**Unit 2 - Constant Velocity Particle Model**

# Instructional Goals

1. CV1: I can represent the motion of an object using a computational representation in which a differential is added for each interval of time.
2. CV2: I can design and modify functions in Pyret to represent constant velocity motion and use them to make predictions.
3. CV3: I can differentiate between position, distance, and displacement.
4. CV4: I can find the velocity of an object using the slope of the position vs. time graph. I can interpret the initial value of a position vs. time graph. I can draw an average velocity line on the graph and calculate the average velocity.
5. CV5: I can create a mathematical representation (function) relating position, average velocity and time, and use it to solve problems.
6. CV6: I can represent the motion of an object moving with a constant velocity using multiple representations (words, motion maps, position vs time graphs, and velocity vs. time graphs), and use them to solve problems.
7. CV7: I can relate displacement to the area between the horizontal axis and the line on a velocity vs. time graph.

# Overview

1. It is important to describe motion in terms of *position* (rather than distance) and time. Position is much less ambiguous than distance (sometimes regarded as the path length). Some authors use the identifier *s* to describe this variable; we prefer *x* for horizontal motion (and *y* when the motion is vertical). We advise against the use of *d*.
2. The ***differential representation*** of motion which is introduced through coding is central to the understanding of motion built in this unit. This idea that position changes iteratively by some small amount over some small time interval is a natural way to think about change and forms the basis of our computational representation of motion. Investigating the relationship between and naturally leads to the definition of *average velocity***,**.
3. Work on making motion maps to represent the position-time behavior of moving objects. Make sure that these semi-quantitative devices are faithful representations depicting where the object is at evenly-spaced clock readings.
4. Motion maps form the basis for the computational representation of motion in this unit. Emphasize that the difference between dots is the displacement of the object for that particular interval of time. As the time interval decreases the displacement must also decrease if the object is to remain moving with the same rate of change.

A

Students have been known to draw the motion map above and state that B was moving faster than A because the velocity vectors were longer. If object B is moving faster than object A, then the space between the dots should also be greater because B’s change in position will be larger during each time interval.

1. The definition for *average velocity*, , should be used explicitly in discussing the physical significance of the slope of a position-time graph. Change in position is superior to change in distance; the latter is a difference of differences. Change in position is the definition of *displacement,* the quantity that helps distinguish velocity from speed. Mathematically stated, . Displacement can be (+) or (-), distance is by definition (+). A positive displacement indicates that the position has changed such that the object moved in the positive direction. A negative displacement indicates that the object has moved in the negative direction.
2. When discussing the meaning of the graphs, be sure to use a wide variety of examples.

Induce the students to describe the motion in full detail (e.g., in graph 4, the object starts somewhere to the right of the zero position and moves to the left at constant speed).

1. Students will have to be taught how to manually produce a graph and do a mathematical analysis of the graph. Students have been conditioned to think of slope only as "rise over run" or ∆y over ∆x. They need to understand that the slope of any graph describes the rate of change in the physical quantity represented on the vertical axis with respect to the one represented on the horizontal axis.
2. Make sure that they have a thorough grasp of the relationship between slope and velocity. The answer "1's slope is greater than 2's" is not a guarantee of understanding. It would be profitable to have students model the behavior of the object represented by a variety of graphs. If you have an ultrasonic motion detector, this is great fun!
3. Make sure that students can, given a verbal description, an algebraic statement, a function in Pyret, an vs graph, or a motion map, recreate the other four representations.

Object A

0

+

Given the motion map above, they should be able to write a verbal description of the motion, express the relationship and draw the graph at right.

1. Be sure to make the connection between vs *t* graphs and vs *t* graphs. "Stacking" the curves by placing the vs *t* graph directly underneath the vs *t* graph helps to illustrate this relationship.

*t*

*t*

*t*

*t*

Make the point that the vs *t* graph yields no information about starting point. In the two stacks to the right, two different vs *t* graphs can be represented by the same vs *t* graph.

1. Make the point that the area under a vs *t* graph represents the displacement, the of the object. This could be both (+) and (-). Avoid always using the trivial case.
2. Key understandings about computation in physics raised in this unit are:
3. Computers allow us to do many repeated calculations which would be too time-consuming to do by hand.
4. Computers can simulate (represent) physical behavior and use those simulations to make predictions about physical phenomena.
5. Instructions which should be repeated multiple times can be written using functions.

