CS 350 Pre-Task 3.0 Answers

Why, in the Mover class, is position defined in 3D using (x,y,z), but then speed is defined in 2D using (speedHorizontal,speedVertical), but then acceleration is defined in 1D using accelerationHorizontal?

Horizontal speed applies to the horizontal x/y plane only. Acceleration applies only to it. Vertical speed is on the vertical z axis.

Is Mover.setHeadingTarget direction supposed to be clockwise when positive?

Yes

Is "Horizontal" defined as the XY plane or the XZ plane? (Is Y or Z vertical?)

x/y

Is turnRate defined by degrees per update?

Yes

Should there be rotational inertia, or should turnRate be able to instantly change?

No.

No, turnRate is constant.

Can it only accelerate towards its current heading, or is it possible for it to "drift" / "overshoot"?

No drifting. It’s not a spacecraft.

Will the Airplane||Mover classes have fields? I assumed as much since there are getter methods but no fields were listed in the API. Just wanted to double check to ensure no compile errors.

Internal implementation is your decision. The Javadoc is the public interface, which has to be exactly as specified.

What kind of identifiers will the airplane have(as seen in getID method)?

Arbitrary.

Does it matter if the mover's speeds are measured in metric or imperial?

There are no units.

Does the airplane class currently utilize the mover class, besides in the get method?

Yes, Airplane has a Mover. Airplane’s update() method calls Mover’s.

Does the Mover/update method need to print any notification to the user of what has been changed, i.e. updated speed, heading or position?

No

In the Mover/update method, we are supposed to update speeds(hspeed,vspeed),position(x,y,z),and heading by "transitioning from current state to the next". Assuming "states" are composed of all the previous fields, how is it checked if a state is changing, i.e. the fields are changing and the state needs to be "transitioned" to a new one?

Update changes the position and orientation of the Mover based on the current configuration (speeds, heading).

I do not have any questions.

Ok

Will ships and aircrafts be present once destroyed? Example: if a ship is destroyed is the object removed from memory in the program, or will it sink and remain a part of the virtual battlefield?

No, none of this is in the story or specifications for Task 3.

How will the simulated war-zone/area be represented in the program? (multidimensional array?)

None of this is in the story or specifications for Task 3.

How will we determine how long different actions/movements will take?

This is too vague.

How is the update() method for Aircraft class intended to be implemented? Mover class? (Is it used to receive updates or give updates to the object representing the map?)

See above.

Will we be putting limits on height/depth of Mover class?

No

How would you like us to calculate turn rate for Mover class?

It’s provided in the constructor, not calculated. It’s degrees per update.

What are the exit conditions for the program/ how do we know when the simulation should be completed?

There aren’t any. There is no simulation. The two classes provide data and control only, no behavior.

Inside the Mover class, the method setHeadingTarget has the boolean variable direction. How exactly does this variable function in the context of how the variables in the other set and get methods will function?

True is clockwise, false is counterclockwise.

As mentioned during lecture, turning does not happen instantaneously but more of small iterations. With this in mind, can we assume that the update method will use a sort of loop?

There’s no looping in the two classes. They respond to calls to update() only and change the state incrementally.

If setSpeedHorizontalTarget(speed) was set to a speed that is slower than the current horizontal speed, how would this affect the horizontal accereation rate of the Mover?

Acceleration is constant as provided in the constructor.

If the target speed is slower than the current speed, then the mover will be slowing down to it.

Is state intialized when the consturctor of Mover is called?

This is too vague.

Does Airplane's update() method call its Mover's update() method?

See above.

How is the z-axis changed during update()?

z is vertical. A call to update() adds the current vertical speed to the current vertical position.

Given the direction is boolean, what does it mean for the direction to be true?

See above.

What does variable heading in the Mover method mean?

Where Airplane is facing on a compass.

If the acceleration horizontal during the next update() would exceed the targeted horizontal speed, should the horizontal speed be incremented as usual but won't change upon the next update() or should the horizontal speed cap at the target speed?

The current speed cannot exceed the target speed, so cap it.

What does heading mean in the Mover class?

See above.

I have no questions

Ok.

