Our project is called Sleep Blaster. Sleep Blaster is an alarm clock that can be set for a geographical location instead of a time. This could be useful for commuters on a bus or a train who are worried about falling asleep and missing their stop.

The interface is very simple. There is no need to locate a specific destination address or drop a pin on at a point on your route, which can be tedious at best and erroneous at worst. With Sleep Blaster, simply draw any line or shape on the map, and the alarm will go off as soon as you cross that line.

There are numerous other apps that can perform an action or set off an alarm once the user arrives at a particular location, but the problem with all of these other apps is that they only support a point-based destination locations. This is not very useful for the purpose of setting off an alarm, because if the user’s route changes after he has already set off he may miss the destination point completely. Similarly, if the GPS is slightly off, the phone may think it is far from the destination, when in fact it is right next to it.

By allowing the user to draw a boundary as wide as they want, and of any shape, we provide a much more reliable solution for alerting the user when they are near the destination or have reached a particular location. For example, if the user is on a road trip and there is a possibility that their route might change, the user could simply draw a line along the entire state border and be alerted when they are leaving the state.

We store each geographical boundary as a collection of contiguous lat/lon coordinates. However, in order to simplify mathematical calculations on these coordinates, we internally convert everything to a Cartesian system before performing any calculations. When we perform our calculations to determine whether the user is approaching the boundary, we must take into account the fact that the contiguous points comprising the boundary may be as close together as several feet, or as far apart as several miles, depending on the zoom level used when the user drew the boundary on the map. Thus it would not be prudent to simply determine whether the user is within a certain range of any of the points in the boundary. Instead, we handle this issue by calculating a hypothetical line between any two consecutive points in the boundary. If the user crosses this hypothetical line, the alarm should be set off. However, we must also take into account the facts that the GPS gets updates at discrete time intervals, and that the location data from the GPS has a maximum accuracy of several meters at best. Therefore we must set a distance threshold within which the alarm will set off the alarm as the user approaches the boundary. Given that the maximum accuracy of GPS technology is about 4 meters, we set our distance threshold to 10 meters on each side of the boundary. For example, if the user is within 10 meters of the hypotheical line between two points, the alarm should be set off. We calculate whether the user is within this distance by creating another hypothetical line, parallel to and 10 meters distant from the line between the two points. We then determine if the user’s current position is in between these two hypothetical lines. In order to save battery power, we use coarse location determination (i.e. cell tower triangulation or Wifi) when the user is more than 8km away from the boundary, and only switch on the actual GPS hardware when the user comes within this distance of the boundary. This is because coarse location determination is not accurate enough to determine reliably whether the user has crossed the boundary, but it uses much less energy than the GPS hardware, the higher accuracy of which is not needed until the user is actually close to the boundary.


description

| Implementation |

We have thoroughly tested Sleep Blaster and it works as intended. Through testing, we discovered that it was very accurate and able to set off an alarm within one meter of a lamppost that was positioned on our boundary line. As for next steps, there are many ways that we could improve our app and make it more useful. One obvious step is to improve the user interface so that it is more pleasing to the eye rather than simply functional.

The next improvement would be allowing the user to set off an alarm based on anti-location - that is, where he is not. For example, I could set an alarm to ring if I am not within a 1-mile radius of a certain place at 3pm this Friday. This would be a good way to remind myself if I happen to forget to be somewhere that day. If I were to remember, then the alarm would not go off. This is useful because current reminders will notify you even if you remembered the task; ours will remind you only if you forget. Implementation-wise, this would involve simply checking if the user’s current location is outside of the user-drawn boundary, which would be a small modification to our code.

Other possible improvements are more standard, such as allowing the user to select a song from his music library for use as the alarm, and having math problems or voice recognition to ensure that the user is actually awake. We are also considering including the option to yell at the phone to turn off the alarm, and a “Dynamite Mode” which would prevent the user from turning off the alarm and would set the sound to max for a full minute to ensure that you wake up. We could also have more integration with the calendar app. For example, when the user creates a new event, we can take that location and prompt the user to create an alarm at the same time. This would provide the user a quick and easy way to create alarms whenever he creates an event in his calendar. We could even do this automatically if the user wishes. In conclusion, our app is both usable and useful, with many possible future expansions to enhance its functionality.

| Acknowledgements |

This project was completed as part of CS 234 Mobile Computing (Winter 2013), taught by Professor Chien, with TA support by Tanakorn L. and Falcon Dai. We are grateful for the support and advice we received from Professor Chien and our TA’s, and especially for the suggestions we received for future improvements on the app.