# Sequence

1. Lab 1: Buggy Lab
2. Reading 1: Writing Functions in Pyret
3. Activity 1: Simulating Motion
4. Activity 2: Simulating Boats
5. Activity 3: Motion Maps
6. Activity 3: Differential Motion Practice
7. Activity 4: Multiple Objects
8. Lab 2: Colliding Buggies
9. Activity 5: Graphing Constant Velocity Motion
10. Reading 2: Motion and Trajectories
11. Worksheet 1: Two Bicycles
12. Reading 3: Number Lines, Distance, Displacement, Average Speed, and Average Velocity
13. Worksheet 2: Distance vs. Displacement
14. Lab 3: Graph Matching with Motion Detectors
15. Worksheet 3: Velocity-Time Graph
16. Worksheet 4: Multiple Representations of Motion

## 

## Lab 1:Buggy Lab

## **Note:** The included worksheet includes pre-lab observations and should be given in conjunction with the lab.

## **Apparatus**

* Any slow-moving battery powered toy vehicle (preferably two—a slow and fast one)
* Some type of timer (stopwatch, app on smartphone, metronome –actual or virtual)
* Meter sticks or metric tape
* masking tape or dry erase markers
* Logger *Pro* software or any graphing/data analysis software (to include graphing calculators)

## 

## **Pre-Lab Discussion**

* Let the vehicle move across table and ask students the following:
  + What do you notice/observe?
  + How can you tell it is moving?
  + What can we change? What are the possible inputs?
  + What is measurable? What are the outputs for our possible inputs?

List observations and then ask which items are measurable. Students are likely to say “speed”; point out that since the buggy doesn’t have a speedometer, this is a quantity we cannot directly measure. Lead them to observe that the BPV (battery-powered vehicle) moves at constant speed; i.e., that it travels equal distances in equal time intervals. The pre-lab discussion should lead to a definition of motion along the lines of “something is moving if its position changes” or “its position is different at different times.” Based on this definition of motion, the experiment is designed to measure the position of the buggy at set interval of times, as determined by a metronome (real or on computer[[1]](#footnote-1)) set to 30 beats/minute.

The variable being measured in this experiment is position (). Emphasize the difference between position and distance (and displacement, if it comes up). You can give them a quick description of vector quantities here to explain why there is an arrow in the symbol if you want. Position will be measured in centimeters. Positive and negative positions are on opposite sides of 0.

Another important thing to keep in mind in this experiment is that students are not recording time as a variable in their experiment. In discussing their data, the individual beats of the metronome are referenced rather than some clock reading or absolute time.

Show students how to make a proper data table for this experiment if necessary. In this case, the data collected fits in a single row or column for the position of the buggy at each beat.

Students should end the lab with some description of a pattern they observe in their data.

## **Lab performance notes**

Slow-moving vehicles make it easier for students to mark the position without the use of a high-tech apparatus. The constant speed buggies can be made to move more slowly by wrapping one of the batteries in electrical tape, then in aluminum foil, or by replacing one of the batteries with a wooden dowel wrapped in aluminum foil.

To obtain position data, students start by placing a mark on the floor to designate as the zero-reference position. They can place pieces of tape on the floor (or mark position with a dry erase marker) at the vehicle’s position every time the metronome ticks. Younger students especially may not have the coordination needed to get good data from the first few tries. Students should be reminded to perform multiple trials with at least 6 data pairs/trials. Averaging the values of position helps them develop a sense of the precision they should carry through the analysis.

**Post-lab discussion**

Students should display their data tables on whiteboards and discuss patterns, observations, and predictions and conduct a board meeting. The goal of the board meeting is for students to come to a consensus about the pattern they observed. Not every buggy moved at the same speed, so groups should not agree on the exact amount added to the buggy’s position each tick, but they should agree that the buggy’s position at the next tick can be found by adding some amount to its current position. Mathematically, this can be expressed as:

|  |
| --- |
|  |

Focus students’ attention on how the position of their buggy changes between each tick. Then have each group predict the next position value after another tick after their last data point and discuss methods. If time allows, have students test these predictions. Students added delta-x to x to form a constant rate of change for the position of the buggy. This conclusion is important as it will form the basis for the Pyret simulation reproducing the motion of the buggy. Then move on to Activity 2 where they will use this idea to write a function in Pyret. They should keep their whiteboards or take photos/notes if you need to wait until the next class for the next activity.

## Reading 1: Writing Functions in Pyret

This reading should proceed Activity 1 and would make a good homework assignment to read between conducting Lab 1 and doing Activity 2. The goal of this reading is to introduce students to the syntax used in writing function contracts, examples, and the bodies of functions in Pyret and provide them with practice doing so before moving on to simulate the motion of the buggy. It also introduces the students to the **Design Recipe**, a powerful template document which they should use every time they write a new Pyret function.

