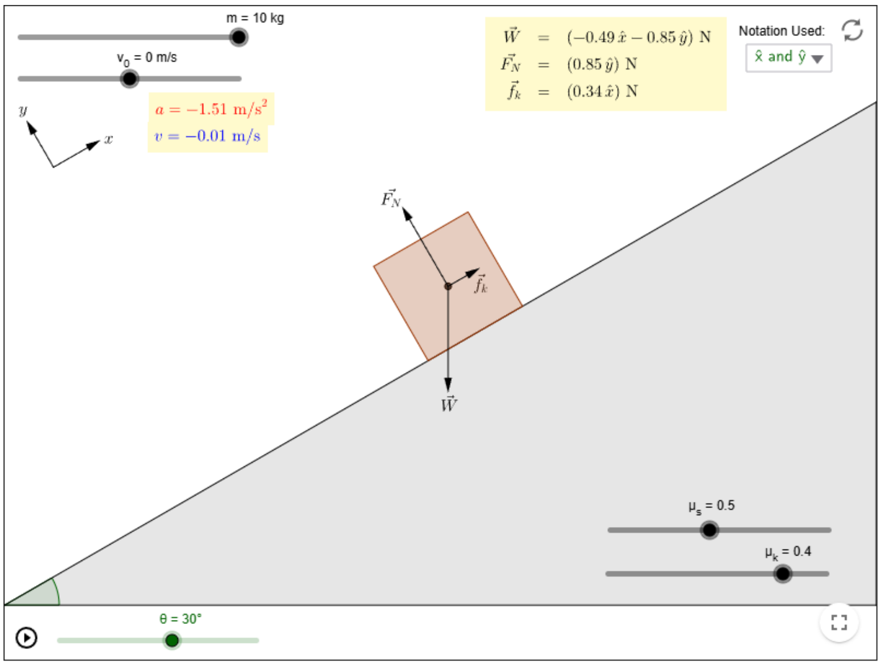
Rample World

My Idea

My programming project will be a type of physics simulator where it will have a ramp and a ball which will simulate the ball going down this ramp with different conditions about the ball and ramp this is most normally used within either A level mechanics maths lessons and A level physics lessons. I am interested in doing this because it is something that relates to me as it is based on mechanics side of maths and it would be able to help people who struggle understanding how all the different parts it relates to each other and will effect each other. I relate to this as i am not that good at this section of maths and i think that this would help me and other who need help understand it more as it can allow people to fully watch the effects of the stuff they work out in maths actually come to life and see what it does. To find out more about my idea i need to find out what type of things they would like to be changeable about the ramp and the ball as i need to know what will help the main people who need this to help them. To find this information out i will look up different sites that already have simulators like this so that i can pull ideas from them and use them for my own. Also i am going to make questions to ask people who are doing physics and maths A level as they are the ones who need the help and what the simulator will be based around i could also ask maths teachers as they will have a lot of experience of what students normally find difficult to understand and what will be needed to make the simulator work correctly. Questions i will ask are. What parts of the ball would you like to be able to change? What parts of the ramp would you like to be able to change? What would you like the simulator to look like? I have asked these questions because they will help to make the simulator and what is actually needed within it.

