1. Hogwarts Hobo game

In your desire to qualify for the Hogwarts school, you have to design a game which goes as follows.

Assume there is a railroad with M tracks, each of which may be occasionally occupied by a train that takes L1 seconds to pass, after which there is L0 seconds before the next train arrives. (All trains move in the same direction.)

You stand at the exit of the tunnel – it's awful out there, so you need protection from rain, snow, hail, brimstone, and other stuff that might be falling down – but since there is no firm ground, you're forced to stay on one of the railroad tracks.

To avoid trains, you jump from track to track with the goal of staying on tracks as long as possible (for obvious reasons). You typically stay on any given track about l seconds before you jump to another track. Note that you may jump from any track to any other track – the distance between them doesn't matter.

As there is a lot of noise and it is actually night-time over there, you can't really know whether the track you're jumping to will be occupied by the train or not. As the result, you may get hit by a train while you're on the track, or you may jump right onto a passing train. While neither of these is lethal per se, your goal is to minimize the probability of such events occurring, as they cause stress and actually reduce your health.

While on track i, you may get some information about the trains that pass on other tracks from other hobos that indulge in the same exercise; however, they send the information by paper planes which makes the process somewhat challenging. Namely, this information may reach you too late (i.e., what you read may be the status of track i some time ago) and even if it does reach you on time, it does not really guarantee a successful jump (i.e., a jump onto an idle track).

On the bright side, the time l is long enough for you to get information about several tracks; but due to the darkness and noise, you may simply miss the paper plane carrying the information; or you can misread the information since there is not much lighting there (remember, it's the EXIT of the tunnel, not its END). And the other hobo may simply lie to you which makes your life even more interesting.

Needless to say, time intervals L0 and L1 are random variables that follow the same probability distribution for trains on all tracks, but with possibly different parameters. Initial choice is Poisson (i.e., negative exponential duration of these times), but other distributions can be substituted. You may want to learn something about individual train patters and use that information. You may also want to learn about the behavior of other hobos and use that information. The time interval l is initially fixed, but it may also be made variable.

You may also develop a band of hobos that try to jump in synchronization. You may want to inform them about the target track by shouting the next track immediately after, or immediately before the jump. (You can hear only the band leader shouting from your track, but not anyone else from any

other track.) An interesting question would be to design a protocol to maintain synchronization, even after being hit by a train which leaves everyone a little dizzy and necessitates that all of you re-group as soon as possible on an idle track. Also, it would be interesting to develop a protocol to find another (unrelated) band of hobos if it's already there.

The game, then, simply proceeds as follows. You should be able to choose the number of tracks, the mean values of train duration and inter-train distance, the probability distribution of train duration and inter-arrival time, the number of other hobos at that same tunnel exit, and other parameters as needed. (But make sure you inform the instructor!). Your job is to devise an application to manage your exercise and maximize your lifetime. As noted above, being hit by a train is not lethal (do not repeat NOT attempt this in real life, though) but the goal is to minimize the number of hits in a given time. To put it briefly, you get a certain amount of health which is reduced by every hit, and the winner is the one who stays healthy for the longest time.

Requirement	Story	Priority	Task	Estimate	Actual	Status
1.0 Train dodging phase	This is the main part of the game which will involve the user having to dodge trains while standing on train tracks. The game will require an interactive window.	HIGH	Create a window that displays the entire compartments for the game on the screen.			
1.1 Tracks on the screen	This will be the playing field of the game as the player controlled character can only jump between these.	HIGH	Create an algorithm that generates the tracks randomly on start up. Draw the tracks using the algorithm.			
1.2 User controlled character	We want the user to be able to dodge trains by jumping between tracks. The user can do this by using the arrow keys to move.	HIGH	Create sprite that moves with the arrow keys. Draw the sprite on top of a hit box.			
1.3 Trains on tracks	The trains will appear at a random time between a time interval and at	HIGH	Create train class and use the train algorithm that generated the tracks			

