CS 1110 HW 5 Due by scheduled demo with gob between Friday 2nd and Tuesday 6th August

*You have many choices. You may choose to do **any one of** Q1, Q2, Q3, Q4, Q5, or propose via email an alternative having sufficient complexity, depending on what appeals to you most (Q6). **You may work in project groups of size 1 to 4**. These questions are pretty open-ended and will require sane choices of initial versions to build, together with thoughtful choices of classes to construct in order to permit relatively painless enhancements as you move to more exotic versions. These are meant to give you scope to use what you've learnt in the course, and to focus on those aspects you find most interesting.*

1. This is one of two enhancements of the town graphs that you've been working with in previous homeworks.
   Model a transport network using a graph of towns and roads. You will need classes for vehicles (have at least classes for cars, taxis, buses, bicycles, trains), using inheritance where possible. The distances along roads/rail should be set according to the distances you decide between their corresponding pair of towns, and all vehicles should be given reasonable speeds (so that journey times can be determined). You should set things so that public transport (i.e., buses, trains, and aircraft) run on a discrete schedule, and that there's a set waiting time for taxis. You will need a class for the transport network itself (allowing for both one-way streets and two-way streets). Finally, you should have people with cash who wish to travel from towns to towns who can choose between speed and cost of transport, with transfer of cash from person to vehicle owner (if public transport) or person to energy company if using cars (have just one energy company for your entire network to keep things simple). Have your program input a file to initialize things, and then run a simulation with selective output to the screen (and full output to an output file) to show how transport moves around your system and various levels of congestion are handled, and you should display interesting data about the behaviour of this transport system.

   You might like to consider various possible enhancements of this. As some examples:
   - Some of your towns might have airports (with corresponding flightpaths), in which case travel by aircraft would have additionally a significant set time prior to departure and post arrival to allow for security screenings and baggage collection coupled with (short) journey times between the town itself and the airport.
   - You might like to explore the use of a GUI to help visualization and give you a chance to see the effects of people and vehicles moving around the network.
   - You might want to implement space limitations on towns or in vehicles (so that a bus would only have a limited number of seats, etc.).

2. A different enhancement of the town graphs.
   Model a transport network (e.g., of roads and intersections) *inside* a town using a graph of buildings and roads (so a building would then be 'on' a road, ie if a person is in a building and wishes to travel to another building, then they'd leave the building onto the road, travel along roads, and leave the destination road to enter the destination building). You will need classes for vehicles (have at least classes for cars, taxis, buses, bicycles), using inheritance where possible. The distances along roads should be set according to the distances between their corresponding pair of towns, and all vehicles should be given reasonable speeds (so that journey times can be determined). You should set things so that public transport (eg, buses) run on a discrete schedule, and that there's a set waiting time for taxis. You will need a class for the transport network itself (allowing for both one-way streets and two-way streets). You should arrange things so that depending on various natural conditions (your reasonable choice, well-commented of course!), priorities can be set (for example, buses win over cars, taxis win over buses, bicycles win over buses). Have your program input a file to initialize things, and then run a simulation with selective output to the screen (and full output to an output file) to show how transport moves around your system and various levels of congestion are handled, and you should display interesting data about the behaviour of this transport system. Feel free to consider similar (or other) enhancements as suggested in Q1 (though perhaps not including aircraft inside towns!).

3. A music question if you want to explore midi a little more. Create a program which allows the input of multiple 'voices', for example, it could simply be the input of your own (humanly) chosen notes to be played back, perhaps similar to the four part harmony created in Q3 (though without you needing to program any of the automatic harmonisation stuff!!!!). This question will require the use of a GUI. You should explore how to make micro-adjustments to pitch via midi, and then have the option of choosing (i) some specific notes, (ii) one voice to be 'bad' so having multiple bad notes, or (iii) randomly chosen bad notes. By a 'bad note' we mean that that note is microtuned to be out of tune. Your GUI would then allow playback of the 'music' with a button (or key stroke) allowing the music to be paused and corrected. The idea being that the user, on hearing something out of tune, could use a pause-and-hold button together with sliders so that the user can pause and adjust the tuning of individual voices. You might want to have a slider to permit the user to slow down the playback so that they can hear things more clearly.