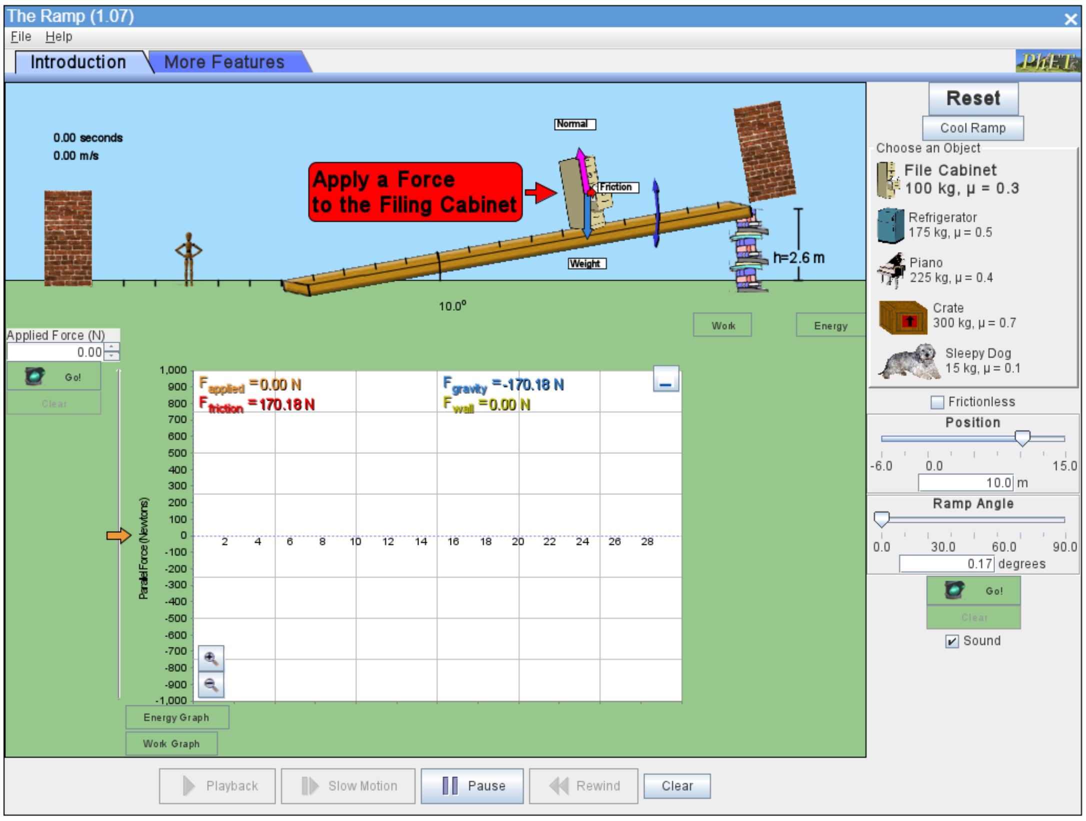
PhET Simulation

Block Sliding Down Ramp With Friction – GeoGebra



This is from Block Sliding Down Ramp With Friction by GeoGebra [Block Sliding Down Ramp With Friction – GeoGebra]

THINGS I LIKE ABOUT THIS GAME	THINGS I DONT LIKE ABOUT THIS GAME
One thing i like about this game is how the actual ramp shown on the screen reflects on what the actual angle you set the degree to. This is because it shows what it would look like and would make it more interesting for the student to use.	One thing i don't like about this game is how to set everything you have to use a slider this is because it would make it annoying to set a very exact number as you would just keep going over it and would be more a distraction than helpful
One thing i like is how all the numbers that you set are then shown on screen and they are not in annoying spaces that would cover any of important parts like the ball and ramp. It also makes everything clear and if you have put anything in wrong you will know and can change it.	One thing i don't like about the game is how you are able to change all of the parts of the simulation while it is running this is because when you are using this for a maths question you are not able to change any of the data while it is going and it only changes when its stated and will be at the beginning of questions not during it.
I like how everything is easy to use and that the style of colours used are simple and makes it all easier to understand	One thing i don't like is how when the ball reaches the bottom of the ramp it then reappears back at the top this is because this then wont show what it would be like in real life as it would just carry on rolling until it slows down or it hits something



0.00 seconds
0.00 m/s

Applied Force (N) 0.00

$F_{applied}$ = 0.00 N
$F_{friction}$ = 170.18 N
$F_{gravity}$ = -170.18 N
F_{wall} = 0.00 N

