Paul Diaz

CS 151

Prof. Kim

SimpleCalendar GUI

In this project, I designed and implemented a simple version of GUI calendar application using MVC design pattern. To represent and manipulate this program, in terms of day, month, year, time of an event starts and time of an event ends, I used the GregorianCalendar class from the Java library that provides a standard calendar system. The entire program contains four classes in total; and each class is described below.

Event class is a class representation of an event, which has a name field, date of the event, time starts and ends (in hours and minutes). This class is the same class Event as the class I did in Project 2 (Calendar using console).

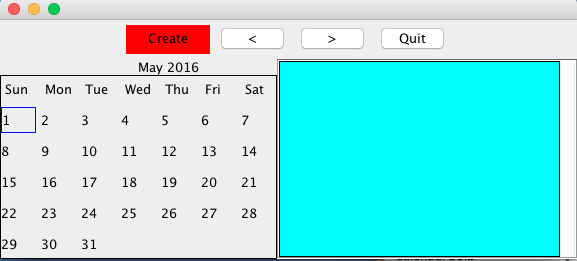
Model class models the code that carries out tasks, and therefore serves the Model of our MVC pattern. This class contains public functions that are used to achieve pretty much all of our program’s functions. For example, the ‘next,’ ‘back ’ functionalities, as well as adding an event to the calendar, Model class is the last one to carry out the task in updating the calendar including the look-and-feel of the calendar JComponent (JFrame, JPanel, etc.).

CreateEvent class is created when a user creates an event. This class is able to check if the new event to be added in the calendar has conflicting schedule. This class in particular serves as the View and Controller of our MVC pattern.

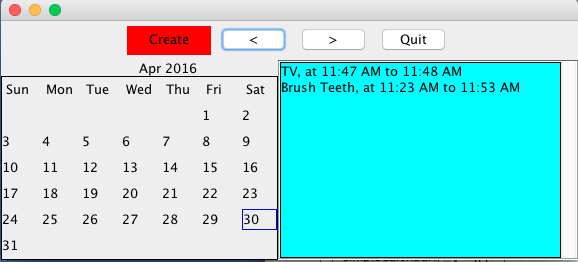
SimpleCalendar class has the main program that initiates to start our calendar program. Here, we created the whole JComponents (JFrame, JPanel, JLabel, JButtons, etc.) to describe the look-and-feel of the calendar. SimpleCalendar also serves as the View and Controller of our MVC pattern. In this class, it searches or creates a file called events.txt that contains some events that will be loaded to our calendar (by calling loadEvents() function). Also, we have a function called saveAllEvents() that we can use to save all events that we created during runtime. Functions such as printMonth() and printEvents() print the whole current month as well as the events saved for whichever day the user selected. And lastly, a function called update() is used to update the look-and-feel our calendar.

Here are screenshot of the calendar:

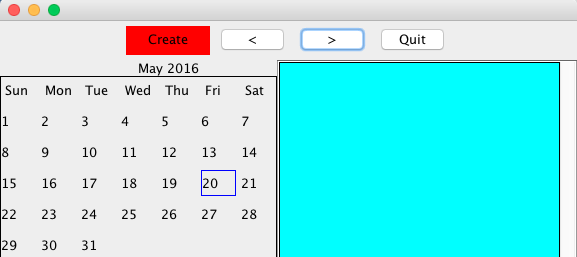
1. After starting the application (initial screen).



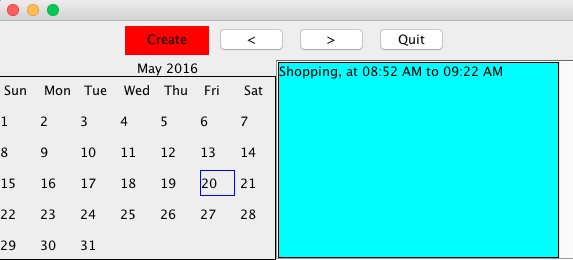
1. After clicking on the previous button on Screen 1



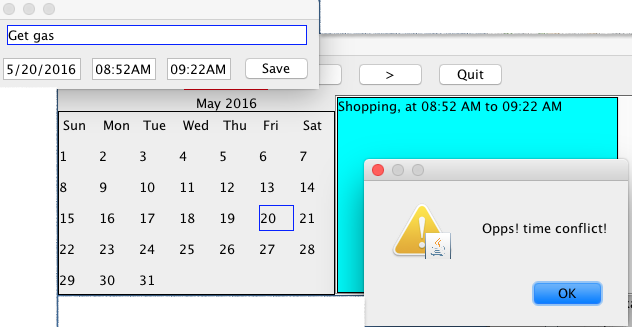
1. After clicking on the next button 20 times from screen 2



1. After clicking on a particular day on Screen 3 and creating an event on that day



1. After clicking on the same day I choose in the previous step and creating an event with a time conflict.



1. After entering an event on the same day I chose in the previous step and creating an event without a time conflict.