Is the speedHorizontal variable in Mover the speed of the vehicle in the heading direction? In other words, does speedHorizontal only affect the x and y values, while speedVertical is the only variable affecting the z value?

Yes and yes.

Does the Airplane update() method only serve to call its Mover update() method?

Yes

What format is the 'String id' in the Airplane Object constructor? Several int's and some Char's?

toString() is not required. Javadoc automatically includes it.

Are we provided with auxilory files to get the code to run? A main method?

No.

Will our Java code be plugged into the visual representation?

No. This is all there is.

1) Is the package name supposed to be cs350s21task2 or is it supposed to be cs350s21task3?

It’s correct as specified.

2) For Pt. 2 of Task 3, are we reverse engineering the requirements to be formatted as what the customer would want?
Ex:
Customer: "I want to be able to get the current vertical speed of the plane"

This would give purpose to the getSpeedVertical() method

Would we then say; "This method gets the current vertical speed of the plane and displays it to the user"?

Yes, but Part II wants requirements, not specifications. It’s enough to say there needs to be a way to query the state of the object, then define what state means in one place. Requirements and specifications don’t usually dictate code-level decisions like method names. (This could happen if there are certification standards in play.)

3) How is the target_speed or target_heading derived? Are these values set by the end user of the program?

They’re provided by the caller.

There is no end user. Users don’t interact with a solution through code.

4) What is the turn rate? Is this measured in degrees or speed?

See above.

5) How will the data in the program change?

Whenever update() is called, the current state changes by the current configuration (speed and heading) to become the new state.

Will the update method in the Airplane class be used to update the mover and the identifyer?

Yes. But the identifier never changes.

What is horizontalSpeed and verticalSpeed?

See above.

What is a heading?

See above.

Why does the vertical speed not have any methods that involve a target? Horizontal speed has a target (setSpeedHorizontalTarget and also getter), but vertical doesn't. Is that necessary in the vertical case?

Vertical speed is defined as being simpler than horizontal.

Does the update method in the Mover class use the getters to update its state?In the Mover class(getState method), does the vspeed and hspeed mean speedVertical and speedHorizontal or does it mean something else?

See above.

for speedHorizontalTarget, are we setting the new speed into the pre-existing speedHorizontal or into a new variable called speedHorizontalTarget?

Internal implementation decisions are your own choice.

How are we testing if our code is correct?

This is your choice.

What is the output supposed to look like?

There is no output.

What does heading refer to?

See above.

Will the parameters for the mover be passed in as input from user?

From the caller, but not the user. There is no user here. See above.

The update method in airplane class says to update the state of the airplane. Does state refer to the same thing as it does in the mover class getState method?

Yes.

1) What exactly is the heading for the Mover class and its purpose?

See above.

2) Since we are submitting only two files with the mover class and airplane class, is our main not going to be implemented in one of files?

There is no main() in the Javadoc specification, so cannot be in your solution.

3) In order to test our code that the airplane is working properly how much files do we need and what should they include?

Testing is your choice. The Javadoc does not specify anything about this.

4) How come there is a method in the Mover class for acceleration for horizontal movement and not vertical movement?

See above.

5) What is the difference between the getHeading and getHeadingTarget methods?

Heading is the current compass orientation. Target heading is the desired orientation to turn to.

How does the update method in both Airplane and Mover work?

See above.

What is meant by "transitioning from the current state to the next" in the context of this program?

See above.

How do x, y, and z values in Mover change?

See above.

In the Mover class there is a getHeading() method is this for getting the direction or target?

It’s the current heading.

Can you give an example of how turn rate would work?

See above.

I don't have any questions.

Ok

I understand what is being asked within the assignment. As well as what the javadoc is explaining. Do I need to have an understanding of javascript and HTML to fully understand this project?

There’s no Javascript or HTML involved, other than this is what the Javadoc uses.

How much outside research is required to do this task?

None required, but if you can’t remember basic trig, looking it up could be.

Can we work with others?

No, it’s an individual task.

1. Do we need to consider every dimension for each element?
i.e. we have getAccelerationHorizontal() but no getAccelerationVertical()
setters for each

See above.

2. turn rate in angle/second? any standard unit?

