Program Design Report

# Problem Definition:

The problem as we’ve chosen to implement it is floor planning in a C-train environment. Since the immediate intended audience is fellow classmates of ENCM 507 and potentially faculty members of the University, most of the audience will have a personal understanding of the struggle of trying to maximize space utilization on the Calgary LRT line. Since the C-trains are laid out in a very specific way, and rearrangements based on the occupants of the train are variable, this problem is easily relatable to the floor plan content taught in class as of the writing of this document.

The type of floor planning we will be evaluating for our design is contiguous placement (think Tetris) since the user’s objective will be to keep white space to a minimum while placing the passengers in the optimal position. This is related to VLSI design because a CPU is made up of several contiguous blocks which implement specific functions of the CPU and need to be placed properly relative to each other.

# Program Explanation:

For our design, the interior of the C-train will be split into different sections (for example static seats, retractable seats, center rotating section, etc.) that will have different points awarded for specific ‘pieces’ placed there (people with backpacks, people in wheelchairs, etc.). The C-train layout and the passengers will all be rectangular in shape for ease of placement (to behave like algorithms seen in class), and will be randomly generated at each station based on difficulty. The weightings of the randomization will heavily favor passengers that are easily placed at a lower generation volume for lower difficulty and higher difficulties will feature more difficult to place passengers more heavily with a higher volume of generation.

There will be two modes of play: static and dynamic. In static mode, passengers will enter the train and stay there until the round is complete. In dynamic mode, passengers will have an internal timer where they will leave the train after a certain number of station transitions and the user will have to adjust accordingly. Round length will always be the same (5 station transitions) and the stations will vary depending on difficulty. Each station transition will feature recorded C-train noises to engage the user, and will follow the real train line order. The user will drag and drop passengers onto the grid and can re-arrange new pieces until either the timer for that station runs out, or the user decided to traverse to the next station manually. A time bonus will be added to the overall score if the user completes the station before time runs out. Our scoring algorithm will be based on scoring multipliers for passenger/seating area arrangements, time bonus, and subtracting unfilled space.

# Storyboard:

Please see attached pages for storyboard, as putting them in this report document was inconvenient and caused formatting issues.

# Project Management:

Work will be divided up on a need to know basis when we encounter the tasks. Generally we will both be scrumming on the tasks to get a better understanding for both of us. Pair programming may be needed in certain circumstances, especially for more complex algorithm implementation. All of this allows us to get a solid understanding as to how our program works, in addition to letting us both be able to troubleshoot problems if necessary. We have put a lot of simpler tasks, like setting up pages with complex tasks to allow us to split up tasks and get work done in parallel. As for our schedule, we have included a Gantt chart to explain how we will be dividing the tasks into chunks, in addition to scheduling the whole project. It also shows our time and activities estimates. Our work breakdown structure also explains the deliverables for each section. Both of these are attached with this document due to formatting issues.