STRATEGY:

- Either beat the computer's perfomance using a more efficient strategy (i.e. Threads) **or** aim to fulfill the objective in the time-frame provided.
- Assignment of movement/tasks, etc. Would be done through a drag-and-drop interface.

IDEAS:

ER

Objective: Make it through the day without any patient dying.

- Patients = Task
- Resources = Charts, operating room, xrays, tools, etc.
- Doctors = Threads

Amusement Park

Objective: As your first day as en employee at the amusement park, you need are assigned a series of different tasks. Don't get fired!

- **The Roller Coaster Problem (*Semaphores*):**

  Suppose there are **n** passengers and one roller coaster car. The passengers repeatedly wait to ride in the car, which can hold maximum **C** passengers, where **C < n**. However, the car can go around the track only when it is full. After finishes a ride, each passenger wanders around the amusement park before returning to the roller coaster for another ride. Due to safety reasons, the car only rides **T** times and then shot-off.

  Additional constraints:

  - The car always rides with exactly C passengers;
  - No passengers will jump off the car while the car is running;
  - No passengers will jump on the car while the car is running;
  - No passengers will request another ride before they can get off the car.

- **Ice Cream Truck**
- **Fish Tank**
  - Fish, of different sizes, compete for food underwater.
  - Larger fish are prioritized, since they need more food. If they don't get it, they may starve (literally).
  - Two fish may not reach for the food at the same time.
- **The Bridge Problem (*Race Conditions*):**

Consider a narrow bridge that can only allow three vehicles in the same direction to cross at the same time. If there are three vehicles on the bridge, any incoming vehicle must wait until the bridge is clear. When a vehicle exits the bridge, we have two cases to consider. Case 1, there are other vehicles on the bridge; and Case 2 the exiting vehicle is the last one on bridge. In the first case, one new vehicle in the same direction should be allowed to proceed.

Case 2 is more complicated and has two subcases. In this case, the exiting vehicle is the last vehicle on the bridge. If there are vehicles waiting in the opposite direction, one of them should be allowed to proceed. Or, if there is no vehicle waiting in the opposite direction, then let the waiting vehicle in the same direction to proceed.

- **The Dining Philosopher's Problem (*Mutex Locks*/*Deadlocks*)**

Barber Shop

(could also be a puppy grooming place… ???)

Objective: Make it through the day without losing any customers.

- Customers = Rasks. There may be different or similar requests.
- Tools = Resources
- Barbers = Threads
- Only so many customers may come in at the same time, if we decide to include semaphores.

If customers wait too long after they have put in a request, then they leave and a customer is lost.

Could we also include the cashier element for race conditions?

If too many threads (barbers) are used, then it gets more difficult to walk around eachother and to share the resources available.