

Robert Matthews
8103636840
Matthew Vollkommer
8102494122
P4 Reflection

Compile and Run:

make
compiles the files

make run
runs the program

make clean
cleans the class files

Scorched Earth Instructions:

Archers! Instructions:

In Archers!, two archers face off against each other.
The archer that hits the opposing archer 5 times first wins the game.
Each archer has 5 health that decreases by 1 each time they are hit.
Each game consists of 5 rounds.
Each round consists of 5 turns per archer.
Each archer will take turn inputting the angle and power of their shot.
Player 1 will put in his angle and power first using the sliders, then click the next button.
Player 2 will then use the sliders to enter the power and angle of his shot and click next.
This will happen 5 times and on the 5th time, Player 2 will click play.
The Round will then be played updating as it goes along.
If someone wins during the round, the round and game will immediately end.
You will be asked if you want to play again or quit.
If no one wins during the round, another round will begin.
You will then enter new shot variable just like before.
If no one wins after 5 rounds, the game is a tie.
You will then be asked if you want to play again or not.
If the Konami Code is Entered in Windows on the Main Menu, you will
be given the opportunity to press a button to dodge the shots in one round.
Do not use it multiple times or you are a cheater.

Tic Tac Toe Instructions:

PVP:

Click in the square you would like to draw your shape.
Turns alternate between X and O.
3 in a row is a win

PVRandomCPU:

Click on the Square you would like to draw your shape.

Turns alternate between X and O.

Click to make the cpu choose a position on the board.

If the CPU chooses an already selected position, you will be notified.

Keep clicking until the computer chooses a place that has not been selected.

There is not an infinite loop. It picks randomly, so at end game it can take

Several clicks.

3 in a row is a win

Design Reflections:

Overview:

For project 4, we, Matthew Vollkommer and Robert Matthews had to design and code a game of Tic Tac Toe and a version of Scorched Earth. In these games we had to use stacks, queues or both to go back and forth for the turns. In Tic Tac Toe, each turn was stored in a stack and we used this stack to be able to backtrack with control z. In Scorched Earth, a queue was used to store all of the move information for the round, 5 turns per player, and was then displayed after all of the moves were inputted. We designed our Tic Tac Toe game to fit the style of the original. Simple X's and O's going back and forth until the board is filled or someone wins. The board was drawn simply as a picture that we placed as the size of the entire panel. The X's and O's were drawn using the Graphics2D class. We did make more changes to the Scorched Earth game. For our version of Scorched Earth, Archers!, we replaced the tanks with 2 archers firing at each other. There are 2 archers on either side of a wall, which is in the center of the panel. They take 5 turns each per round until someone wins, or 5 rounds are played.

Main Menu:

- * The main menu includes 3 buttons. A button for, Archers!, Tic Tac Toe, and instructions for Scorched Earth.
- * If the instructions button is clicked, a new pop out frame is created so that the user may look at it while playing the game.\
- * The menu is simple, yet attractive and easy to use, so we believe that it is designed well.
- * The buttons are actually JLabels that highlight red if they are hovered over to indicate that if the mouse is clicked at that point, the JLabel will be pressed. This makes our menu intuitive as well.

Tic Tac Toe

- * Our Tic Tac Toe game is exactly what you would expect to see, it's a board with the 3X3 grid of squares in which you click in and either a O or X is printed.

- * After the game ends, a dialog box appears asking if you would like to play again, or exit.

- * The turns and wins and losses are kept up with in the frame title so it is always available to see.

Archers!:

Layout

- * Our Layout is simple and intuitive.

- * There are 3 panels organized vertically.

Information Panel

- * The Information is at the top like a scoreboard, which is where most people look for information while playing games.

- * Player 1 is on the left side of the screen, so the information about his health and wins are above him on the left side of the screen.

- * The Round is indicated at the top center of the screen, where it is large and easy to see.

- * When a player is hit with an arrow, their health is updated immediately to show that they have taken damage.

- * The panel is symmetrical and well labeled. It would be very hard to get confused as to who is winning and where in the game you are.

Game Panel:

- * Our game panel is simple and attractive

- * We place 2 archers on either side of a wall pointing up, indicating that you must shoot over the wall.

- * If the wall is hit with an arrow, the arrow sticks in the wall for the length of the time that you play the game.

- * The arrow also stays when a player is hit to indicate who has been hit.
- * The archers stay in the same location for the length of the game. We believe that if their locations changed between rounds. The game would be too difficult.
- * There's grass beneath the archer, which we believe is a nice touch

Turns Panel

- * 2 simple and intuitive slider that are labeled clearly.
- * The next button displays “Next” until the 5th turn of P2, when it displays “Play”
- * This button is red when hovered over, like the main menu.
- * The angle goes from 0-90 degrees because there is no way to hit the opposing player with degrees outside of that range.
- * The power is adjusted to be reasonable with the size of the board.
- * Whose ever turn it is is clearly indicated, as well as which turn from 1-5.
- * The sliders and button may not be pressed while the round is being displayed

Archers General

- * The round and game play is all recursive from a mousePressed method that is connected to the next button. This is because power and angle is taken in when the mouse is pressed.
- * If the game ends the winner is indicated as well as if it ends in a tie and the players are asked if they want to play again.

Code Specifics

- * We used cardLayout, so that we would be able to switch between panels from within a frame seamlessly.
- * We organized information by panel so it would be easy to understand and keep up with.
- * We had many methods with one specific purpose so that we knew exactly what each method's purpose was.
- * We had no static variables.

TicTacToeGame:

Our game has all private class variables. It can run independently of all other class files in our program. We recursively call stacks and other methods throughout the game. The program keeps track of who is where with a really simple 2D array.

Each move is stored in a stack. Each stack is called when a move is undone. It is also called at the end of the game and cleared to clear the board for a new game.

Code Reusability

TicTacMenu, ScorchedKonami, and CPUTicTacToe show how reusable our dynamic code is. They take the code from the other related classes with only a few Statements changed in each.

The Round class is used to store moves by both the TicTacToe games and the scorched earth games. X and Y by tic tac toe. Power and angle for scorched earth.

TheInstructions class can run independently from the rest of the code. I regularly use it in different assignments in different courses while changing few statements.