Below is a sample solved code for the exercises which go along with the reading. Copy and paste this code into a Pyret program to make it easier to read. The full code can be found here: <https://tinyurl.com/yc554898>.

|  |
| --- |
| #######################################  # Sample Function 1 - double-number #  #######################################  # A function which consumes a Number x and produces a Number which is twice as large  # Contract  double-number :: (x :: Number) -> Number  # Examples  examples:  double-number(10) is 20  double-number(0) is 0  double-number(7.5) is 15  end  #Instructions  fun double-number(x):  2 \* x  end  #########################################  # Sample Function 2 - find-difference #  #########################################  # A function which consumes two Numbers x and y and produces a Number which is the difference between them.  # Contract  find-difference :: (x :: Number, y :: Number) -> Number  # Examples  examples:  find-difference(10, 10) is 0  find-difference(5, -5) is 10  find-difference(20, 10) is 10  find-difference(10, 20) is 10  end  #Instructions  fun find-difference(x, y):  num-abs(x - y)  end  #######################################################################################################  # HINT #  # There is a predefined function called NUM-ABS in Pyret for finding the absolute value of a number #  # #  # num-abs :: (n :: Number) -> Number #  #######################################################################################################  ######################################  # Sample Function 3 - find-average #  ######################################  # A function which consumes two Numbers a and b and produce a Number which is their average  # Contract  find-average :: (a :: Number, b :: Number) -> Number  # Examples  examples:  find-average(0, 10) is 5  find-average(-10, 10) is 0  find-average(50, 100) is 75  end  #Instructions  fun find-average(a, b):  (a + b) / 2  end |

## Activity 1: Simulating Motion

**Introduction**

The goal of this activity is for students to learn how to write a differential function to reproduce the motion of the constant velocity buggy. They will learn how to define the function next-x and use it to tweak a simulation of a moving tumble buggy.

The accompanying worksheet for the activity includes the design recipe used throughout the course. The questions in part I of the worksheet guide students through all the information they need to fill in the design recipe. It is strongly recommended that you keep a stack of blank design recipes somewhere in your classroom for students to take when they need them.

Students should reflect on the pattern found in the buggy lab data, that for each beat a constant value (delta-x) was added to each position to form a constant rate of change. Follow the Design Recipe Performance Notes below to assist students in completing the design recipe.



**Design Recipe performance notes**

There are five sections for students to complete, under each grey bar.

**Design Recipe** - Function to be designed

* Give students this prompt word for word and have them copy it down. For example, “Write a function which takes in a position and produces the buggy’s next position.

**Physical Interpretations**

* Students need to note what each input is and the units it has. For the buggy, we want the position to be the input, measured in either cm to match the physical lab or pixels. The output would be the next position of the buggy, also measured in cm or pixels.

**Contract and Purpose Statement**

* Brainstorm function names as a class. What do we want the computer to do each time it calculates the position of the buggy? The background code requires the function to be named next-x.
* Contract is based on the datatypes of the input parameters and the datatype of the output parameter. Both positions are numbers.
* The description should be in the format “function-name consumes a … and produces …”
* For this function:
  + # next-x :: Number -> Number
  + # consumes a position and produces the next position

**Examples**

* The examples are concrete and help students design their function.
* The evaluates to (right side) should be written as an expression that “shows the work” and lets the computer do the calculation. Though there are times when you want to start from a concrete value before writing the expression.
* Have students circle the parameters that change are different in each example as a way to help them find the variables. Then name them.
* Ask, “If your buggy is at a position of \_\_\_\_ cm, what would you expect the next position to be?”

**Function Definition**

* Now students write the function with the correct arguments named.
* Then write the general expression following the pattern of the examples but with the argument written as a variable rather than a concrete value.
* Students need to circle each variable and draw an arrow to where it was defined. This should be in the parentheses after the function name. For this function, the input/initial positions should be circled as they will become the x in the function.