See above.

3. frequency of update?

However often the caller calls update().

4. getState() need to include other variables like acceleration or direction (might be included in heading?)?

Only as specified.

5. Any other methods other elements other than id that separate airplane from being a mover?
i.e. classification as a fighter/bomber etc...

This is too vague.

Q1. What happens when the Airplane updates vs when the mover updates, does the airplane call the mover update?

See above.

Q2. How should we define a heading in terms of a double? Radians, degrees, a scale from 0-1, ect..? Is it only positive?

Internal implementation decisions are your own choice.

See above.

Q3. Should we assume the mover will move horizontally across the X Z plane and vertically on the Y axis?

See above.

Q4. What will be our "North" or reference point for a heading in terms of the X Y Z coordinate system?

See above.

Q5. For setHeadingTarget(double heading, boolean direction), does it matter what direction the direction boolean correspond to? (False = counterclockwise vs True = counterclockwise)

See above.

What does "update" do for both Mover and Airplane exactly?

See above.

How are we defining the headingTarget, speedHorizontalTarget, and headingTargetDirection?

They’re the desired heading and speed, and which way to turn to get to the heading.

A mover has the attributes x, y, z, heading, speedHorizontal, speedVertical, accelerationHorizontal, and turnRate, but the number of lines of code for the constructer indicates one or maybe two more attributes, what are they?

The statement count is not part of the specifications. Think about else might need to happen in the constructor to create a complete object.

The package is named "cs350s21task2" in the documentation despite this being task 3, is this still correct?

See above.

How should the code be tested? Should we create another java file or create a "main" method in Airplane or Mover, or is there an existing test file?All all instance fields for both classes (Airplane, Mover) meant to be private and only accessible via getters and setters?

Testing is your choice. The Javadoc does not specify anything about this.

Is each Airplane intended to accept any reference of a Mover instance, that is, even if a Mover is already a reference for another Airplane or Object?

Testing will play fair. This is a legitimate concern where two Airplanes could share the same mover, but there is no way to prevent it.

How should an Airplane class instance represent itself as a String returned via the ‘toString’ method?

See above.

Does the Airplane class’s ‘update’ method simply call its set Mover instance field’s update method?

See above.

Does the Mover class simply process vertical speed changes as being instantaneous (as it does not have an accelerationVertical)?

See above.

Are all instance fields in the Mover class that concern distance (x, y, z, speedHorizontal, speedVertical, accelerationHorizontal) assumed as using all the same unit type (i.e all meters or similar)?

See above.

Are all instance fields in the Mover class that concern change over time (speedHorizontal, speedVertical, accelerationHorizontal, turnRate) assumed as using all the same unit type (i.e all seconds or similar)?

See above.

Are all instance fields in the Mover class concerning direction (heading, turnRate) assumed as using all the same unit type (i.e all degrees or similar)?

See above.

Are all instance fields in the Mover class meant to accept any double that they are passed, nulls included?

A double can never be set to null.

Are the instance fields in the Mover class concerning directions assumed as pertaining strictly to the cardinal directional plane only (i.e. North, South, East, West, and no upwards or downwards)?

See above.

Does the class maintain fields for desired values (i.e for desired heading, desired horizontal speed)?

Yes, but internal implementation decisions are your own choice.

Direction is a boolean, what is this supposed to represent? If heading in correct direction true, if not heading in correct direction false? Or is this clockwise(true)/counterclockwise(false)?

See above.

Heading is a double, is this double for the representation of a degrees or radians?

See above.

Is there a specific coordinate system we need to be aware of, or can we assume East is 0 degrees, North East is 45 degrees, North is 90 degrees, North West is 135 degrees, West is 180 degrees, South West is 225 degrees, South is 270 degrees and South East is 315 degrees?

See above.

Can you expand a little on what turn rate does?

See above.

x,y,z do not have setters, I assume we can still change their values through the update method?

Yes, but it depends on what you mean by “we”: is it your solution making the changes itself or you personally? Only one is possible.

Do you already have a main function that will use these classes? We are not supposed to hook up any of the parts other than declaring members and methods for each class and can assume your program will initialize everything?

See above.

