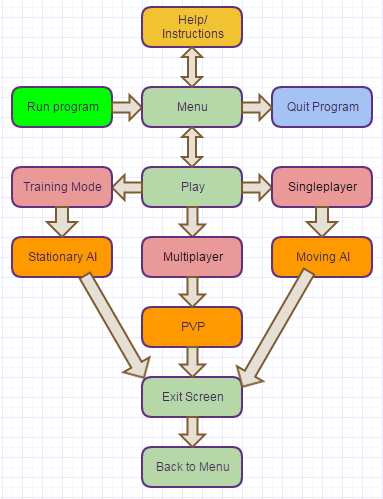
## GROUP 1 – COMBAT Report

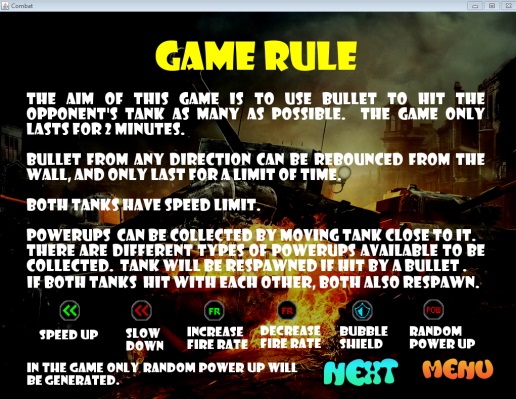
Introduction

Our project was to remake an old classic game (Combat by Atari) which would be suitable for 8 years olds to play. The game was developed using Java and we did revamp the overall designs which might seem more appealing for younger kids. The report will include what the requirements were and if they were met and if so, how they were met, top level view of our game system (states), how some major issues during development was overcome, special features which were added and any improvements which could made for future reference.

Requirements

Our developed system meets the requirements as soon as we run the program, it comes up to our first Menu page (with 1024 x 768 boundaries) where we could choose to play the game or read the instructions on which keys to use for playing the game. Then we can choose which player mode we could play in, which includes the three modes: Training mode, Single player and Multiplayer. Training mode is a single player mode where one player gets to practice on a stationary enemy, whereas Single player has a moving enemy. Multiplayer mode is a 2 player mode where we can locally go against someone else on the same computer. The pause function was perfectly fine as you can do so during the two minutes with no problem whatsoever using the ‘p’ button. While in pause, you can exit back to the menu with the esc button or resume game with the ‘p’ button. The tank will die and spawn again if it dies while the other tank gets the score. While in game, you can always go to the exit page right away by pressing the ‘PgDn’ button. Sounds are not be played in our game due to lack of time but we do have 3 power ups spawning up every 30 seconds for 10 seconds which contains different effects as discussed in lecture back in week 2. The tank detects walls but there will be some clearance pixels in between making it slightly off accurate but it is not as bad as it seems. The game starts after counting down to 3 seconds and then goes on for 2 minutes until the game ends and sends you to the exit page showing you who won and a phrase that goes with it and then you can return back to the Menu page to repeat.

Top-level view of system



HELP2

HELP1

MENU



Exit Screen

Play Options

Game

Significant issue while developing

1. During development, the most challenging issue was that it was really hard to meet up with each other for the first 4 weeks since Benjamin is Electrical, and the ELECTENG 310 team project (which kept us from progressing too much during those 4 weeks). Benjamin was always occupied by his EE 310 team and actually had to cancel 4 of our meetings due to the their team calling out Benjamin nearly every time we had some time to spare. To overcome this, we actually couldn’t do too much but to slowly work on it until Benjamin was done with his EE 310 project. It could have been slightly better if I (Jack) kick-started the project a bit more earlier rather than wasting most of the time waiting for Benjamin to finish off his first priority completely.
2. We had a hard time trying to understand each other’s code which eventually became quite an issue when we wanted to help one another by implementing on each other‘s code. This resulted in not being able to finish the game to a level we wanted it to be, hence we couldn’t really overcome this issue in the long run.

Special features/functionalities

1. We have very decent looking concept for the tank game as personally I think our design for all the states (menu, help, exit, game levels) are all quite well planned and it came out better than we expected. All the pictures for the tanks, bullet, walls and power-ups were made by us on Gimp (software similar to Adobe Photoshop) and the background pictures were taken for the internet.
2. We have quite a number of different level designs and they will really be liked by 8 year old children as the designs were designed to prefer younger aged players as they are the target for this game.
3. The help page was added to show the players the keys to operate the tanks in the single or multiplayer mode and briefly explaining the game rules and certain features in the game. This page was created to give the players a better understanding of how the system was built and to prevent and possible confusion about the game or the keys involved during gameplay.
4. The UI can be navigated solely by the keyboard to make it less confusing and the selected modes will turn color to aqua before you press ‘enter’ on it to move onto the next game state (e.g. Menu 🡪 Play; like in the screenshots in the 2nd page). This makes the UI very simple to navigate and also using the obvious keys (arrow keys + enter key) just like most other games so the player does not feel awkward navigating through the game states. The ‘buttons’ throughout the game is big and clear so it is very easily to read and it will prevent unwanted confusions.
5. The background pictures are only called in the beginning of each state ONCE as importing it into the game every render cycle will significantly decrease the fps as you would be trying to import a 1024 x 768 picture 30 times a second; which is totally unnecessary. This will be able to keep the game going without too many delays or lagging occurring during gameplay or navigating through the UI.

