CSI 2300 Course Project

Schedule Planner

Ian Price

Purpose

- Allow users to create a schedule for a week.

- Events with titles, descriptions, and start and end times can be added to or removed from the schedule, or viewed.

- When two events overlap with each other, the corresponding conflict will be visualized.
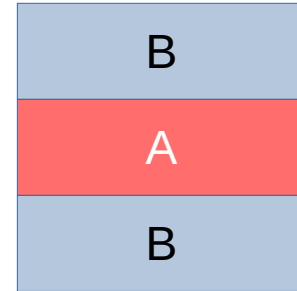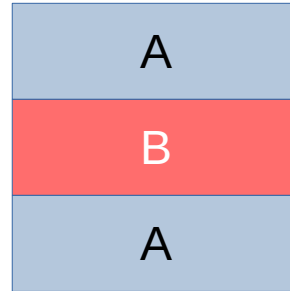
Why make this program?

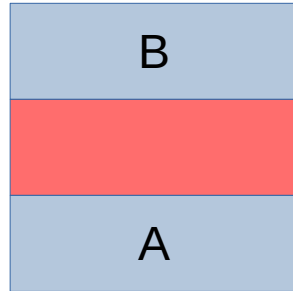- When planning a schedule, it's important to make sure two plans don't conflict. This program automatically finds and clearly displays any conflicts in a schedule.

- Adding events in this program is also very quick and simple.

Program Features

- Add events to a schedule

  - Each event has a name, description, start time, and end time.

- Remove events from the schedule

- View the details of any given event.

Program Features

- Gives a clear visual representation of events and conflicts on the schedule

- Automatically finds conflicts in several different conditions:

A		B		A		B
B		A		B		A
B		A		A		B

UML Diagram

ScheduleItem

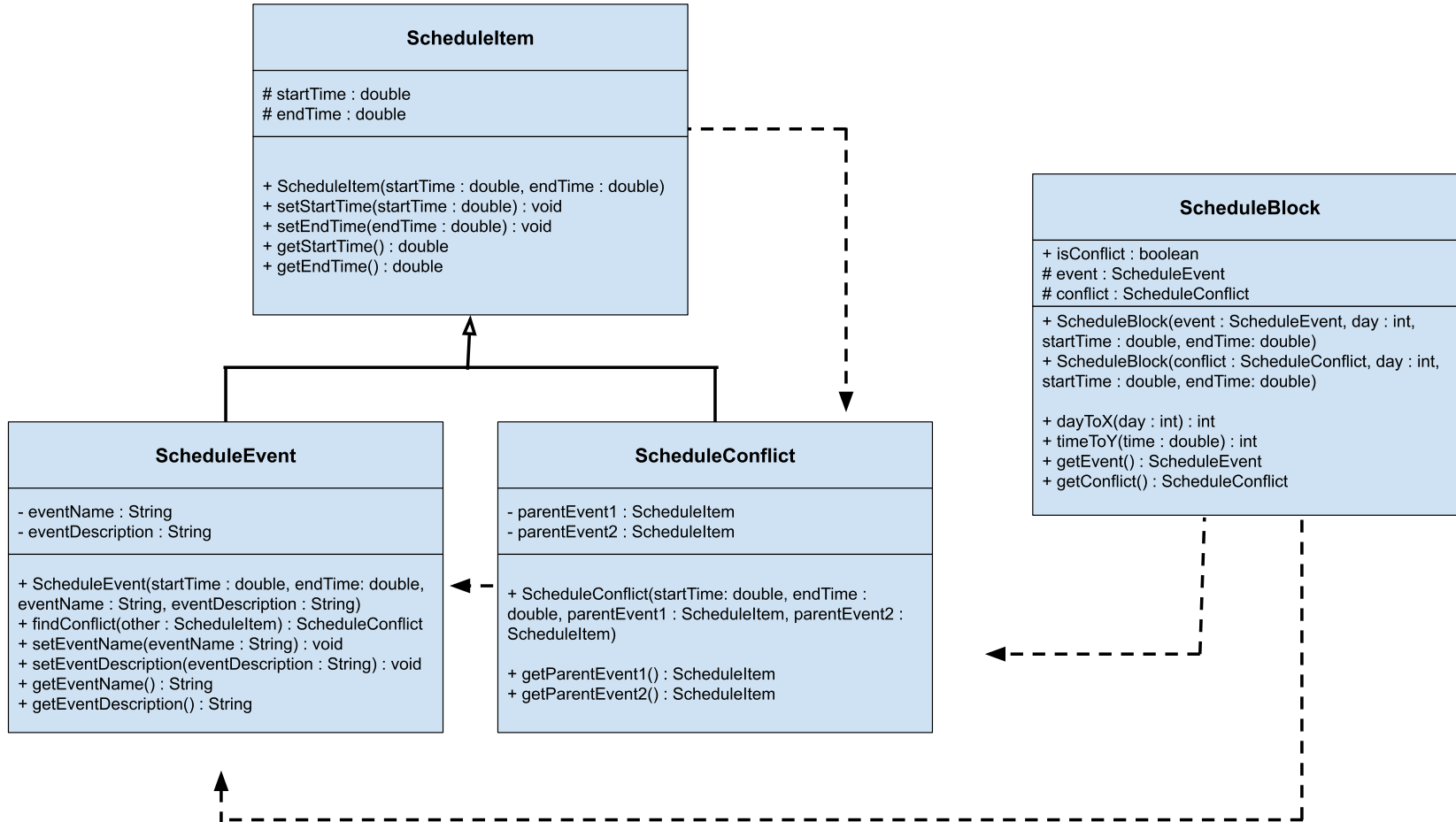# startTime : double
endTime : double

+ ScheduleItem(startTime : double, endTime : double)
+ setStartTime(startTime : double) : void
+ setEndTime(endTime : double) : void
+ getStartTime() : double
+ getEndTime() : double

ScheduleBlock

+ isConflict : boolean
# event : ScheduleEvent
# conflict : ScheduleConflict

+ ScheduleBlock(event : ScheduleEvent, day : int, startTime : double, endTime: double)
+ ScheduleBlock(conflict : ScheduleConflict, day : int, startTime : double, endTime: double)

+ dayToX(day : int) : int
+ timeToY(time : double) : int
+ getEvent() : ScheduleEvent
+ getConflict() : ScheduleConflict

ScheduleEvent

- eventName : String
- eventDescription : String

+ ScheduleEvent(startTime : double, endTime: double, eventName : String, eventDescription : String)
+ findConflict(other : ScheduleItem) : ScheduleConflict
+ setEventName(eventName : String) : void
+ setEventDescription(eventDescription : String) : void
+ getEventName() : String
+ getEventDescription() : String

ScheduleConflict

- parentEvent1 : ScheduleItem
- parentEvent2 : ScheduleItem

+ ScheduleConflict(startTime: double, endTime : double, parentEvent1 : ScheduleItem, parentEvent2 : ScheduleItem)

+ getParentEvent1() : ScheduleItem
+ getParentEvent2() : ScheduleItem

Lessons Learned

What I'm glad that I did:

- Before writing any code, I mapped out all aspects of the UI to-scale on graphing paper. This saved me a great deal of time.

What I'm not glad that I did:

- I didn't plan as much of the layout of the code as I should have before starting to write. This resulted in the code being harder to read than I would have liked.