**Unit 1**

**Unit 2**

**Feedback March 2**

**# 01-U2 Act1**

**- Tube is odd as a "function" (no distinction between inputs and outputs)**

**- Function examples:**

**\* rectangle uses names, not types**

**\* Input + 10 versus "Input is greater than 100"**

**- "primary uses of computer is to run functions for a user" we disagree**

**- Examples of computer use**

**- Of the three examples: have the students/teachers seen `rectangle`**

**before? How do I know that, from reading these docs? If they \*have\*,**

**then all three functions should be in Pyret notation (input + 10,**

**rectangle(1, 0.5, "solid", "gray"), input > 100). If \*not\*, then all**

**three should be in English (add 10 to input, draw a rectangle of the**

**given width, height, style and color, is input greater than 100).**

**(Meta comment: the dependency ordering among concepts/vocabulary**

**should be obvious or easily discoverable, preferably the unit writeups**

**and/or teachers guides would make this apparent.)**

**This activity has been removed from the completed unit**

**# 02-U2 Lab1 -- respond to teacher demo and record data**

**# 03-U2 WS1 -- analyze data on paper**

**- Q5,6 is an opportunity for an `example`**

**- Q8 is relevant for implementing functions later**

**This worksheet no longer exists**

**# 04-U2 Reading1 -- is this the very first intro to Pyret? Or did**

**something else happen sooner? (The tutorial?)**

**- very first line is exclusivist (it says Pyret has ONLY 4 datatypes)**

**- Do we need more than numbers yet? (Melissa made this point in the**

**U2&3 Development Notes.)**

**- Introduces function contracts without implementing functions OR having activities that use them.**

**- A bunch of small typos**

**This reading has been revamped.**

**It seems like the introductions to Pyret are fixated on units and**

**defining things, but not using them. But coding is meant to be**

**Used…**

**z**

**# 05-U2 Act2 -- Pyret buggy simulation**

**\*\*\* we were initially confused by the sudden mention of the "design recipe": where was this introduced? (it's the \*next\* file)**

**- call out to the DR worksheet**

**The design recipe is now introduced in the previous reading and included here as well.**

**- doc says "next-x", starter code says "update-x"**

**Code has been corrected to now read “next-x”**

**- "There are two things you need. The first is ..." and the second**

**is never even mentioned. Is the second value the slope that was**

**measured in 03-U2 WS1?**

**- image of the ruler should be of higher quality -- can't read the**

**tick marks at all**

**- could "tick" be "metronome-tick" in the table to be even more specific?**

**The activity has been updated to remove the ruler and clarify delta-t/tick**

**# 06-U2 Act2 -- blank DR worksheet with extra questions for physical**

**interpretation**

**- there should be a worked example of a DR-designed function to get**

**them started. a purely blank page is tough for the first try at the DR**

**Moved to integrate with Act1**

**# 07-U2 Reading2 -- motion maps**

**- Conceptual question about motion maps, when coming right after**

**Act2: the first few examples show "a car moving fast" and then "a**

**car moving more slowly", but we just got through saying how we could**

**take more snapshots of the motion (via delta-t being smaller), and**

**since the arrows in a MM don't have any absolute scale (not even**

**relative to the scale bar underneath), I (as an expert) am confused**

**about the meaning of these.**

**- Graphs of motion-maps to position maps: what do these graphs mean**

**without any units or axis labels?**

**- We're confused about the meaning of motion maps, based on the**

**diagram in Teacher Notes U2, Overview, point 5: is this diagram**

**\*wrong\* and the arrows should all be the same length, is the diagram**

**fine and the arrows' lengths are irrelevant and the student**

**interpretation is wrong, or is something else going on here**

**altogether?**

**[This entire section's comments may be implicit physics knowledge we lack]**

**This reading has been revamped.**

**Side note: I have typed and retyped explanations of motion maps several times and am having difficulty explaining them aside from how I craft this for my students. In short, the arrows are only there to show the direction of motion; their length does not matter. The points plot the position of the object each second, on a very simple axis that only shows position.**

**# 08-U2 WS2 -- motion maps (with Pyret?)**

**- "performed one computation per second" is better said (physically and computationally) as "sampled the car's position at a rate of**

