Project 2

Abstract

a program that plays this questions game with the user and learns from its mistakes, building a database of questions and guesses along the way. The program is *persistent* in that it remembers its database between subsequent runs of the program. This way, you can *train* your program to recognize more and more objects over time by holding data and going back to it to tell your object by yess and no question. The Learning Genie will hold information in a text file when the program is running so itll know more objects each time you start the game.

Introduction

Learning genie will understand user answer and gain AI and will be better and better as game goes on. However, unlike a human that chooses their questions with care, these applications “learn” from all the games it has played in order to ask questions. At the end of each game, if the application makes an incorrect guess, it asks the user for a discriminating question that distinguishes between the application’s guess and the user’s intended object. The application then stores this and uses this question in future iterations of the game. I plan on making this game just keep running until the user plans on stopping the game. Even if the game wins it will prompt the user to say yes or no to keep going. It will use decision trees and nodes to make this work and there will be text file to hold data as game goes on. This game will just keep running until the user plans on stopping the game. Even if the game wins it will prompt the user to say yes or no to keep going. It will use decision trees and nodes to make this work and there will be text file to hold data as game goes on. The text file that holds it will save so the game will not reset unless you want it to so that the more you play the more it'll learn and the more it knows the longer the game will go on and it'll learn more objects.

Detailed System Description.

Thee code for this system all relies on the other pages to gather its information. The main method grabs information from all the nodes. The main method gets the amount of objects in the tree the questions it should ask and the answers. But all these nodes need to be able to get there information somewhere and store further information it gets like new objects and new questions it gains from more more and more playing. The nodes store and get their information from a decision tree and the the text file that it writes to. All the nodes have a different purpose on what to gather it is pretty clear by the name of each node what it is gathering and storing to the DecisionTree which are the DecisionNode, QuestionNode, and GuessNode. The decision node will get the proper guess for the main method while the question node and the guess node will get the proper answer for or question for the learning genie to ask. All the nodes who feed and get information the decision tree, at first i tried to make one node but i could not follow my work so i found it easier to make more nodes.

Requirements

The reason for this project is to make a game to keep people entertained. The game is a 20 question esk game which people play when they are bored and want to stay entertained.

Literature Survey

There have been many games that have created for this purpose of staying entertained. This game is simple and there are many like it. It is a guessing game like Akinator, 20q, and Animal Game behave in a similar manner. You, the user, think of a thing and the application asks you a series of questions until it feels like it make a guess.

User manual

The learning genie will keep asking you questions as the game goes you will need to answer them truly because if you ever answer a question wrong it will mess up for future games. The learning genie will ask you an object once it runs out of questions to ask then it'll will prompt you to answer if it is correct. If the learning genie is correct it'll will win but if it does not get your object it'll ask you to give a clarifying question that distinguishes your object from its guess.