CMSC 498M:Fall 2007 Dave Mount

Solutions to the Take-Home Final Quiz

Solution 1: (Note: The problem erroneously failed to state that the car's initial velocity is 0. You may either treat the initial velocity as a variable or as 0. In either case I'll give full credit.)

(a) By the basic laws of kinetics, we know that $F(t)dt = mdv$, where $F(t)$ is the force at time t, which is just the constant T. To determine the velocity at some time t, we integrate both sides. At time $t = 0$ the velocity is 0, and so we have

$$\begin{aligned} F(t)dt &= m\,dv \\ \int_{t=0}^{10} F(t)dt = \int_{t=0}^{10} T\,dt &= \int_{v=v(0)}^{v(10)} m\,dv \\ T \cdot t\Big|_{t=0}^{10} &= m \cdot v\Big|_{v=v(0)}^{v(10)} \\ 10 \cdot T - 0 \cdot T &= m \cdot v(10) - m \cdot v(0) \\ 10T &= m \cdot v(10). \end{aligned}$$

Therefore, we have the final result $v(10) = 10T/m$.

(b) To determine the location, we could set up the kinematics motion equation $dx = v(t)dt$ and integrate both the left and right sides, as we did in class. Because the force is constant, and hence the acceleration is constant, it will be easier to apply the equation derived in class for computing the position assuming constant acceleration, which is

$$x(t) = x(0) + v(0)t + \frac{1}{2}at^2.$$

We have $a = dv/dt = F(t)/m = T/m$. Since the car starts at $x = 0$ with zero velocity, we have $x(0) = v(0) = 0$. Thus, the position at time 10 is

$$x(10) = 0 + 0 \cdot t + \frac{1}{2}\frac{T}{m}t^2 = 50\frac{T}{m}.$$

(c) In this case we have a force $F(t) = -B \cdot v(t)$. Here we start with the kinetics formula $F(t)dt = mdv$, but because the force depends on velocity, it will be convenient to express this as $dt = (m/F(t))dv$. For the purposes of part (d), we compute $v(t')$ for any $t' \geq 10$. Integrating over the range of times $[10, t']$, and range of velocities $[v(10), v(t')]$, we obtain

$$\begin{aligned} \int_{t=10}^{t'} dt &= \int_{v=v(10)}^{v(t')} \frac{m}{F(t)}dv = \int_{v=v(10)}^{v(t')} \frac{m}{-B \cdot v(t)}dv = \frac{-m}{B}\int_{v=v(t')}^{v(t)} \frac{1}{v(t)}\,dv \\ t\Big|_{t=10}^{t'} &= \frac{-m}{B}\ln v\Big|_{v=v(10)}^{v(t')} \\ t' - 10 &= \frac{-m}{B}(\ln v(t') - \ln v(10)) = \frac{-m}{B}\left(\ln v(t') - \ln\left(\frac{10T}{m}\right)\right) \\ \ln v(t') &= \ln\left(\frac{10T}{m}\right) - \frac{(t'-10)B}{m} \\ v(t') &= \exp\left(\ln\left(\frac{10T}{m}\right) - \frac{(t'-10)B}{m}\right) = \frac{10T}{m}\exp\left(-\frac{B}{m}(t'-10)\right). \end{aligned}$$

Observe that when $t' = 10$, this matches the value of $v(10) = 10T/m$. As t' increases, the velocity decreases as an exponential function of time and reaches 0 only in the limit. (Of course, our physical model of breaking does not take into account more complex forces that would cause the velocity to fall to 0 much sooner.)

Plugging in $t' = 15$ we have the final result:

$$v(15) = \frac{10T}{m} \exp\left(-\frac{5B}{m}\right).$$

(d) To obtain the final position, we need to integrate the velocity over time. Recall that $dx = v(t)dt$. Thus we have

$$\begin{aligned}
\int_{x=x(10)}^{x(15)} dx &= \int_{t=10}^{15} v(t)dt = \int_{t=10}^{15} \frac{10T}{m} \exp\left(-\frac{B}{m}(t-10)\right) dt \\
x\Big|_{x=x(10)}^{x(15)} &= \frac{-10T}{B} \exp\left(-\frac{B}{m}(t-10)\right)\Bigg|_{t=10}^{15} \\
x(15) - x(10) &= \frac{-10T}{B} \left(\exp\left(-\frac{B}{m}(15-10)\right) - \exp\left(-\frac{B}{m}(10-10)\right)\right) \\
x(15) - 50\frac{T}{m} &= \frac{-10T}{B} \left(\exp\left(-\frac{5B}{m}\right) - 1\right) \\
x(15) &= 50\frac{T}{m} + \frac{10T}{B}\left(1 - \exp\left(-\frac{5B}{m}\right)\right)
\end{aligned}$$

Solution 2:

(a) The number of rows is $(150-(-50))/5 = 200/5 = 40$. The number of columns is $(100-0)/5 = 20$.