Position: 10.0 m
Ramp Angle: 0.17 degrees

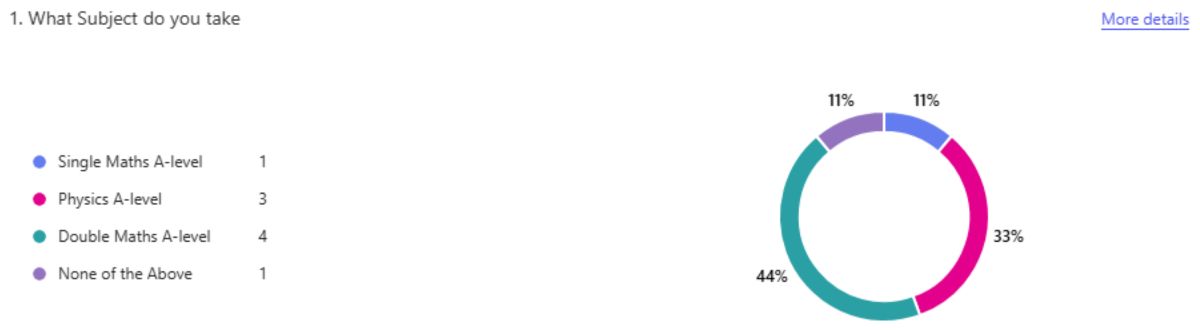
This is PhET Simulation PhET Simulation

THINGS I LIKE ABOUT THE GAME	THINGS I DONT LIKE ABOUT THE GAME
One thing i like about the game is that there is a typing section for when you want to set things like the angle of the ramp this is good because it allows you to set a very specific number and also you dont have to worry about having to find the number with a scroller which is a waste of time.	One thing i dont like about the game is that you can set the position that it starts from this is because i would like them to start at one spot so that they can be compared against each other as for things like A level maths they start just at the top of the ramp and there is no need for a thing to set the position
One thing i like about the game is that there is a wall just away from the ramp this is good because it then stops the ball going on forever which would then cause the simulator to run for a long time for no reason also it doesnt make the ball appear back up at the start with the speed still going which would not be realistic.	One thing i dont like about the game is that you cannot set a specific weight or friction for the object to be and you have to choose what object you want which already has the weight and friction pre set onto them. This is not good because it doesnt allow the settings to be specific for the users preference and for the simulator i just need a ball and not lots of other objects.

One thing i like about the game is that there is arrows that are telling you what to do this is good as if you are new to using the simulator it may be confusing on what you should do.	One thing i dont like about this game is that you are able to interact with the object with your cursor this is not a good thing because it is not what you would do in this actual simulation as using your cursor to add force will not be very accurate to put the correct amount you want and also it allows you to change the force while the game is running.
	One thing i dont like is that the colour scheme i think that there is too much going on and that it may be too overwhelming for the user and may confuse them on what is going on

Stakeholders

For my project the people who are mainly going to be using my simulator are going to be A-level Students who study maths or physics. This is then shown in my Survey


1. What Subject do you take

More details

● Single Maths A-level	1
● Physics A-level	3
● Double Maths A-level	4
● None of the Above	1

This is most likely to be true because many people who would use this would be doing an A-level in one of the two subjects. For the simulator you would need to know what was going on so would need to do either maths or physics A-level. As they are doing their A-levels they would be between the ages of 16-18 and as they would be interested in mechanics and how objects work this is all backed up with the survey where 89% are doing either maths or physics A-levels. The reason why i want the simulator to be set at this audience is because it will be able to help them understand more on how it all works and its set for there age group because in GCSE you arent taught about it much and in A-level its when you first have to learn it and is a quite big section of the curriculum. This will be made use of because it can help students who are struggling with this in there lessons to understand what is going on more and maybe understand it more by actually watching what would happen. They will get involved in the design and what will be used in the simulation and they will get involved by completing a survey i will send out to them which will ask them questions on what they would like so then i can go through them and the most suggested ones can be added to it.