**once per second"**

**- this worksheet struck us as weirdly qualitative (e.g. "note that you can't figure out the slope from the graph")**

**- Q5,6 are really interesting! students can't quite answer them**

**yet, is that the point?**

**- Are students expected to write a Pyret function for the differential functions?**

**- If so, have students been taught conditionals and booleans yet? is connecting**

**to that content the intent?**

**- we don't get why piecewise-constant functions are part of the Constant**

**Velocity Particle Model.**

**The piecewise functions have been moved to later in the unit, with the introduction of average speed.**

**# 09-U2 Act3 -- mm with Pyret**

**- no examples block**

**- the second picture struck us as (perhaps amusingly, or intentionally)**

**misleading: the first one has 7**

**lighthouses but 6 seconds, and the distances are exactly right; the**

**second has 5 lighthouses and 5 seconds, and the distances are**

**exactly wrong. The later examples are similar to the second one.**

**Feedback March 6**

**High-level**

* **General concern that programming doesn't feel integrated with all of the concepts -- not that every lesson needs programming (this is logistically challenging, among other things), more that more of a cross-section of concepts should be in the programming activity, which may well only come 2-3 times per unit. We highlighted some examples where there is a corresponding Pyret function that could be written, whether in-flow with the material or saved for later as a call-back.**
* **More doing design recipe worksheets on paper and in-flow to prepare for computer sessions, without necessarily typing everything in.**
* **More scaffolding in computer sessions with call-outs to past sessions**
* **We don't know what was said in class or what lecture accompanies the activities. We're taking the attitude of a teacher who is referring to these materials to prepare their discussion/lessons leading up to these activities. Maybe there is prescribed "audio" that needs to go along with some of these that is causing us to miss context. If so, that should be written down somewhere.**

**10-U2\_WS3**

* **Instead of "Explain why," say "give one reason why"**
* **Why not have 6 & 7 involve writing a Pyret function/doing a DR worksheet, even one that isn't typed in but just helps develop the symbolic form**
  + **Have students come up with one example that *interpolates* and one that *extrapolates***
  + **Use examples to derive symbolic form using "how'd you get that" for the RHS**
* **After question 8, go back to the DR and work through it again, what changes do you need to make?**

**11-U2\_Rdg-3**

* **Axes notation -- the x(m) and t(s) look a lot like invoking a function, not units**
* **Several "we" vs "you" perspective switches in prose**
* **Sample Pyret sim idea -- write two functions, one for each rider, and animate the riders' positions next to one another. Show both the cars in physical context (on the table) and the image being generated one tick at a time.**

**12-U2\_WS4**

* **Does "write a differential function" mean in Pyret style?**
  + **Note, make sure we know (or maybe even link back to!) where differential function is defined, or keep the definition here, or have a glossary sheet**
* **Question 1E/2D -- is "the riders crashed!" a good answer? Does this need more specification?**
* **Question 1A/B -- ahead implies positive x is "better" or oriented somehow**

**13-U2\_Rdg4**

* **5 is used as a constant for distance and time and shows up a ton of times. Not clear if that's necessary for the problems or if it could be made more different to distinguish more.**
* **This is a lot of reading for a 9th grader!**
* **Idea: instead of each paragraph, fill in highlighted cells in this table:**

|  |  |  |
| --- | --- | --- |
| **Measurement** | **Description** | **Value** |
| **Position 1** | **Where it starts** |  |
| **Position 2** | **Where it ends** |  |
| **Distance** | **Position 2 - Position 1** |  |
| **Time elapsed** | **How long did this take?** |  |
| **Average speed** | **Distance / Time** |  |
| **Displacement** | **Final position - position 1** |  |
| **Average velocity** | **Displacement / time** |  |

**Repeat for each trip, then compare highlighted cells to understand the distinctions**

**14-U2\_WS5**

* **The "new terms and definitions" section should be described in the teacher notes (double-check). What are students supposed to come up with here?**
* **Writing the mathematical expression as "6m - 0m" will conflict with the ideal way to write this in Pyret, which would be to write "final - initial", using the defined variables**
* **None of 1-8 have a car moving away from its pointy end. But 9 and 10 require the the car goes into "reverse" relative to its orientation. Even more interestingly, 9 and 10 say that the car makes a *turn* which suggests a change in orientation that the pictures don't show**

**15-U2\_Lab2**

* **Cool!**
* **Same question as for an earlier unit -- it's not clear why *piecewise* constant velocity is part of the constant velocity unit. Should we have a definition of piecewise functions?**
* **Any interest in the other direction, from velocity graph to position graph?**

**16-U2\_WS6**

* **position plot is missing (m) units**
* **Question F! Cool. What are the units of the area is a very interesting question**
* **Should there be a corresponding area question in part 2?**
* **It's interesting to note that the graphical interpretation of speed is just absolute value of velocity. The worksheets don't call that out, but you can do 1F like questions about speed with that interpretation (or with negative-signed-areas)**
* **Question 2F/G call back to WS14**