(b) First, observe that the position of a point with coordinates (x, y) relative to the rooms lower left corner is $(x - 0, y - (-50)) = (x, y + 50)$. Since each square is of width and height 5, it follows that the column is $\lfloor x/5 \rfloor$ and the row number (counting up from the bottom) is $\lfloor (y + 50)/5 \rfloor$.

(c) Over 0.1 seconds the point can have moved a distance of at most 8 meters, and therefore, it might lie within any square that is within distance 8 of the boundary of the square [10][10]. Fig 1 (left) illustrates this set of squares.

(d) We simply expand the result of part (c) by an additional distance of 3. for a total distance of 11 from the boundary of the initial square. (See Fig 1 (center).) (In practice, you would probably just compute a bounding box and visit all the squares within the bounding box.)

Solution 3:

(a) Figure 1 (right) shows the C-obstacles A' and B'. These result by growing a 4×2 rectangle about each point of A and B, and taking the union of the resulting shapes.

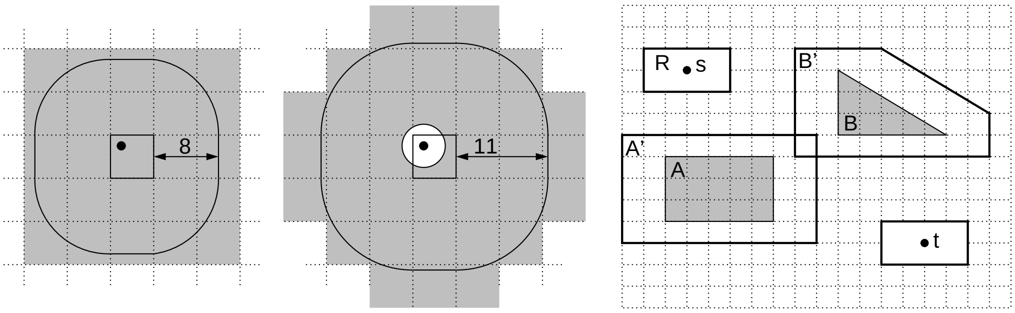

Figure 1: Solutions to Problem 2(c), 2(d), and 3.

(b) Because the two C-obstacles overlap, it follows that there is no free translation path between A and B for R. (Thus, it would either be necessary to go around the obstacles or use rotation.)

Solution 4:

(a) This is a pretty easy scenario where any simple heuristic will suffice. Best-first search should be adequate.

(b) City environments are not too hard to navigate within (since presumably humans have designed them) but there may be blind alleys and the like. A^* search would be the most efficient, since it gives preference to direction toward to destination. Best-first search might wander up a blind alley, and then be forced to back-track. Dijkstra's algorithm would certainly work, but would take longer.

(c) Best-first search would certainly be a poor choice. Clearly Dijkstra or A^* would be the better options. Depending on the amount of debris, there may be little advantage to be gained by A^* search, since an entirely global analysis may be needed. Of course, A^* has the advantage that in regions of lower complexity, it will tend to perform better than Dijkstra's.

Solution 5: If the radius is too small, the a boid that is at some distance from the group is no longer constrained at all, and may tend to fly in a random direction. (Although cohesion may eventually bring it back to the flock.) If the radius is too large, the total effect may be for all the boids to fly in exactly the same direction, which is rather unnatural. In addition, a large radius implies that every boid must track the position of many others. This could lead to decreases in efficiency (assuming that you use a good data structure to identifying nearby boids).

Solution 6:

(a) Network layer.

(b) Session layer.

(c) Transport layer.

Solution 7:

(a) Snap: The easiest to implement. But results in discontinuities if the object moves rapidly or unexpectedly.

(b) Morph: More continuous than snapping. Morphing can result in the object passing through walls, if collision avoidance is not applied. Because of latency, the object is always playing "catch-up".

(c) Converge: This is most computationally intensive of the three. Works better than morphing for objects that move along predictable paths (e.g., airplanes and cars), but not good for unpredictable rapidly varying motion (e.g., sword fighting), since predictions are either unreliable or take a great deal of computation time.

Solution 8:

(a) Protagonist: The central character of the story (typically the hero).

(b) Antagonist: The principal character that opposes the protagonist.

(c) Suspension of belief: Players acceptance of the artificial reality that is created within the game.

(d) Foreshadowing: A plot device that provides a hint of some event that will happen in the future.

(e) Shadow: Counterpart to the hero of a story.

(f) Breaking the 4th Wall: Violating the illusion that that the characters/action is part of a story, thus breaking the sensation of immersion.

(g) Monomyth: A common multipart structure to classic myths (which is imitated in modern stories as well) based on the story of a hero's departure, struggle, growth, victory, and successful return.