Requirements

<u>User Requirements</u>

Generic concept part

1. There should be a section in which the user can enter the data for the parts of the ball and ramp.
2. The ramp should alter to the correct angle of the entered one also all of the information entered should show up on screen as well as others that have been calculated by the code.
3. Should have a ball that can roll down the ramp and the stop when it reaches the bottom.
4. There needs to be a method in which it calculates everything and then the ball is adjusted to them settings.
5. Have a button that allows you to then go back into the settings and change the data.
6. Have 4 different sections which will be ball, ramp and wall Information.

	Features		Sub-Features	Explanation	Justification	Importance
	Ball	1	Interactable	This will make the ball be able to interact with objects within the simulation	While i was looking at other simulations like the one i would like to make all	Essential

				and will effect each other	of the different objects within the simulator are interacting with each other for example the ball can actually go down the ramp and wont go through it. This is very important as without it nothing would actually work.	
	Ball	2	Gravity	This will make the ball be able to fall to the ground and behave as if it were a real ball and without it the ball would just float	All of the simulators i looked at had gravity in them because it has to be otherwise nothing could work because the whole point of this simulator is for the ball to go down a ramp which means that it has gravity so it can actual run properly.	Essential
	Ball	3	Speed	This will work out the speed that the ball will go down the ramp and then make it go down	This is needed because the simulators main objective is to show what happens with the ball when all of the data is put onto it and is then put at the	Essential

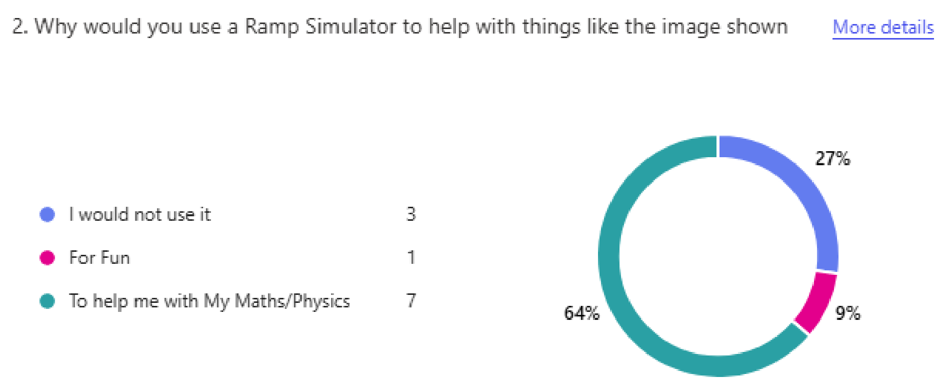
					top of a ramp to see the effect of them all.	
	Ball	4	Appearance	This is so that the ball actually has an image which will be created also this will have a basic design.	This is needed as you need the object to have an image otherwise you cannot see it.	Essential
	Ramp	1	Interactable	This will make the ramp into an actual object that other objects wont just fall through	This is needed because the ball needs to be able to roll down the ramp meaning it must be able to have objects hit it so that the simulator can actually work as it is supposed to	Essential
	Ramp	2	Moveable	This will alllow the ramp to move up and down depending on what the angle entered into the simulator	All of the different simulator that i have looked at had a part that made the ramp able to go higher and lower in degrees this is and it then actually shows and moves on your screen. Its good to have as it can make people understand what going on more if	Important

					its visible to them.	
	Ramp	3	Appearance	This will be used so that the ramp will have an image on the simulator otherwise it will just be an invisible ramp	This will be needed as without it there will be no ramp as it will have no dimension so when you try to add it to the simulator it will just have nothing appear.	Essential
	Informati on	1	Info_enter	This will show all of the information that you enter and print it onto your screen into small boxes.	This was shown on some of the games that i looked at this is good as it allows the user to see what they entered is correct which will be good if they are using it for a question they have .	Essential
	Informati on	2	Main_info	This will show all of the information that is created and used in the simulator that is not entered by the user	This was shown in the simulators as it showed information of the speed while it was going down the ramp and others like that. It will be good to have as it will allow the users who are using this for helping them	Essential