Internal implementation decisions are your own choice.

I do not have any questions as of yet the assignment was thoroughly explained. I know they will form as I begin Task 3 Part 2

Ok

I do not have any questions about the content of Task 3.0 at this time.

Ok

In regards to the 26 elements we looked at in Task 1, it seems like the Aiplane/Mover classes can apply to anything that has the ability to move on an X, Y, and Z axis. Is this correct?

Yes

In relation to the aforementioned question above, since the Mover class implements methods that relate to both horizontal and vertical movement, is this an 'either or' situation or does the element from Task 1 have to perform both? For example, a battleship cannot move vertically, on horizontally.

Mover is general. If it provides the capability to an element to do what's needed, then this is an appropriate match. There's no way here to prevent what's not needed.

When you mention in Part 2 'cross-reference indexing or use of shall', what do you mean by this exactly? I'm struggling to understand and I may have overlooked/missed something.

Don't do anything like 1.1.2.1.A.3 or try to get too fancy with the wording. Just state what has to be satisfied in short, comfortable English.

If we fail to execute the anticipated result from the assignment due to lack of understanding, then I would assume that at the very least compiling code will receive some credit? I say this because these assignments are very abstract in comparison to my rigid way of thinking.(To a degree this expected when learning something new)

There will be multiple opportunities to get the correct results.

In the Mover class, what is considered to be a turn rate? I haven't heard this term/phrase before. Like the degree of the turn or...?

See above.

Is our aim to apply these classes to as many elements in Task 1 as we can?

See above.

1. Should we make a basic Mover class and have a more specified subclass for Airplanes (ex. AirMover)?
 - This should allow for move flexibility in the future for more vehicles such as boats (ex. make a BoatMover class)

Exactly as specified, nothing more, nothing less.

This is really my only question. I am mainly wondering if we should design our program for future adjustments, or simply make it fit the requirements for this specific task.

See above.

How should the Mover update function update the speed, heading, and position?

See above.

What should the tostring function print for both the Mover and Airplane classes?

See above.

I do not have any questions. I believe I have enough information to work with in order to finish the project specified.

Ok

In the mover class for the update method what is meant by transitioning from the current state to the next?

See above.

In the airplane class does updating the state of the airplane mean calling the update method in the mover object for the airplane?

See above.

What exactly is the purpose of the update() method in Airplane? Which states should be updated and what should they be updated to?

See above.

In Mover, what do you define as "horizontal"?

See above.

What coordinate system are we using? As in, left handed or right handed?

See above.

Is y or z the vertical axis?

See above.

Is there a particular unit we should be using for measuring speed, or is it arbitrary in this context?

See above.

1. In the mover class, the method "getHeadingTargetDirection" returns a boolean.
But what direction is represented for true or false? Left --> false, Right --> true? Or vice versa?

See above.

2. I noticed a method for getting and setting the Heading target. Should we store an extra variable for the heading target, which has some initial value?

See above.

3. With regard to the heading. First, are we using degrees or radians?

See above.

Second, is heading calculated clockwise or counter clock wise?
Third, what does that heading start at, 0 and north right?

See above.

4. With regard to the "turn rate". What does that entail? In other words are we turning x amount of degrees in 1 update worth of time?

See above.

5. Ok I think my question 2 might've been misguided but still noteworthy.
Anyways, the get desired heading is related to the "turn rate" and "current heading" correct?

Related is too vague.

For the Inferred Requirements in part 2, are we supposed to deconstruct what the program is trying to execute? Or are we trying to go backwards and figure out why this is needed?

See above.

What is meant be the heading variable?

See above.

In the update method for the Airplane, does the ﷺtateﷺ of the plane just refer to the Mover?

See above.

In the Mover, do all of the target headings and speeds need to be separate variables? If not, then what is the difference between the getHeading() and the getHeadingTarget() methods?

Internal implementation decisions are your own choice.

See above.

How do we know what the direction of the heading target is? And in the setHeadingTarget method, why is the direction a boolean?

See above.

For the update in Mover, it says transition from the current state to the next. Would the next state just be changing the current heading and speed to the desired ones, and what changes the position since there is nothing passed in?

See above.