Isabelle Chun, Jannitta Yao, and SIlvia Zeamer
Project name: Choose Your Own Wellesley Adventure
CS 230 Spring 2018

Final Project: Technical Report

Our project consists of a GUI with a graph to represent a map of the Wellesley campus, a Linked Binary Tree to represent the different choices and results from the Choose Your Own Adventure quiz, and a Queue of Places to track the places a user visits through the quiz.

First, the Place class contains three String instance variables for a map location's name, a short description or fun fact about it, and a jpeg file name for the associated photo in the Photographs folder of our directory. This class has getter methods (the toString serves as the getter for name), samePlace(Place p) which checks if two Places are the same based on their names, and a main method for testing.

The WellesleyMap class is made up of an AdjListsGraphPlus graph of Places called map, a LinkedList<Place> placeList, and a Place instance variable currentPlace. Its constructor is parameter-less and calls a helper method fillMap. The AdjListsGraphPlus<Place> map is unweighted and undirected.

WellesleyMap has methods fillMap(), getVertex(String placeName), setCurrentPlace(String placeName), getCurrentPlace(), getPlaces(), requestNeighbors(Place vertex), toString(), and a main method to test WellesleyMap methods.
fillMap() scans the text files place_list.txt and arcs_list.txt to add Place vertices and edges to the graph and accumulate a list of vertices (Places) as placeList. It also sets the currentPlace to the Place named "Your Dorm (Stone D)" since the quiz begins with the user in their dorm.
getVertex(String placeName) returns the Place from the graph with the name of the placeName input, or null if that Place does not exist in the graph.
setCurrentPlace(String placeName) calls getVertex(placeName) to update the currentPlace of WellesleyMap.
getCurrentPlace() returns the currentPlace instance variable.
getPlaces() returns a LinkedList<Place> of all of map's vertices.
requestNeighbors(Place vertex) calls getSuccessors on the graph to return a list of Places with adjacent to the vertex input.
toString() returns the map's toString along with the currentPlace.

For Adventure mode, our program uses two classes: Question and ChooseYourOwnAdventure.

The Question class has three String instance variables for the question, the left answer option, and the right answer option; and two Place instance variables for the Place associated with the left answer, and the Place associated with the right answer. The Question constructor simply sets each of its five inputs as the five instance variables.

Question methods include getters for each of these five instance variables, getQuestion, getLeftPlace(), getRightPlace(), getLeftAnswer(), getRightAnswer(), isLeaf(), and a main method to test Question methods.
isLeaf() returns true if the Question's left answer and right answer are both empty strings. It's used in ChooseYourOwnAdventure.

The ChooseYourOwnAdventure class has four instance variables: a LinkedBinaryTree of Question objects that sets up the LinkedBinaryTree of ChooseYourOwnAdventure, a LinkedBinaryTree of Question objects that is modified as the user plays the Adventure mode, a Question of the current question (the root of the subtree), and an ArrayQueue of object Places that is used to keep track of where the user has been throughout the day. The ChooseYourOwnAdventure constructor creates a new ArrayQueue of Places objects, and also creates the LinkedBinaryTree of the Adventure mode.

ChooseYourOwnAdventure has methods addPlace(Place p), containsPlace(Place p), getQueue(), getCurrentQuestion(), answerQuestion(String dir), and a main method to test the methods in the class.
addPlace(Place p) adds the specified Place p to the Queue of Places .
containsPlace(Place p) checks to see if a Place p is already in the Queue of Places, and returns a boolean.
getQueue() returns the Queue of Places that the user has traveled to.
getCurrentQuestion() returns the current question being asked to the user
answerQuestion(String dir) takes in input from the GUI, and traverses the BinaryTree so that the user can play ChooseYourOwnAdventure.

The WellesleyAdventure class is the class which unites all of the other classes used in the program into a cohesive whole. The WellesleyGUI's main method creates a new instance of the WellesleyAdventure class, which extends JFrame and contains a variety of custom JPanels. WellesleyAdventure integrates the backend classes with the GUI by adding action listeners to their buttons and defining methods which these action listeners call. In addition to a no-parameter constructor, it has the helper method initUI(), which sets up the user interface inside the WellesleyAdventure JFrame, update(), addPlaceListeners(), getNav(), goHome(), goToMap(), returnToPlace(), goToAbout(), and goToQuiz(), which control the contents of the screen and are called in response to input by the user.

The WellesleyAdventure class also contains three implementations of ActionListener, NavListener, PlaceListener, and QuizListener, each of which contains a single method, actionPerformed, which responds to clicks on the buttons which control the program's functionality.

A number of custom-defined JPanel classes make up the user interface defined by WellesleyAdventure, in order to create a cohesive appearance for the user interface, and to encapsulate the particular needs of its desired functions. These classes are MapPanel, PlacePanel, QuestionPanel, ImagePanel, and NavBar.

MapPanel contains a map of the Wellesley campus and a grid of buttons which take the user to PlacePanels that display photographs of locations on campus, the name of the place, a fun fact about the place, and a list of buttons leading to adjacent locations. It has a

no-parameter constructor and a method getLinks which returns a list of the JButtons it contains so that ActionListeners can be added in the WellesleyAdventure panel. It has two instance variables, WellesleyMap mapGraph and LinkedList<JButton> linkList.

PlacePanel represents a place. It has a constructor which takes a String placePic, a LinkedList<String> links, and a String title. It is also used for the opening screen, where it is instantiated with the Explore button which leads to a PlacePanel of the currentPlace (initially Your Dorm (Stone D)) and the Adventure button which begins the ChooseYourOwnAdventure quiz. It also has a method getButtons() which returns the buttons contained in its InfoBar and a method getInfoBar() which returns its info bar. It has two instance variables, InfoBar info and JPanel img.

The QuestionPanel uses a specialized version of the InfoBar class to display a Question to the user and its two answer options. When the user comes to the end of the quiz, the QuestionPanel presents an analysis of the user's movements through campus. It has a constructor which takes a Question q and a ChooseYourOwnAdventure object a, and a getButtons() method which returns a LinkedList of its JButtons. It has seven instance variables, LinkedList<String> buttons, LinkedList<JButton> JButtons, String lAnswer, String rAnswer, String rQ, String lQ, and ChooseYourOwnAdventure adventure.

InfoBar is one of two component elements used to build the navigation bar and the various screens accessible in the GUI. It has multiple constructors corresponding to the various purposes for which it is used: InfoBar(LinkedList<String> buttons, String title), InfoBar(String title, String info), and InfoBar(ArrayQueue<Place> queue, String info, String title). It also has a method getButtons() which returns of LinkedList of any JButtons which are defined in the InfoBar so that they can be accessed in the WellesleyAdventure class and be connected to ActionListeners. It has one instance variable, LinkedList<JButton> buttonList.e