|  |
| --- |
| BALL |
| Flowcharts |
| Data Dictionary   |  |  |  | | --- | --- | --- | | Name of Data | Type of Data | The Use of the Data. | | velocity | Float | To get how fast the ball is going so i can update the position | | mass | Integer | To get how heavy the ball is so i can get the force of the ball going down the ramp | | rampAngle | Integer | To get the steepness of the ramp so i can work out the acceleration of the ball. | | friction | Float | To work out the acceleration of the ball as the friction will slow it down the more there is. | | time | Integer | To work out the velocity of the ball using SUVAT as the time is used in v = u + at | | startButton | Boolean | To start the ball moving down the ramp by clicking on it to change it from false to true. | |
| Write part of the code.  Screenshot it and say what its done and justify it.  Run tests you made.  Fix Problems.  Repeat.    Firstly I created a new subclass called ball this is so that it has its own section that I can put all of the code for it in there.    I then added my variables to my class and set them a value this is because i will need them to be used to store the same thing when i ask the user for an input in world class and then need to use them here. These are all global variables because i will need to be able to use them all in different methods within the class.    In the act method i put an if statement for if startButton == true then it will run some code. I done this because when i have my code fully working if there isnt a button to start it the whole of the code will run endlessly and the ball will be always rolling down.    This also means that i need to change the top area where i put the global variables because i need to add another one called startButton and it needs to be a boolean because it has to be either true and it can run or false and it wont run yet. It will be set to false because you will want to be able to set all your data in first before you are able to start running the ball down the ramp.      I added time to the code this is because i will need to have the time when using SUVAT equations to work out the speed the ball needs to be going. I put time++; in the if statement because it needs to start counting from when the ball starts to roll down the ramp. Also when i use the time variable in my SUVAT equations i will need to divide it by 60 every time this is because the act method runs 60 times a second meaning if i need how many seconds its been running i will need to divide it by 60.    I made a new method within the ball class. This method will be mainly used to do all of the calculations so i am able to get the speed to ball needs to be going and also the location of where it needs to be and how much force there is. This is all being done in its own method because it is better for it all to be separate then at the end i can just call the different methods into the act method and it will all run together.    I changed the name of the method to acceleration this is because i am going to use different methods for different calculations so i gave this one a better name. I also changed the void to a double this is because i want there to be an output from this as i will need to be able to use the acceleration within another method to get the velocity I used double instead of int because it means the answers will be more precise as with int they will just round and the end result may be very different to what it should be because of this. The calculation i used is just a SUVAT equation to get the acceleration then i took away the friction as with the acceleration it slows the ball down.    This method is to work out the velocity the ball is going at. This is being done because i will need this to work out where the ball needs to go and so that the ball can move the correct speed on the simulator. I called the method acceleration so that i can use the variable returned from it this is good as it splits the different bits of code up into smaller areas so its easier to understand what you are doing. I also divided time by 60 as the act method is adding it 60 times a second so if i want to get how many seconds it has been i need to divide it by 60 to get the correct amount.      There was an error with the first time i wrote this code because it wasn't allowing me to calculate a double and a float together. To then fix this i changed the all of the floats into doubles as i changed the global variable velocity to a double and also the method into a double. The second problem was that i hadn't put a return on my first version of it so i made one and told it to return velocity.    I then added the method velocity into the act method this is so that it can be ran in the class so when another method is trying to move the ball into its new position it will work as time in it will be increasing. |
| Test Using the Test Plan in that section and mark them as working when they are.  Fix any Problems.  Show screenshots of the section fully working. |
| Write up how the section went. |