					purposes can see what the data they inputted actually is.	
	Information	3	Calculator	This part of would calculate all of the information that i need for the simulator to work.	I would do this in this section because this is where the user would enter all there data and it would go to and once everything is calculated in here it would be sent out to where its needed.	Essential
	Information	4	Math_or_Physics	This will be a button that will ask you if you are using this for maths purposes or physics	I think this will be a good addition because there are some differences between them both when it comes to the data meaning it will be a good thing to have so you can choose what data you start with	Desirable
	Wall	1	Interactable	This will be used to stop the ball when it reaches the end of the ramp so that it can stop.	This is used in some of the games that i have looked at and i think that it is the best as it is more realistic that it will hit a	Essential

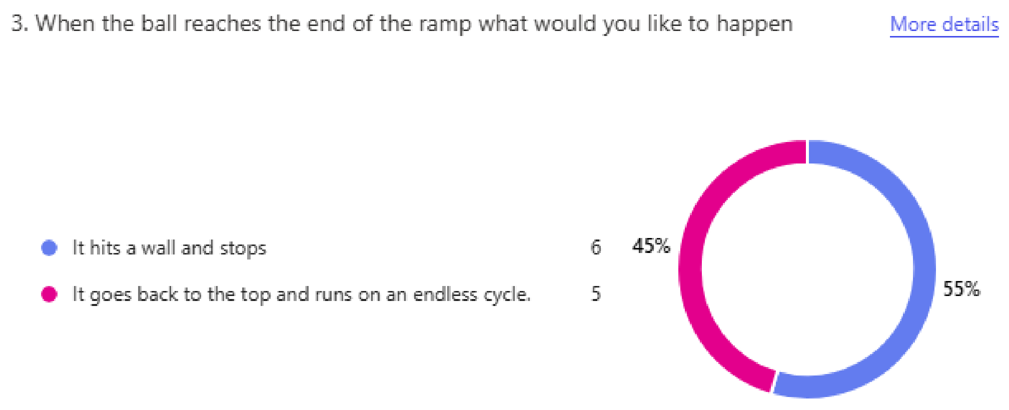
					wall and stop rather than it just gets teleported back up the top and endlessly runs.	
	Wall	2	Appearance	This is so that the wall has an image which will be created	This will be needed as you need the wall to have an image otherwise you cannot see it and the other objects will just hit into nothing and will a simulator where you cannot see anything.	Essential
	Wall	3	Reset	This is so when the ball hits the wall the ball will be reset to its starting position and ask for your data once again.	This will be needed otherwise the ball will get to the bottom of the ramp and stop and then will have no other way of getting back to the start other than restarting the whole simulation	Essential

Forms Quiz


2. Why would you use a Ramp Simulator to help with things like the image shown More details

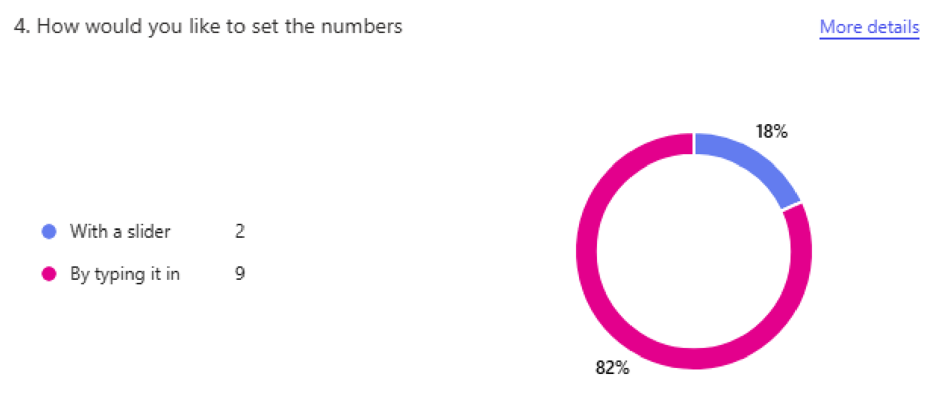
●	I would not use it	3
●	For Fun	1
●	To help me with My Maths/Physics	7