Comments about tools used

We used Java to implement our entire game and I think it was a good language to use, as it is one of the most powerful languages out there, which is commonly used for making applications and games for PC and Android smartphones. It was very tricky to learn how to use Java at first especially where most of the game development programming was to be self-learnt since it wasn’t covered to the depth we needed in lectures. We used Java Swing over JavaFX as we thought it was a more suitable option for us beginners and also it wasn’t necessary to use JavaFX since the game was not demanding for extra features which cannot be done using Java Swing.

Improvements for future development/reference

1. Starting early with the project was the key point to successfully finishing the game on the level where you would be satisfied.
2. Good communication with your partner is also a key point which cannot be ignored as minimal communication with your group members could lead to possible confusion or one being not satisfied with the final outcome. Just to add here, meeting a partner with similar goals and overall schedule is also recommended as the teamwork could be much better with those being closely matched between the two.
3. Self-learning skills cannot be forgotten as it is one of the key to success and if both partners put enough time to self-learn the needed skills to complete the project, finishing the project could be done with more ease. Programming is all about understanding the concept of coding for any use and once you understand it, it would be the most valuable skill you would learn as a programmer.
4. Setting priorities was one of the things we really should have thought about before realizing our project will not be up to our initial standards, and we learnt that the hard way. We realized that we didn’t prioritize the project as much as we were supposed to and it was too late set it as top priority 4 weeks into the semester.
5. Asking for help when we were stuck could have been a much better choice than working on it for hours by ourselves. It is good to try work things out yourself but when that starts to build stress for you; it seems to negatively affect you, which could lead to procrastination and so on.

Group Number: 1 (Seung Bin Choi and Wan Chung Tong)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Features | Yes/No? | Comments | | Team members |
| All minimum specifications = 50% | | | | | |
| 0 | Compiles and runs fine without errors/Code quality - comments, indenting, etc. | Yes | No problems | Choi: 50%  Tong: 50% | |
| 1 | Welcome screen: select a game mode using keyboard, three game modes | Yes | No problems | Choi: 50%  Tong: 50% | |
| 2 | Start game: stationary tanks, countdown timer from 3, tanks should not be able to move | Yes | No problems | Choi: 50%  Tong: 50% | |
| 3 | Objects should not exceed 1024x768 boundaries | Yes | No problems | Choi: 50%  Tong: 50% | |
| 4 | Hit registered when shot collides with enemy, event(s) follow | Yes/No | The detection is no problem but the action that comes after the collision detection is not really working so well... Had a lot of problems and errors for a long time. Tank-Wall is the only working collision event at the moment. | Choi: 50%  Tong: 50% | |
| 5 | Score and time displayed on screen | Yes | No problems | Choi: 50%  Tong: 50% | |
| 6 | Game can be paused/resumed with ‘p’, exited with ‘Esc’ back to main screen | Yes | No problems | Choi: 50%  Tong: 50% | |
| 7 | Exit screen at end of game with summary, PgDn to skip to exit screen | Yes | No problems | Choi: 50%  Tong: 50% | |
| 8 | Appropriate sounds for shots and other collisions | No | We didn’t have enough time to implement sounds due to other game logic issues holding us back. | Choi: 50%  Tong: 50% | |
| 9 | Log file generated for each game, with time stamped details of each keyboard action | No | Same reason as above in why we couldn’t generate a time stamped log file of keyboard actions. | Choi: 50%  Tong: 50% | |
| 10 | Objects (including powerups) as listed in Game Rules v2 | Yes | No problems generating them | Choi: 50%  Tong: 50% | |
| 11 | Object Interactions as listed in Game Rules v2 | Yes/No | Some is working but some isn’t. Tank-Wall works but the other don’t really work.. Our biggest flaw in the project. | Choi: 50%  Tong: 50% | |
| 12 | Object Control as stated in Game Rules v2 | Yes | No problems as the tank moves around using Tank Control; side arrow keys rotate the tank instead of moving sideways. | Choi: 50%  Tong: 50% | |
| Design Elements (worth 50%) | | | | | |
| 1 | Menu/Help Page advanced features | Yes | Personally made the help page and all the ‘buttons’ in the entire game which can be controlled by keyboard actions. | Choi: 50%  Tong: 50% | |
| 2 | Objects images | Yes | We did not use any online images for the objects as they were all personally made by us using GIMP. | Choi: 50%  Tong: 50% | |
| 3 | Level Designs | Yes | The levels were made using a decoder to make maps suitable for the 8 year old children (our target audience). | Choi: 50%  Tong: 50% | |
| 4 | Fancy titles/Word ART | Yes | All the titles were made using GIMP and they were imported into the project to gain attention from players. | Choi: 50%  Tong: 50% | |
| 5 | Threading | Yes | Threading was used so it could be implemented on our timers and our pause function etc. | Choi: 50%  Tong: 50% | |