	random spots on the track. They will always be heading towards the user (straight).		to randomly put the trains on the tracks.			
1.4 Darkness	The screen will be completely dark other than the area around the player and at the front of the trains. (as they drive down the track)	**MED**	Create a shading class that shades the screen of black in areas you want.			
1.5 Health for character	Every time character is hit by a train/jumps into a moving train their mental health will decrease until "0" (from stress)	HIGH	Create a counter inside the character's class. Keep track of number of hits and display it on screen.			
1.6 Timer	There will be a timer at the top showing how long there is until the round is finished	HIGH	Have a counter on the board class that ticks down to 0 and when that happens, it sends a signal			
2.0 Planning period	After stages of trains coming at the character, the user will be given a certain amount of time before the next round of trains	HIGH	Have the timer reset to 30 when hits to 0 that halts the signal of generating the new trains till the timer hits 0 again.			
2.1 Hobo plane exchange	During the planning period, there will be random airplanes flying on the screen that contain information on where trains will be during the next round. When they are initially flying they will have a glow, but the glow	**MED**	Use the algorithm from the tracks and trains to generate the planes that have hit boxes to allow the player to click on them to generate hints for next round.			

	will slowly dissipate as it flies.					
2.2 Information type	The planes will contain information such as: 1. What track a train will be on at what time 2. What the busiest track will be 3. The safest track for the round	**MED**	Have a database of hints from usefulness or not that is generated by the algorithm and display as many as the planes were clicked.			
2.3 User and plane interaction	The user will be able to click on the plane while it is flying, to gather the information for the next round	**MED**	The click will increase the counter of information given			
2.4 Text Box with information	At the end of the planning phase a text box will appear showing all learned information	**MED**	The hints will be pulled from the database onto the screen.			
2.5 Hobo personalities	Each plane will be from a certain hobo. Sometimes hobos lie more than other hobos. When the information is displayed in the texbox, each piece of information will have the hobos name in front of it.	**LOW**	Have a hobo class that relates to each hint in the database.			
3.2 Visible Hobos	The same hobos that send the planes will also be visible on the track trying to dodge trains.	**LOW**	Have a generated hobo that moves in a randomize pattern on the track.			
3.3 Ability to create band of hobos	We want the user to have the ability to befriend the hobos	**LOW**	Have a friendship counter on the hobos and once the hobo's			

	and create a “band of hobos”, that work together in dodging the trains.		friendship reaches a certain level it will follow you and help you.			
3.4 Other bands of hobos	We want other bands of hobos to also appear on the tracks. They may potentially be hostile by not allowing the user to jump to the same track they are on.	**LOW**	At the start of the round, hobos will have a chance to join each other by “rolling” on their friendship with each other.			
3.5 Noise	We want loud train noises playing throughout the train dodging phase. We also want weather sounds playing for both the train dodging phase and the planning phase.	**LOW**	Import soundtracks that play throughout the game in a loop.			
3.6 Weather (hail, brimstone, snow, rain)	We want things falling constantly to add intensity and atmosphere.	**LOW**	1. Create the different type of weather effects.			
			2. Have a randomly generated weather effect for hail, snow, and rain.			
			3. Have brimstone weather only when user is one train hit from losing.			
4.1 Title screen	We want a title screen with the name of our game	HIGH	Create a start button, create an instruction button, and create a			

	("Hogwarts Hobo"). It will also have two options: 1. New game 2. Instructions		menu layout that appears at the boot up of the game.			
4.1.1 New game options	New game will start the game. The user will be able to set these elements of the game: 1. Number of train tracks 2. Average value of train duration on a track 3. How varied the train duration can be 4. Number of hobos	**MED**	Create input boxes for the number of train tracks, average value of train duration, number of hobos, and how to vary the trains' duration. Then generate the game with those the input.			
4.1.2 Instructions	A short explanation of the goal of the game, the elements found in the game, and the controls.	**LOW**	Create a short setting for the game information when the instruction is clicked.			