2. This question finds out why people would use the simulator. There was a high majority of people who said they would use it to help them even though there was 27% of people saying they would not use it this may be because they do not take any subjects that relate to it. As the highest percentage of people want to for help i will make sure that the simulator in general is suited for people who are using it for serious things and not for fun meaning i will use more basic colour schemes instead of a range of colours as it wont be mainly used for enjoyment.


3. When the ball reaches the end of the ramp what would you like to happen More details

●	It hits a wall and stops	6
●	It goes back to the top and runs on an endless cycle.	5

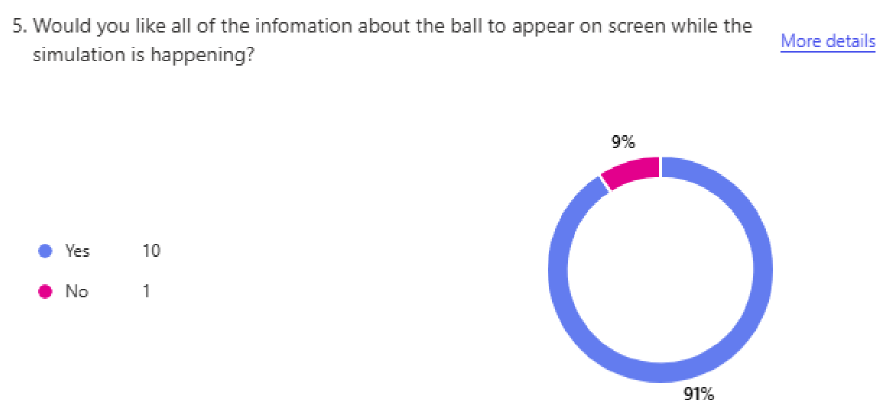
3. This is useful for when i create the simulator this is because this shows how people want the simulator to run and this shows that they would rather the ball to hit a wall at the end of the ramp and stop before then starting the simulator again this is a good option to be chosen because it will make the simulator more realistic because the other choice would cause the speed of the ball to go faster and faster and you would not be able to stop it without stopping the simulator.


4. How would you like to set the numbers

More details

● With a slider	2
● By typing it in	9

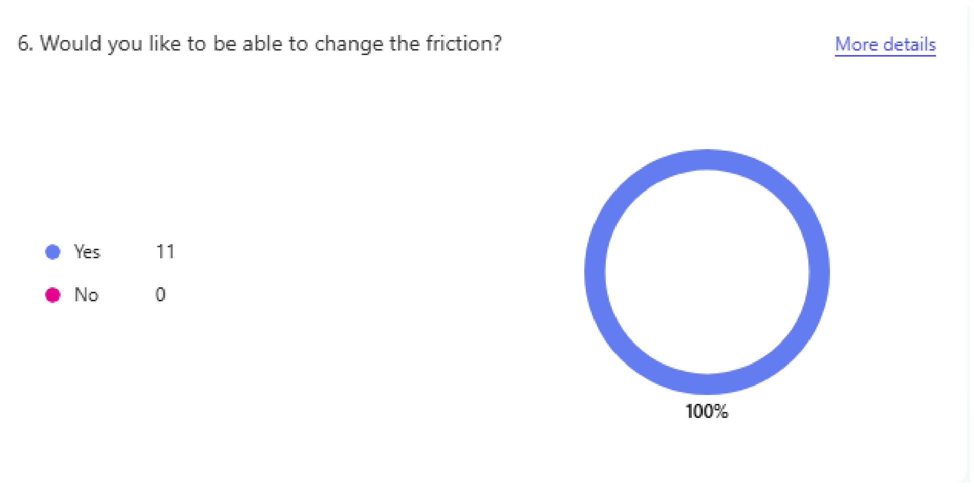
4. This question is useful as it shows that the users would like to enter there data about the ball and ramp by typing rather than using a slider to enter it. This is a good thing as it will allow the users to enter more precise data and also it will be easier to enter the data as with a slider you may go over your number when trying to get to it. This also will definitely be the option this is because it was chosen with a very high choice so will be a top priority to make sure it will be this way.


