Tic tac toe presentation script

Slide 1: Parth Hello everyone. Today, we will be presenting you a game that has been referred to as the most classic game of all time.

Slide 2: Parth So, to begin, where do you think people would have played this game?

Slide 3: Jinesh Well, they played tic tac toe during war when they were in trenches. This was because they didn’t have electronics like us. Tic tac toe was also played in school where children would get bored. Remember when we would open the last page of our note books and all we find is all these games? It is also played by people bored at home who are tired of tv. Lastly it is played in sleepover parties for young kids.

Slide 4: vragang That was all fine. But what the tech nerds of the time were trying to do is find a way where 1 person alone can play this and many other games. That is when we came up with the word algorithm. An algorithm is a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer. So basically, it is where the computer knows exactly where to make the next move. And not any move, the perfect move!!

Slide 5: Vragang So, we have created our own version of this game.

Slide 6:Parth So, what exactly did we want? Well anyone can make a computer make a move, by using the math.random method, but what we wanted was to create a computer beast. We wanted to make the computer player in a way that it could never lose. So, we did. We call him the perfect player. Meaning he either wins or draws. He never loses. We also wanted to create a game with great graphics, unlike the one in the diagram that was given to us about our summative assignment. We wanted to go beyond and create something amazing that simulated Google’s game. We also wanted an intro page, exit page. And last but not least, we wanted it to be physically appealing.

Slide 7:Jinesh Assumptions. At first, we assumed that if we want to make a perfect player for the computer, then we must find out every single move that the player can make. In simple terms, all the possible boards that can exist in the world and we must program our game with them all. There turned out to be over 200,000 different board patterns. Then we realized that there must be an easier way to do this. And to our surprise there was. We will show you guys later in the presentation. We also assumed if we wanted to make the best of the best, it would take time and thought. So, we decided to not waste any time and work hard every day. We spent lunch hours thinking of algorithms that would satisfy our needs. We also assumed that since this is a proper game, it will take a lot of space in the end.

Slide 8: Jinesh The process

Slide 9: Vargang First we created the methods to animate the x, the o, and the board itself.

Slide 10: Parth Then we created the intro and exit page animations.

Slide 11:Parth After that, we created the results page and the would you like to play again page. In the play again page, we just created a loop for it to continue on and on until the player inputted a ‘N’ for no.

Slide 12:JineshAlgorithm. This was what we made after all the graphic work.

Slide 13: vragang First we created a method that figured out if there were any moves that the computer could make to win. In other words, is there 2 o’s in a row. That is what the computer checked for. Because if there is, then the computer can put it there and it can win. So, it returns a array of size 3. In the first index, it holds if it can win (1 = yes, 0=no), in the second and third index it holds the row and column for the computer to move into.

Slide 14:Parth Then we created a method to see if there are any places that the player can put to win. This is like the method for the checking the computer move, but it is for the player. We find out if there is any way that the player can move, and we block it by putting the computer’s move there later. So, this method returns an array. In the first index, there is 1 for yes, the player has a way to win, or 0 , no the player doesn’t have a way to win.

Slide 15: Vragang Then we import those methods into the method called perfect move and store them in a local array. In this we first see if there is a way for the computer to win, and if there is we automatically do it. But if there isn’t , then we see if we can defend, and if there isn’t a way for anyone to win, then we assign a random number to the row and column. Then we see for double traps. For example, if the player draws 2 x’s , one on the top left corner, and one on the bottom right corner, then we place the computer move there. We will explain everything in detail when we show you our code, because these are pictures, so you can’t understand much. And after all this, we added some additional features and put it all together.

Slide 16: Vragang Now we will show you a breakdown of our 1000-line code and what everything means. (Press esc, open code, and explain every segment.)(Use high level plan for support)(Also use pictures (flowchart by vrajang).

Slide 17:Jinesh So initially we created this game for 1 person only, but then we also created a small surprise to show you in the end. It is designed for player vs computer only!!! This means only 1 person can play this game at a time. During our development, we created many small java classes that held different stuff. Such as we created one with just the intro, one with just the conclusion, one with the x, the o, and so on. Then we combined it all into 1.

Slide 18:Parth So, we approximately pressed that run button just over 600 times , me, Jinesh, and vrajang combined. Because we all made various parts. Aside from the graphics, our code was usually without a flaw. We rarely had to fix up the numbers and stuff. We just had trouble in the graphics because we were looking for something particular and we had to keep on running it to see how the new graphic turned out. We know our code works, because in our free time, we went around asking people to test our game, and they gave us positive feedback.

Slide 19:Jinesh Extra features.

Slide 20:Everyone One feature we had was animation. This is great because if we were to publish our game, then people would want to play it as it looks nice because it is animated. We also have an intro page and an exit page, this makes the player feel welcomed. We also have an auto-exit meaning that the app will close as soon as the player says he doesn’t want to play anymore. It will save the user some time in the end making him say something positive about our game. We also have all inputs as type char. This is great because it will save the player time as the computer goes to work right after the player hits the key he/she wants to play. If we used string or int, then it would be less efficient as the player would have to press enter every time. We also have 3 modes of the game. Easy medium and hard. This gives the player an option of what type of match he/she would like to play. A feature we would like to include would be to have an exit button in the corner at all times. When the player presses a key on the board it will close the game. We would also like to include a feature that allows the player to go to the home screen whenever they want to. Just in case they change their mind and want to play an easier or harder level.

Slide 21:Parth lets play a few rounds of tic tac toe.

Slide 22: Everyone Although we feel that our game is great the way it is, but we feel we do need some improvements. We think we should use something like unity or flash. And it turns out that unity also supports java, so language won’t be an issue. These developer engines are much faster to use as they reduce the time wasted on graphics as you can quickly draw them. Also, you can import pictures easily and not have to draw them. Another change we would like to do is make the game clickable, this is because we don’t want our player to be using the keyboard to enter the location, we want them to tap on the area where they want to click, and it should work.

Slide 23: Anybody Thank you for watching, and we hope you liked our tic tac toe game.