CS101 PROJECTS 2014

BATTLE OF THE TANKS

Abhishek Majumdar(group leader)	140110082
Anagh Trivedi	14D170019
Ritu Jhare	140010032
Pooja Doifode	140110039

PROBLEM STATEMENT

- To make a Graphical User Interface capable of handling events like mouse clicks and registering them in the memory .

- Using of Graphics to design the components of the game i.e. The Tanks , Terrain and the missiles .

- Designing our own functions for collision detection , point calculation after a “hit”.

- Simulate a flying projectile by moving an object frame by frame .

PROJECT PLAN

- The first step was to come up with a method to align the flow of control depending upon the click (i.e. Executing a specific task on getting a click on a specific area .
- Then we proceeded by making the tanks and taking the input of power and fire graphically.
- After that the firing mechanism was implemented and point calculation was done .
- Finally the terrain was inserted and relevant functions (breaking of terrain and collision detection with terrain were done) .

The work distribution was as follows :

Overall implementation , Firing mechanism , Collision detection : Abhishek and Pooja

Terrain generation , Main menu : Anagh and Ritu

Critical tasks : Terrain generation , Collision detection .

INNOVATION AND CHALLENGE

CHALLENGES :

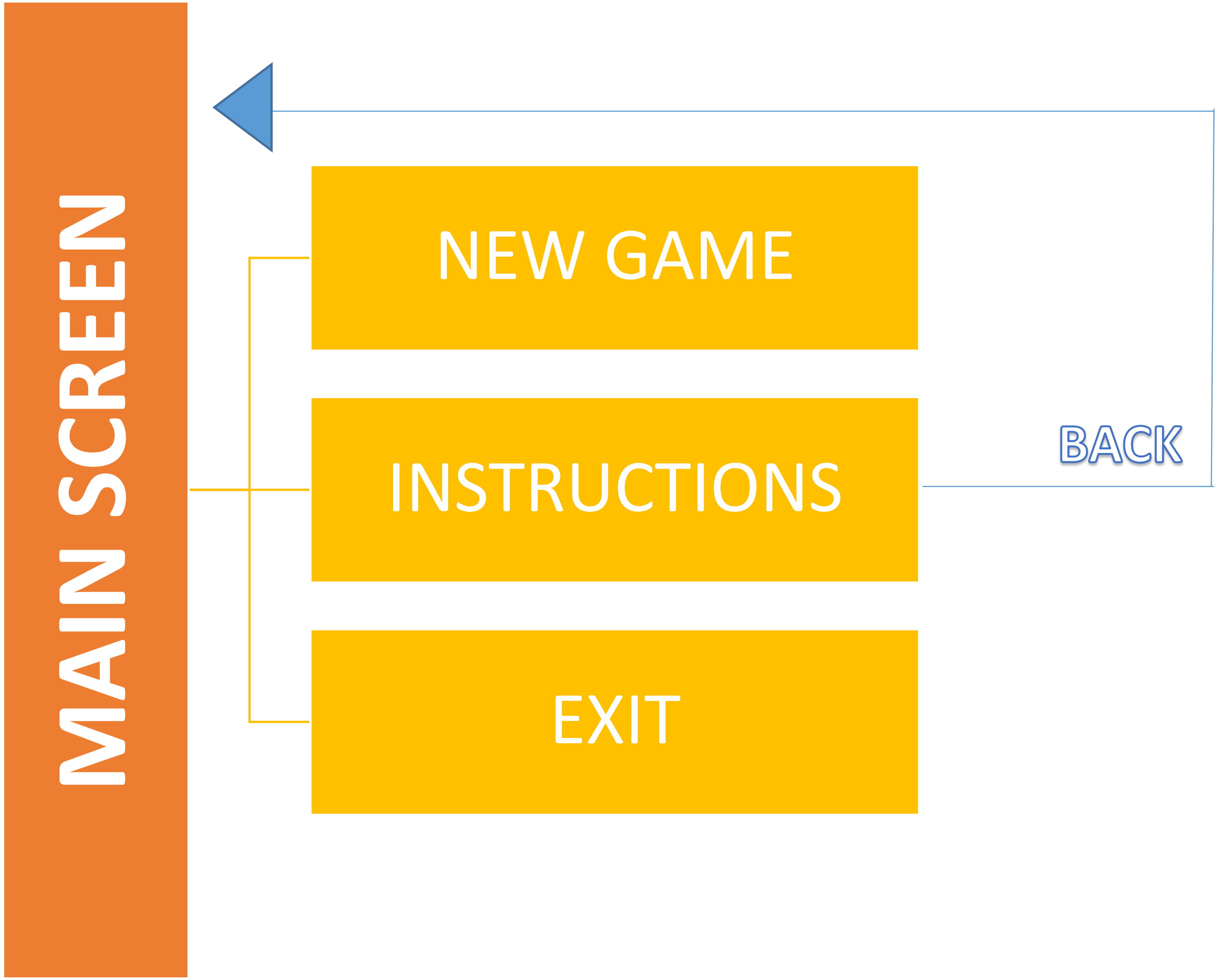
- The terrain generation was a big challenge due to the limitations of the simplecpp library (it crashes if given to handle very large graphics) .
- Collision detection took a long time to implement .
- Since there were no inbuilt functions for event handling , we had to implement all functions by ourselves .
- Taking input from the Canvas required quite a bit of thinking and was finally implemented using Coordinate system .

INNOVATION :

- There is a random wind factor in the game which makes it interesting .
- Instead of Health , a point system has been implemented . The points are awarded on the basis of how close a “Hit” is .

