code will be commented thoroughly.
	3. The prioritized list and plan moving forward will be regularly updated to reflect progress.
5. Communicate
	1. Members will meet regularly both in person and online.
	2. Members must come to a consensus in regards to all project decisions.
	3. Members are expected to keep team informed of any new development, setback, completion of task, or anything in relation to the team or project.