**“Where’d you get that?”**

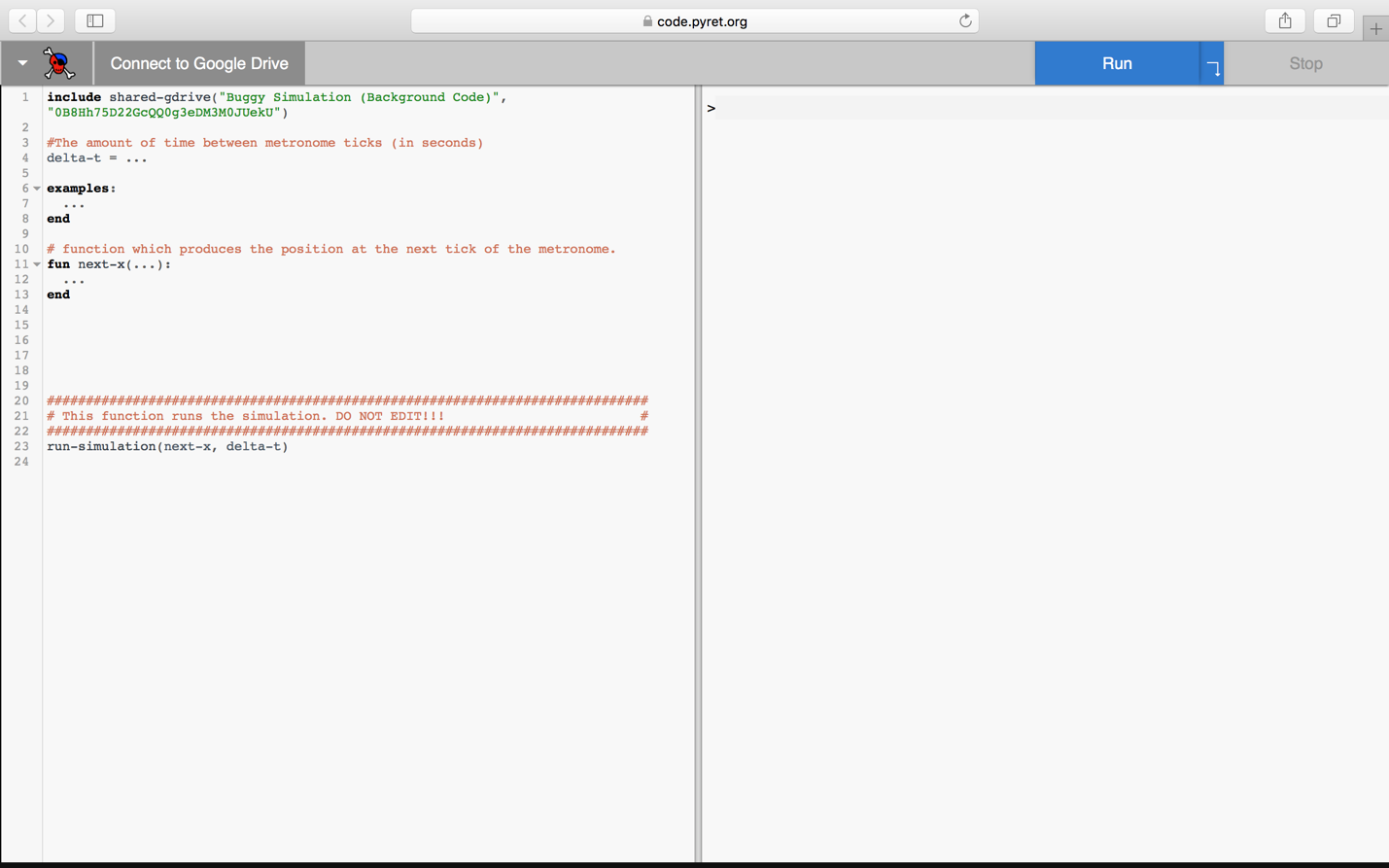
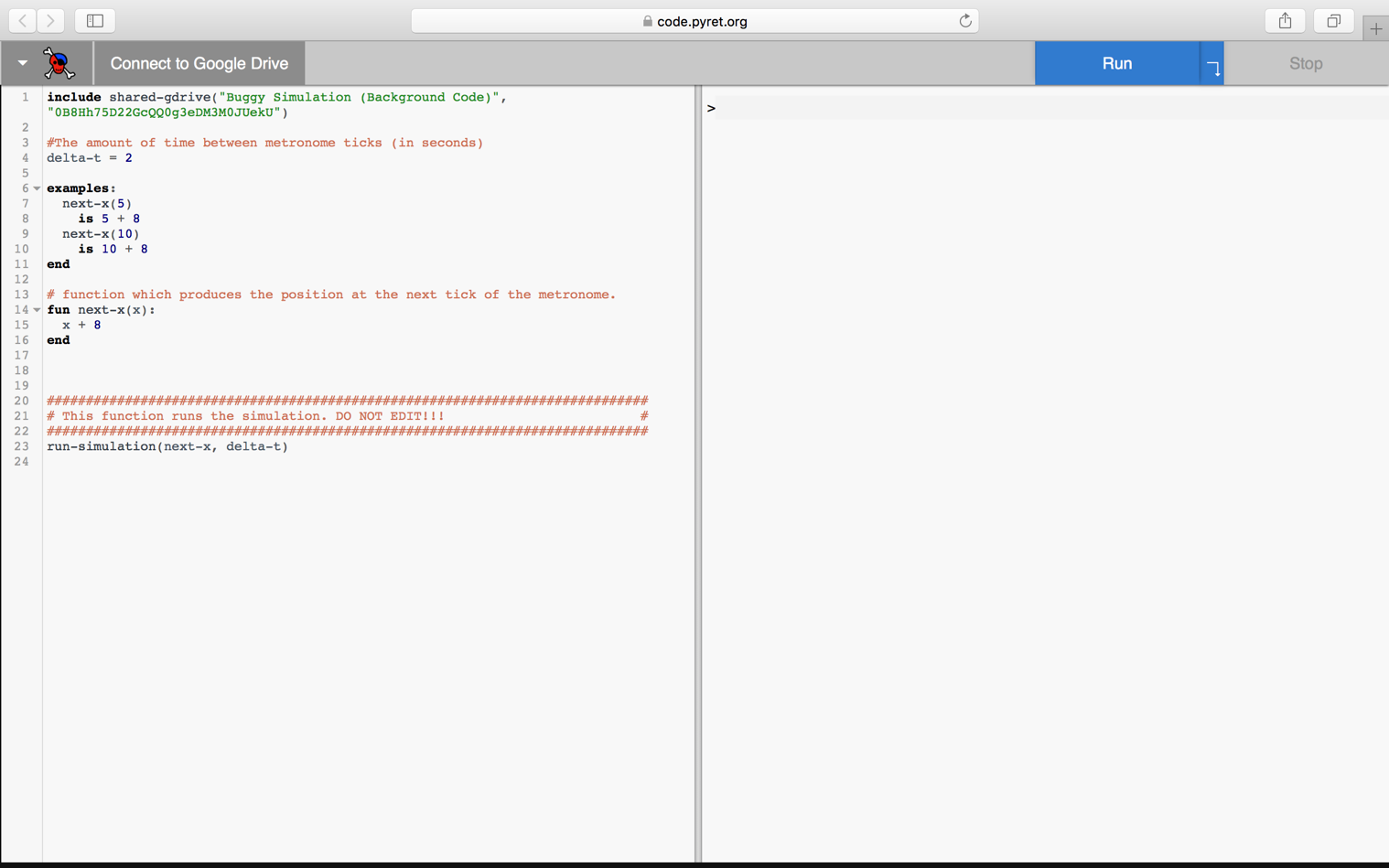
After students write their design recipes, they will defend their work to each other. This exercise encourages students to explain their reasoning in constructing a function.