5. Would you like all of the infomation about the ball to appear on screen while the simulation is happening?

More details

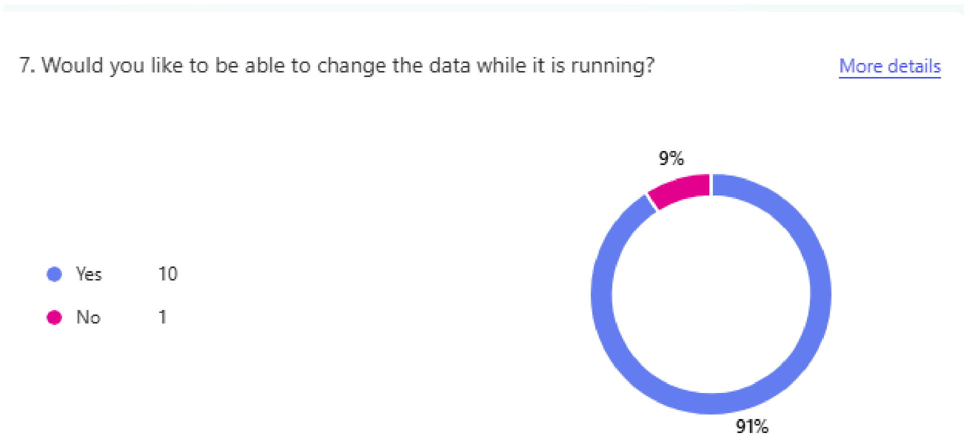
● Yes	10
● No	1

5. This just shows that the user would like the information to be shown to them on screen about what is happening with the simulator. This will be useful as it can help the people who are using the simulator to understand what is going on with the ball and ramp and it will be able to show them what is going on while they watch it for example they will see the ball go down the ramp but they will also see the speed the ball is going at. This is a main priority as it is what the stakeholders and there is a high percentage that would like this.


6. Would you like to be able to change the friction? More details

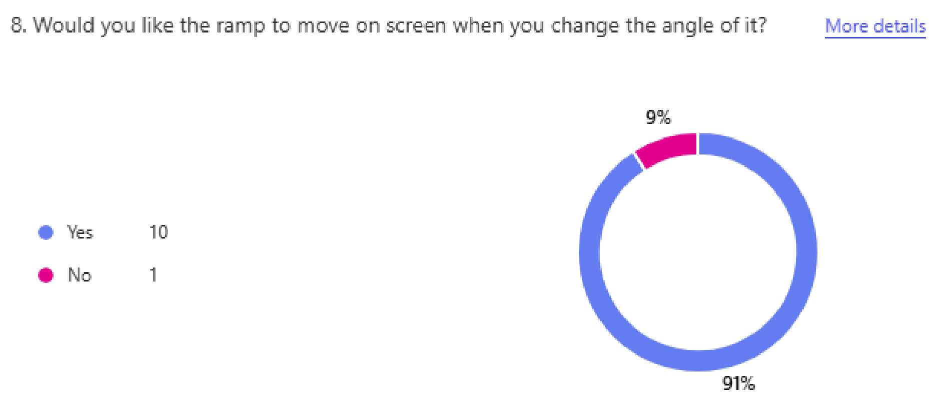
●	Yes	11
●	No	0

6. All this shows is that all of the stakeholders that took the survey want friction to be a part of my simulator. This will be put into the ramp as that is the part that friction is made from. Using friction will make the simulator better as it is more realistic and also it is used in many of the questions given to you in maths and physics. As every single stakeholder said yes to the question and also it is a good thing to put into the simulator it will be important to do.