**17\_U2-Act4**

* **No examples block in starter code; students *will* skip if it isn't there.**
* **Pyret function mandates a signature -- what if it doesn't match the data they collected?**
* **Also, the Pyret code might obviate some of the earlier thinking if a student jumps ahead to the link**
* **When running the code, the reactor at the end shows decimal values, but the display shows only integers**
* **Also, are the students intended to come up with the contract and purpose statement here? If not, this seems like it's going to encourage "hack till it works" rather than design. If so, where are the instructions?**
* **Saying "use the design recipe" in starter code is too late; students will just start typing in code and clicking run. On paper first!**
* **I might scaffold this by giving the contract and purpose statement, and expecting students to write examples that correspond to each car, and then fill in the definition**
* **Make a big point of generalizing from examples to get to the function definition! This is a case where it works beautifully; looking at examples for the two different cars should show both their velocities and positions varying, with the delta-t as constant**
* **Consider using the DR worksheet that has positions for the units**
* **Direction of travel in Part III is not enough information (need magnitude! that is, velocity)**
* **The boxes in Part III.2 should have labels for time and position**
* **Idea: Before Part II**
  + **Plot the data**
  + **Draw an estimated line of best fit (no software for this, just draw it).**
  + **Make a table per buggy:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Measurement** | **Time** | **Change in time** | **Measured Position** | **Measured change in position** | **Position on line** | **Change on line** |
| **1** | **2** | **-** | **0** | **-** | **0** | **-** |
| **2** | **4** | **2** | **5.1** | **5.1** | **5.2** | **5.2** |
| **3** | **6** | **2** | **10.6** | **5.5** | **10.4** | **5.2** |
| **4** | **8** | **2** | **15.5** | **4.9** | **15.6** | **5.2** |
| **Predict** | **10** | **2** |  |  | **20.8** | **5.2** |
| **Predict** | **11** | **1** |  |  | **23.4** | **2.6** |

**(Plus another table, say with velocity -1.4 m/s)**

* **Then use the Time/Position on Line to get the examples. If writing the differential is the main target, then the goal should be to get students to write, say, the following examples, and then generalize to the function body  
  examples:  
   next-x(20.8, 2.6) is 20.8 + (2.6 \* delta-t)**

**next-x(54, -1.4) is 54 + (-1.4 \* delta-t)  
 end**

* + **This is a sophisticated argument! It may be students' instinct to gravitate towards the parametric version (since they measured position/time). The scaffolding should guide them away from that.**
* **Another option would be to write two separate functions, one for each buggy, and then generalize to the version that takes velocity as a parameter. There could even be a version that takes delta-t as an argument to the function, because it could represent table rows more easily.**
* **Overall point -- more explicit scaffolding before jumping into a starter file to make students come up with examples before implementing.**
* **Part II of this in particular is super-cool and I think a really, really compelling example of aligning the computational content with in-lab stuff.**

**18-U2\_WS7**

* **Note that this version takes delta-t as an argument, unlike the buggy simulation. Maybe worth harmonizing/discussing with the prior collision lab? Is it intentional and useful that these have different shapes?**
* **This worksheet has pretty good scaffolded starter code that might assist before the buggy lab. This partially-complete design recipe is a great way to get students to write on paper then move to the computer.**
* **We think something like page 1 here would be great to have *before* the buggy collision lab as a way to scaffold the writing of some next-x differential. While the signatures are currently different (2 vs 3 parameters), having some kind of scaffolded example before diving into the open-ended implementation in the lab seems like a major win. Without this, the buggy collision lab runs the risk of conflating incidental confusions getting a function written and running with the useful confusions of the problem domain and understanding the units of velocity and delta-t, generalizing from measurements, etc.**
* **Is the implementation of next-x intended to be provided here? There's a great opportunity to have students complete the examples and show generalizing from the examples here.**

**Feedback April**

**# 04-U2 Reading1 -- is this the very first intro to Pyret? Or did**

**something else happen sooner? (The tutorial?)**

**- Do we need more than numbers yet? (Melissa made this point in the**

**U2&3 Development Notes.)**

**- A bunch of small typos**

**The language comments we made are cleaned up, and this seems like it's re-ordered, and several typos are fixed, as far as we can tell**

**However:**

**- It is still using strings, which may well be fine, but note that it seems to answer our question about numbers implicitly with "yes, we need more than numbers".**

**- We still don't know if this is the first introduction to Pyret! We think it is, because it's listed as the second thing in the sequence doc. But that was actually one of our main questions/concerns, and I'm not sure if the update is taking a \_position\_ on that choice with some argument backing it up, if that question was missed, etc.**