Student A questions Student B:

1. Students turn to a different partner than they worked with on the design recipe.
2. Student B folds over their design recipe so that only the bottom two sections are showing. Student A starts at the bottom of Student B’s paper and section by section asks their partner “Where did you get that?” as they point to different parts of the work.
3. The responding student must provide an answer from the previous section of the design recipe. For example, the first round Student A asks about the Function Definition, and Student B must provide an answer from the Examples.
4. Then Student B reveals the next section upwards, back the top of their paper, so they can no longer see the functions prompt.
5. Student A now asks, “Where did you get that?” from the Examples section and Student B must provide an answer from the Contract & Purpose Statement.
6. Continue revealing section by section through to the top. Then switch partners.

**Pyret performance notes**

Next have students write their examples and function definition into the student version of the buggy simulation. The full URL for the simulation is: <https://code.pyret.org/editor#share=1rNvhVmHQK8CcNPfQt8j2i4T6x19bW1t6&v=f9e4ffe>



It is important that students pay attention to the **#**, **:**, ->, **is,** and **end**, so that they have the proper syntax. Establish the culture of reading the feedback, thinking about what it is telling you and trying to debug. Don’t be afraid to let students make mistakes in their code. They learn more from getting feedback from Pyret than from getting everything right the first time.

Some conventions to clarify:

* # indicates a comment – this is recognized as text rather than code (comments appear red in Pyret)
* **is** gets used in examples blocks. It is a testing expression which means that two things should be equal.
* ->can be read as “produces”.
* **end** is used to indicate the end of an examples block or function.

Start with students setting delta-t based on what number was used by the metronome such as 2 for 30 bpm or 1 for 60 bpm.

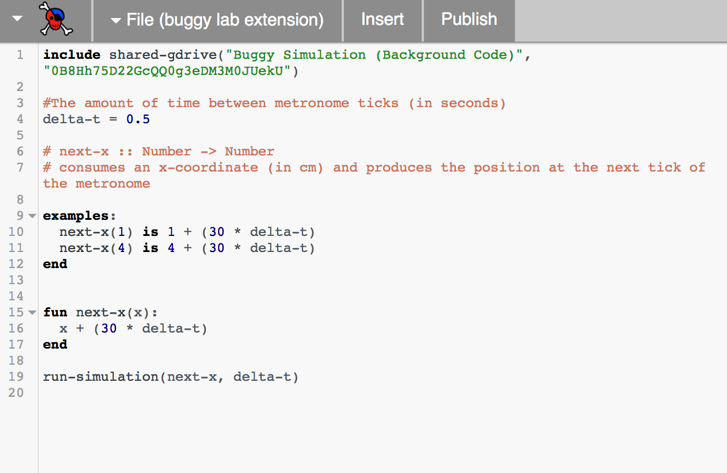
**Function “next-x”**

**Performance Notes:** Again, in pairs with 1 screen between them. Four eyes are better than two for catching mistakes and bugs and it likely helps keep students on task on the computer.

The differential function next-x students need to develop is initially , where is a number which they found in their lab. Have students play around with different delta-t values and see how it affects the motion and the data table. Students should realize delta-t is the time interval of metronome.

When students get their simulation working, they will notice that it doesn’t produce smooth motion, but rather shows a buggy teleporting from position to position. This is because the function to update its position is only running once every time the metronome would tick. This naturally leads students to want to produce smooth motion. Their first instinct to try and fix this would be to reduce that value of delta-t.

