Andriy Yavorskyy & Tad Edwards

CS 1302

April 9, 2014

Reflection

Our arcade page which will always appear in the center of the user's screen has two gifs that are buttons. When you click on whichever you wish to play it takes you to the game. On both game pages you can refresh the game as well as go back to the main menu. Tic-Tac-Toe is standard no extra features were added except what was required. When you click on a block that has already been selected a pop up appears saying that you cannot click on that box and you should choose another. The undo button undo's the user's previous move. It pops their move from the stack as well. At the bottom of the Tic-Tac-Toe screen it says whose turn it is. We thought this was a good feature that helps the users determine who's turn it is if they have a mind blank. We also have winning and tying animations. After a game the user can click new game and it load a fresh game. They can click new game at any time. The code is well designed because the user can't mess the game up in any way. Going back to the main menu in the middle of a game, etc. won't crash the game. When the player goes back to main menu the scores are reset. We always reset the board when the user goes back to main menu. We used JLabels and JButtons to display almost all of the content. It's well designed because mostly everything is in the frame, having multiple dialog boxes annoys users and slows down game-play. Also our graphics all fit a central space theme so it is pleasing to the eye. Our scorched earth doesn't have a landscape besides a huge barrier in the middle. We thought this would be a nice change to the game. The "tanks" which are space ships in our game just float in the air. We decided to make both games humorous in their own way so that users would be encouraged to share our program and to add to the enjoyment of the game. The instructions for both games are located within their respective panels and they will be displayed only if the user clicks the button labeled "instructions" within each game. Similarly, the instructions will not go away unless the user selects "Hide Instructions". We felt it was a good idea to do this so that if a user forgets how to play while in the midst of a game, the instructions can easily be located and referenced.

Some troubles we had while programming our project 4 arcade were mostly due to our inexperience using JPanels and painting objects to the screen. Such as getting the ammunition each ship fires to display correctly as their values were updated. Initially we used loops thinking this would be the solution, however we found no results this way and resorted to using timers to update and display the ammunition firing across the screen.

For TicTacToe we had the entire game programmed within that one panel. In hindsight we likely should have split up the code for that game within multiple classes to make our code more readable/manageable. In fact, we learned our lesson before moving on to scorched earth where we put our physics engine, the information stored in the queues, and the actual panel itself in separate classes. This made all the difference to us when it came to managing our code more effectively. Hope you enjoy our games!