CS 1110 HW 5 Due by scheduled demo with gob between Friday 2nd and Tuesday 6th August

4. Another music question *(for those currently taking computing in the arts, there is an expectation that if you choose this question, your solution will be significantly enhanced):* Build a system to create music automatically. The core idea is that harmonising a (diatonic) melody (having no modulations) can be done with triads. For example, a melody in C major would be built out of the notes A, B, C, D, E, F, G, and a note, e.g., F, can be harmonised with any of the three triads FAC, DFA, BDF. Therefore, for each note there will be three possible triads containing that note, so for a first pass at harmonising we could choose purely randomly between the three options for each note in the melody. To do four-part harmony (soprano, alto, tenor, bass) would require choosing three notes from each triad for each note in the tune (hence at least one note name will be repeated). In order that the end result can sound reasonable, we will want to ensure that not all voices go in the same direction all the time; hence will decide almost randomly whether a given voice at a given time; hence will decide almost randomly whether a given voice at a given time should go up, down, or stay the same (if that's an option given the following chord), and should ensure that there's some variation between pairs of voices (i.e., avoid the case where bass and tenor always move in the same directions). Also, each individual voice should be 'singable' in the sense of not having too many jumps, and ideally not being overly boring (too much constant or stepwise motion). Once that version has been done, one should 'improve' the program by using the rhythm of the melody to decide whether to ignore harmonising some of the notes (i.e., perhaps not harmonising notes which are not on 'main beats', e.g., beats 2 and 4 in 4/4, or beat 2 in 3/4) by continuing the prior harmony during those 'non-harmonised' notes. All of these choices can be made quasi-randomly by providing functions which bias choice. You should provide a user interface to allow the user to experiment with various biases.

   Finally, we tackle the problem of writing a tune. This will be done by first selecting harmonies (as a sequence of triads, where we'll again use biased choice in selecting which of the seven possible triads will follow the current triad), and then picking a 'singable' tune from the notes in that harmony (where 'in' means either actually in the corresponding triad or ignoring the current triad if it's not on a main beat). The output of this program will be named notes and note durations which can be fed into a program to create a midi file which then can be played.

5. Consider the following enhancement of friendship networks. Suppose you have two countries A and B, each with vast networks of people and corresponding friendship networks. Suppose also that you have a small number of friendship connections between people in the two countries (i.e., a few people in A are friends in B, and vice versa). In a friendship network, some people are more connected than others, so one could view the network as clusters of people, with clusters being connected to other clusters by relatively few connections (the clusters themselves being highly interconnected within themselves). As suggested in lecture, since you are the person creating this friendship graph, you have the ability to ensure that there is a dramatic difference between clusters and non-clusters, and that there is a sufficiently clear difference between highly connected people and others in their friendship group – use that ability (it's a seriously hard question if you don't!!!!!!!).

   If someone in A wants to influence people in B, then you could think of this an opportunistic disease trying to choose which parts of the network to infect, given that the process of infection has a cost, and so wanting to minimise the number of people in B to infect yet yielding maximal impact. In the context of the friendship network, it would therefore be more effective for the person from A to try to 'infect' some of the most highly connected people in B, since that way, each cluster in B that those people are connected to can become 'infected' due to the multiple connections within the cluster. Try building a system to simulate this and explore the effects. You will want to build ways to find clusters (relatively highly interconnected people separated by relatively few connections), and then construct functions to show the proliferation over time (e.g., infection moves to all the immediate friends of those infected once per time interval). After that, you might want to explore ways of 'inoculating' people in B from such infection under the (cost) constraint that you won't be able to inoculate everyone on B, so will need to choose which people to inoculate in order to minimise the spread of infection. You can talk with me to discuss further aspects or seek more understanding about this question.

6. A question of your own choice ☺ Contact me via email or in person to bounce your ideas – as I said in class, the most common danger is that you'll be too ambitious in your ideas … almost anything is possible, given time!