Instruct students to change only delta-t. What happens? Is the motion of the buggy the same? The answer is no. The previous representation using x + delta-x has limitations and the representation must be adjusted. Students should see in the table that the buggy’s displacement every one second changes when the delta-t is changed. That means the buggy’s speed is changing because we changed the metronome rate, which does not make physical sense. Have them update their function to incorporate delta-t so that changing delta-t automatically changes the amount by which the buggy moves each tick. Push the students to see that the motion is smoother with smaller time steps, but their code needs a rate of change that can account for this time step. What would happen to their ∆x if the metronome clicked much more rapidly?

Push them to develop a new version of next-x that is an expression that will always give the right amount of movement even if ∆t is changed. They will include the rate of change so that is “how far the car goes in some time interval ” and the next-x function becomes . The idea is they need to use a rate of change of position per second so that at a specific time value the position is the same no matter how many time steps occur. They can rerun the simulation and see that instead of going faster the simulation becomes smoother and they get more data points in the table. You can focus their attention on getting the same position at 3 seconds even if delta-t changes.

Emphasize that this iterative nature of calculations is one of the primary values of computational science. Smaller gives smoother motion that is a better approximation of reality but takes more computational time (hence the invention of supercomputers). Pixar uses a that is small enough for your eye to see fluid motion. Climate scientists use large often because they have a very complex system to represent and they want to go many steps into the future.



Graphically: Using the graph to the right, we will look at how to build the value for xf.

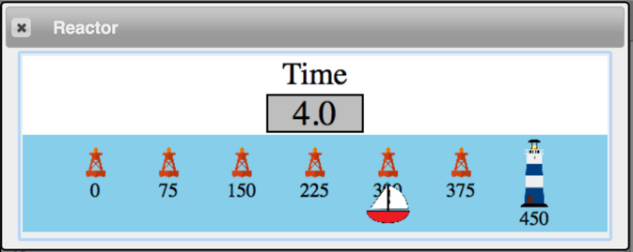
Using the definition of average velocity we know that

So,

(For function-writing purposes, next-x = x + (v \* delta-t)

## Activity 2: Simulating Boats

In this Pyret-based activity students will use differential functions to move a boat from one buoy to the next. Each problem requires a new function for next-x based on the position of the buoys in the water. Student code can be found at <https://tinyurl.com/y723d3da>.

The goal for this activity is to reinforce the computational representation of motion. Students will need to use the positions of the buoys and the desired travel time to calculate the necessary rate of change for the boat. Once students are able to successfully move the boat to the lighthouse in the given amount of time they are challenged to modify function parameters to produce a smoother simulation.

## Activity 3: Motion Maps

The goal of this activity is to develop the idea of ***motion maps*** as a representation of motion. The concept of a motion map should be an easy transition from the computational model of motion which has been developed thus far in the unit. Students are already used to thinking about motion as iterative, with the position of the object changing by a certain amount over equal time intervals.

At this point students should be familiar with the making simulations of motion. The motion map is introduced in the context of taking a screenshot of the simulation at set intervals of time and superimposing them on top of one another, creating an effect similar to a stroboscope effect. These images are simplified to dots for ease of drawing by hand, and arrows are added to indicate the velocity of the object. The code for this activity can be found here: <https://tinyurl.com/y89jru2k>.

## Activity 4: Multiple Objects

The goal of this activity is to get students to understand that the next-x function they have been using in their simulations represents a general rule of motion which applies to all moving objects, rather than a specific rule of motion which moves only a single motion.

This change is both an important conceptual step in the understanding of motion and an important milestone in the coding of motion. In order to accomplish this, students must write their next-x function differently than they have in the past. Previously, students wrote next-x functions which took only one input, the current position x.

|  |
| --- |
| next-x :: (x :: Number) -> Number  **fun** next-x(x):  x + (v \* delta-t)  **end** |

This worked in the past because there was only one velocity which the program could use to calculate the object’s next position. Once you have two objects moving in the simulation, however, the program does not just need to know what the object’s current position is, but also which velocity it is supposed to use in order to calculate the next position. As such, the function must now take two inputs, the current position x and the velocity to use v.

|  |
| --- |
| next-x :: (x :: Number, v :: Number) -> Number  **fun** next-x(x, v):  x + (v \* delta-t)  **end** |

The important thing for students to recognize is that the body of the function itself does not change, only the inputs. The phrasing of what the computer “needs to know” in order to calculate the next position as a way of helping to identify the necessary inputs to this new next-x function. Thinking of velocity as an additional input to this function will be essential to developing a computational representation of accelerated motion. The link to the student code can be found here: <https://tinyurl.com/ycjc9uan>.

## Lab 2: Colliding Buggies

*Rationale*: This deployment lab incorporates computational modeling skills into the more “traditional” CVPM representations when students predict the collision point between two cars traveling towards each other or where one car catches up with another slower one. Students will create a computer simulation that will function as a prototype for the phenomena they are trying to model, which is the collision of the two cars.