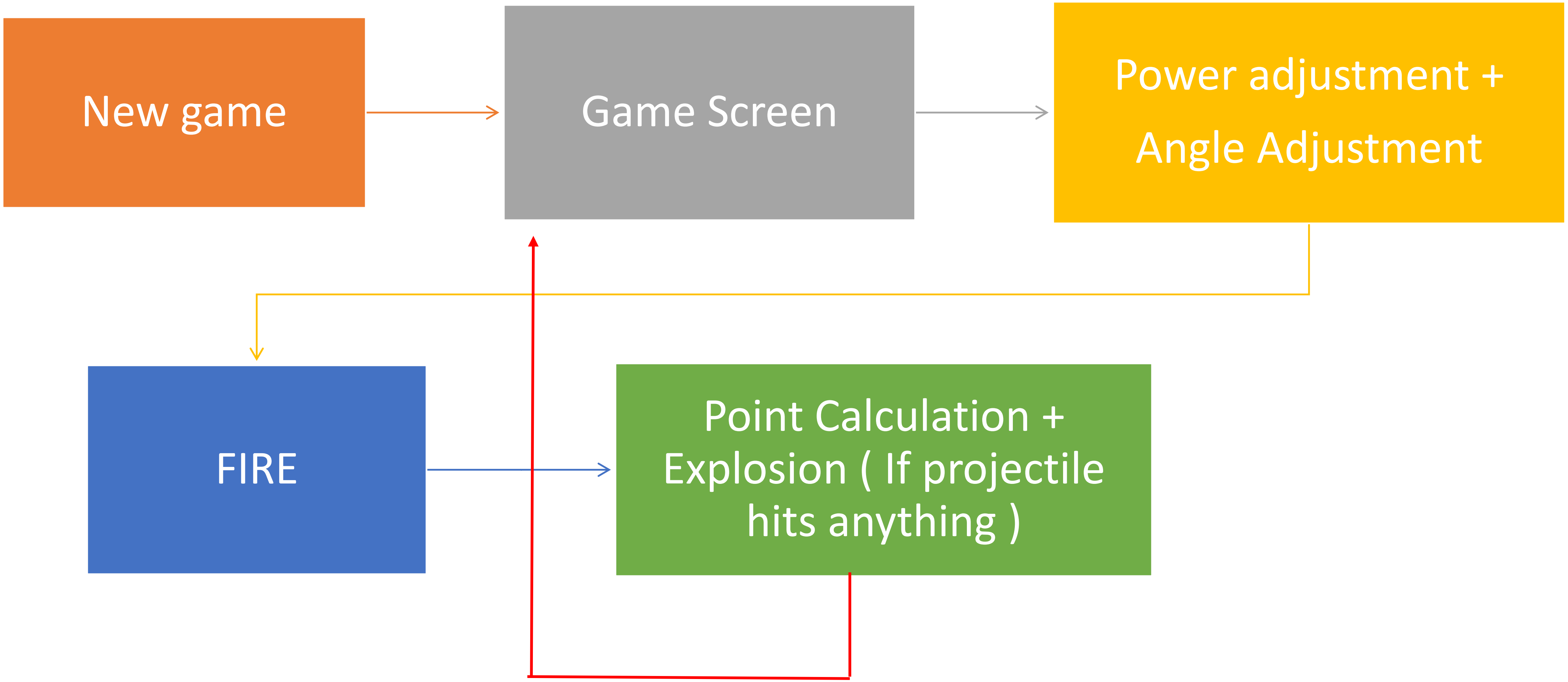
FORMAL SYSTEM DIAGRAM

MAIN SCREEN



FORMAL SYSTEM DIAGRAM

UNDER NEW GAME



TASKS COMPLETED

S.NO	TASK	PROBLEMS	SOLUTION
1)	Implementation of GUI	Registering Clicks	Compared the X and Y coordinate to an area and used conditional execution for that area .
2)	Taking input from the GUI	No provision of textual input in CANVAS	Made a rectangle whose width changes upon clicking . Scaled that width value to input .
3)	Collision Detection	No function to detect if objects are overlapping	Ran the projectile frame by frame and at every frame checked whether its center is coinciding with and existing object i.e. Terrain .
4)	Point calculation	Detecting the closeness of a blast	Calculated the distance of the blast from the tank using the coordinates of centre of the missile .

TESTING

In a **GRAPHICAL INTERFACE** , not much testing is required as you can always time to time run the modules independently and see if they are working as expected or not.

CRITERIA & DESCRIPTION : We consoled out the power , angle and points to check if the graphical inputs are working as exected or not . The Criteria was to check those values against the logic formulated and see if any error occurs .

RESULT : The parameters were working properly as concluded by the following observations :

- The values followed trends shown by graphical inputs i.e. increased when the width of the bars were increased and vice versa .
- The points varied according to how close the projectile hits and full points were awarded when the shot landed on the tank .

REUSABILITY FEATURE

The overall code is very Lucid and easy to further develop upon . The code has been made as modular as possible with functions for every action . The basic structure is quite simple and can be taken as a working basis by someone who wants to further develop the game . Following are the prominent reusable features :

- The firing mechanism .
- Terrain generation (though a possibility of enhancing it exists and would be suggested)
- Point calculation .

The User who wishes to further develop the game should have SIMPLECPP library integrated with his IDE .

FUTURE ENHANCEMENTS

- The most important enhancement to be done is development of an extended weaponary which we couldn’t do due to time constraint .
- A better and more interactive terrain can be made (Using advanced methods like Perlin noise generation) .
- The tanks can be given more features i.e. Rocket pads et cetera .
- New backgrounds can be implemented . (Though a big drawback of simplecpp is it dosen’t provide IMAGE LOADING facility) .