**There are some little issues that aren't fixed:**

**# 05-U2 Act2 -- Pyret buggy simulation**

**- doc says "next-x", starter code says "update-x"**

**- image of the ruler should be of higher quality -- can't read the**

**tick marks at all**

**I'm not sure if this was changed or not, the starter code inconsistently uses next-x and update-x, causing an error when run. Also, the meter stick image is the same, etc. If the answer is "we updated some things but not others", that's fine. But it's really difficult for us to know which changes or lack of changes represent \_decisions\_ and which might be small omissions we should be reporting again.**

**It seems like there's a bit of a question of feedback \_format\_ – maybe there's something different we could be doing (maybe comment more directly on the files, or make a checklist in a document within the folder)?**

**Unit 3 -**

**- We can't find any links to code in unit 3, either in the exercises or in the**

**teachers' notes or the Instructor Resources folder.**

**Agreed. The links to the code is in progress. We don’t have the final versions yet, but they are very close to done.**

**- From the teachers' notes, as far as we can tell, Activity 2 seems to be about**

**"renaming" the parameter `v` to `v-avg`. We're concerned that this could give**

**students the misconception that the variable name is driving the behavior of**

**the program, which isn't true (the variable name should usefully describe its**

**behavior, which it would have no matter which name we chose). There seems to**

**be missing scaffolding about using a DR process, which functions students are**

**supposed to write (vs just tweaking parameters), etc.**

**Pairing the teacher’s notes with the Worksheet for Activity 2 would show a very different activity than the activity itself. It is not about the ‘name’ at all, but about what AREA of the graph we should use to approximate the displacement more accurately. The students are evaluating three different areas and looking at the way that using the ‘v’ the ‘next-v’ and the ‘avg-v’ values to determine the rectangular areas compares to the actual motion they are trying to replicate.**

**- In Worksheet 2, we don't see the link to code for the highway hazard**

**challenge at the end. Similar scaffolding concerns apply: we're not sure**

**what they're supposed to do.**

**- It seems odd to us that the free-fall sections don't have any Pyret**

**connections: in some sense, this part of unit 3 should have some of the**

**\*best\* Pyret connections!**

**This is absolutely a legitimate concern. The alternative concern is extending the unit beyond the length we can allocate to it. We are constantly battling within ourselves how much depth we can give to any activity versus how much time we can afford to allocate. We are planning a practicum lab simulation where students will be programming a freely falling object and a constant velocity target, and they must coordinate so that the falling object lands on the target.**

**Unit 4**

**Unit 4 --**

**What is the point at which you introduce booleans? Where is it in these materials?**

**- 05\_U4-Act3b: If students return a negative number from Friction-Force, the**

**program crashes. If they type in something other than "left" or "right" it**

**crashes the program. Also, the use of `PUSH = "left"` or `"right"` rather**

**than of using the sign of the velocity, seems not quite connected to the**

**physics. It seems like there's an opportunity to both clean up the code (by**

**specifying the initial direction of motion) and clean up the physics, by**

**tying the sign of the velocity to the friction force's direction.**

**Some suggestions:**

**- An additional function called "is-moving-right" that returns true or**

**false based on the sign of velocity:**

**is-moving-right :: (velocity :: Number) -> Boolean**

**This could be passed to the sim instead of "left" or "right", and used**

**internally.**

**- An extended Friction-Force function that takes velocity as a second**

**argument, and uses it in the conditions (maybe more difficult to program)**

**- 06 -- (Luigi's Pizza) This worksheet should come earlier than the coding in**

**05, right?**

**Yes, the instructions for this worksheet state to place it between activities 3a and 3b.**

**- 10 -- (Weight on different planets)**

**We felt that there are a few missed opportunities here:**

**- Opportunity to reinforce piecewise functions by having students implement**

**the `if planet == "Pluto"` themselves**

**- Did students enjoy it? Sometimes we (Ben & I) think we might discount how**

**much students like the image popping up.**

**Mine did, more than I expected.**

**- It's worth making the point that students can do many interactions with the**

**one function; they don't have to click "Run" each time but could do**

**(in interactions, after clicking run:)**

**> measure-weight("Pluto", 200, weight)**

**{image of pluto weight}**

**> measure-weight("Mars", 300, weight)**

**{image of mars weight}**

**- 12&13 -- this one is actually an experiment! We like this one, though it's**

**not quite true to the original unknowns of Millikan's experiment :)**

**Unit 5**

**Only final touches left to go…**

**I will upload what we have. We are waiting on the sims being finalized though.**