This is akin to, for example, NASA engineers predicting a landing point on an asteroid for a spacecraft. They have to tweak parameters such as spacecraft velocities, trajectory angles etc. and run these in computer simulations to land the spacecraft safely. Here students will have to know the velocities of each car and write a computer program to simulate the collisions. This is a simpler version of what NASA engineers may do but it emphasizes the role of computational modeling and its uses in scientific endeavors.

In both the physical activity and the simulation, students should be asked to predict the point and time of collision for the two buggies.

## **Apparatus**

* Fast toy car
* Slow toy car
* stopwatches
* approximately 6m of floor space

## **Teacher Background**

This lab is a more entertaining, more hands-on iteration of the classic “A train leaves from Boston and another leaves from New York” situation. The lab can be done as a whole class lab practicum. The class works as one large group on the problem with little input from the teacher. When the class agrees on an answer, they put the solution (graphical for sure—mathematically if they are able) on the whiteboard. The teacher then calls on one student to present the solution. One part of the entire class grade depends on how well the student presents the solution. The other part of the class grade depends on whether or not the solution works.

Note:

1. If you have enough equipment, instead of having a whole class lab practicum, have different groups of students run the activity separately.
2. You can also have cars play catch up. Different versions of this are:
   1. Slow car starts ahead of fast car and both are released at the same time
   2. Slow car starts ahead of fast car and the slow car is released ahead of time.

## **Pre-lab discussion**

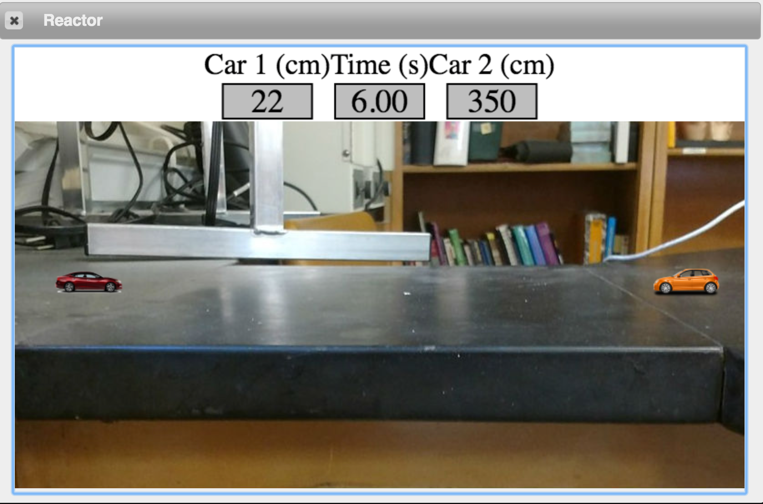
Show the students a diagram of the situation.

Fast car

Slow car

1. Explain that the cars will be moving towards each other at two different constant velocities from a known distance apart. The goal is to determine where they will hit (or at least pass each other) by placing a marker on the floor.
2. Explain how the class lab practicum will work. Students will be allowed to work with only one car at a time. Once data has been gathered from one car and students start work on the other, students cannot work with the first car.
3. Students will first predict where the cars will collide using mathematical, graphical, and diagrammatical representations (motion maps). They will then use Pyret to simulate the collision. This provides students another way to represent their prediction.
4. Have students use the design recipe to write their function. What information will they need to tell the program? The goal is to write next-x(x, v): x + (v \* delta-t) with the initial position and velocity of each car set as initial parameters.

Pyret Collision Student Code: <https://tinyurl.com/yclk7v4k>



This what students will modify:



## **Lab performance notes**

Once predictions have been tested using Pyret, have students run the physical lab.

Students might need hints:

* How could you represent this situation with a graph?
* Is the slope the same for both cars?
* Is the y-intercept the same for both cars?
* Doesn’t this look a lot like the bicycle question from worksheet 4?
* Students should use differential functions to simulate the collision.

## **Post-lab discussion**

If students are unable to accurately predict the collision position or clock reading for the two buggies, conduct a post-lab discussion in which they analyze their procedure for possible errors.

## Activity 5: Graphing Constant Velocity Motion

Goal: Students use data from a simulation to produce a graph of a moving bus’s position over time. Students are introduced to the concept of being able to identify/calculate an object’s position at a specific moment in time.

Using the simulation, students can change the delta-t between each data point and see how that does not change the pattern they can see on the graph. Students can also change the initial position and velocity of the bus and see how this changes the graph produced, leading them to define the physical significance of both the slope of the graph and the vertical intercept. Note that we do not call this the “y-intercept” since neither axis is graphing y as a variable.

The emphasis is for students to articulate that the rate of change is how the dependent variable changes as a result of a change in the independent variable. Having students replace the word “per” with the phrase “for every” is encouraged as a way of fully understanding what a rate of change means, e.g. the bus moves 3m for every 2 second time interval. (refer to Arons’ arguments for avoiding “per)

Students should finish this activity with a working knowledge of the parametric representation of motion: . Function notation should be used and stressed here, since this equation gives the position of the object ***at a specific time t***. Without that, this equation has little physical meaning. Student code can be found here: <https://tinyurl.com/yc3oumqu>.

## Reading 2: Motion and Trajectories

This reading explains the difference between the two mathematical representations that have been developed at this point in the unit. It distinguishes between ***dynamical laws***, which are generative rules which determine the behavior of a system, and ***trajectories***, which are descriptions of the motion that did or could occur under certain conditions. This distinction is crucial in higher-level physics, but is often left unstated in introductory courses. It is addressed here because students are both observing behavior and using simulations to generate behavior, and so they are directly interacting with both dynamical laws.

The reason this distinction is addressed here is to explain why we code our simulations using the differential representation rather than the parametric representation. The parametric representation of motion is not a universal law, but rather a specific solution to the laws which govern the motion of objects under certain conditions. There will be times later in the curriculum where students will be generating motion for which the parametric representation is either unknown or impossible to represent mathematically at the high school level.

## Worksheet 1: Two Bicycles

This worksheet is quasi-quantitative and forces to students to pay attention of other salient attributes of the graph that can help them describe the motion of the cyclists. Some of these attributes are:

1. The intersection of the two lines represents a meeting point, a location of where the cyclists are at the same time
2. The steepness of the graph is an indicator of the cyclists’ speed.

## Reading 3: Number Lines, Distance, Displacement, Average Speed, and Average Velocity

Reading 3 establishes the important differences between scalar (distance and speed) and vector (displacement and velocity) quantities. This common vocabulary will be used throughout the course. This reading can be assigned prior to worksheet 5 or following part 1 of worksheet 5.

## Worksheet 2: Distance vs. Displacement

Part 1 is the introduction to the concepts of distance and displacement. This can be implemented as a class discussion where you project the image on a screen and have students discuss answers. You can also have students work on the worksheet first before discussion.

Have students come to consensus as to who traveled farther. Lead them to think that the answer depends on different definitions of “who traveled farther.” For example, students can say “Dorothy got closer to Oz, but Toto traveled farther.” This provides a mental conflict that highlights the need for two new terms that describe how far something travels: **displacement** and **distance**. Displacement represents the difference between an object’s final position (aka location) and the initial position. Distance represents the “path length” an object traveled from its initial position to its final.

Part 2 provides examples for students to practice these new conceptions. This is suggested as a whiteboard activity. During the board meeting, important points to emphasize are:

* Displacement is either positive or negative
* Distance is always positive

## Lab 3: Graph Matching with Motion Detectors

## **Apparatus**

* Motion Detectors - either a Go!Motion Detector connected to computer or a regular Motion Detector connected to an interface.
* masking tape
* Lab handout: 15-U2\_Lab2-MotionDetect

## **Teacher Background**

The purpose of this activity is to give the students a kinesthetic experience in the relationships between verbal, graphical and diagrammatical representations of motion.

## **Pre-lab discussion**

Show students one lab station.

tape on ground 0 1 2 3 4

with marks

motion

detector

Explain how the motion detector can be used to determine a target object’s position and teach students how to zero the detector.

## **Lab performance notes**

1. You will need to clear as much space as possible for the lab. Desks and chairs need to be pushed aside or removed from the room.
2. Smoother graphs can be obtained by:
   1. Having students hold a whiteboard in front of them to use as a target while they move.
   2. Setting the motion sensor for “long range.” On Pasco detectors, there is a switch on the detector and the “car” symbol represents short-range, and the “person” represents the long-range setting. For Vernier detectors, the “car” symbol represents the short-range, and the “basketball” represents the long-range setting.
3. One of the biggest mistakes that students make when doing this lab is zeroing the detector from whatever position they are standing when they begin each trial. Make it clear that students only need to zero the detector once—when they are standing on the floor at the 0 mark.

## **Post-lab discussion**

There are sufficient problems for each group to prepare a whiteboard and be able to present the multiple representations of the motion described by the position-time graph.

## Worksheet 3: Velocity-Time Graphs

This worksheet is adapted from 2013 AMTA materials. It introduces how velocity vs. time graphs can be extracted from position vs. time graphs, building upon observations students may have found in the optional motion detector lab.

## Worksheet 4: Multiple Representations of Motion

This worksheet is adapted from 2016 AMTA materials and is meant to be whiteboarded. All representations of motion learned thus far are used to construct a multi-dimensional model for the motion of an object. Students should at this point be able to use motion maps, Pyret functions, position-time graphs and velocity-time graphs interchangeably.

1. <http://www.webmetronome.com/>

   [↑](#footnote-ref-1)