7. Would you like to be able to change the data while it is running? More details

●	Yes	10
●	No	1

7. This Data shows that the users want to be able to change the data while it is running this is useful because it lets me know that i can have the part where you write the data in running at all times this may be difficult to do but a 91% of users would like it so it will be an important thing to do.


8. Would you like the ramp to move on screen when you change the angle of it?

8. This data shows us that the users want the ramp to visibly move on screen when you adjust the angle of it. This will be a good addition to the simulator as it will make it look more realistic and also it may get people to understand what is happening better. This may be difficult to do as you need to make the image of the ramp to change shape while the game is running however it is a part of it the stakeholders want so it is important to do.

Limitations

Limitation	Explanation
Pully	This would be a good addition however it would be very difficult to code as you would need to have forces then pushing back on the first ball and it would need to know when the pully ball hits the ground. It would also be difficult to design as you would need a string attaching them and which would mean having to have the string object constantly moving with them both which would take a long time to code.

Slider	This is a limitation because only a few stakeholders said that they wanted this and also it would be difficult to create as you would have to make an object that was able to be interacted with and that would send inputs back to the code with information this means that it is too difficult to code for the amount of users who want it.
Rigid Objects	This is a limitation because the simulator isnt focused on how the ball looks so there wont be any settings for it to change if there are bumps in the ball. This is being done because it isnt about the simulator it is as it just wants it to go down the ramp it doesnt need to have anything about deformations in the ball and also in Maths A-level questions there are never any deformations within the balls.
Air Resistance	This is a limitation because there wont be any air within the simulation for me to calculate it and it wont be necessary. If there was it would affect the speed. This is not needed in the simulator because within Maths A-level questions which the majority of users are doing it does not include this meaning when trying to help students it will not be needed.
2D World	This is a limitation because it will not be a 3D world that balls will be

	able to change direction and fall off the ramp. It is not going to be 3D because that when using the simulator the majority of users will want it to be a 2D simulator because the questions that they are getting help for are on paper that are 2D and are always shown and talked about as 2D this means that making it 3D would be very unnecessary as it would not help the users at all and would remove the point of the use of this simulator which is to help them.

Hardware and Software Requirements

Hardware

- The CPU or GPU wont need to powerful to run it you will just need a mid-range one so that it can run images and work out inputs without breaking the computer.
- As this is just a simulator it means that the laptop or desktop you use wont need any additional things added onto it.
- You will need a mouse to be able to click on buttons used in the simulator.
- You will need a Keyboard so that you can enter the data needed for the simulator so it can run.
- You will not need to have much storage as you will only need to download greenfoot to be able to run the simulator and greenfoot is only 200MB in size.
- You will need to have either a desktop or laptop computer that allows you to download greenfoot this is because some devices will not allow you to download greenfoot.

- You will not need a massive amount of RAM as you will just need enough to run the operating system and the simulator without it crashing the system.

Software

- I will be using greenfoot to be creating the simulator and GUI which will use java code it all. I am using this because there are lots of libraries within greenfoot that will be useful to make this and also greenfoot has its own built in GUI meaning people will not need to download multiple things to run the simulator.
- Users will need to download greenfoot onto there device so that the game is able to run.
- You are able to download greenfoot for free on many different operating systems For Windows you need windows 10 or above and you will not need to download any additional software it is also good as the majority of students have a computer at this level.
- You only need to have a basic graphical interface for your device this is because you will not need it to have high definition but you will need one so you can actually see the simulator.

Computational methods

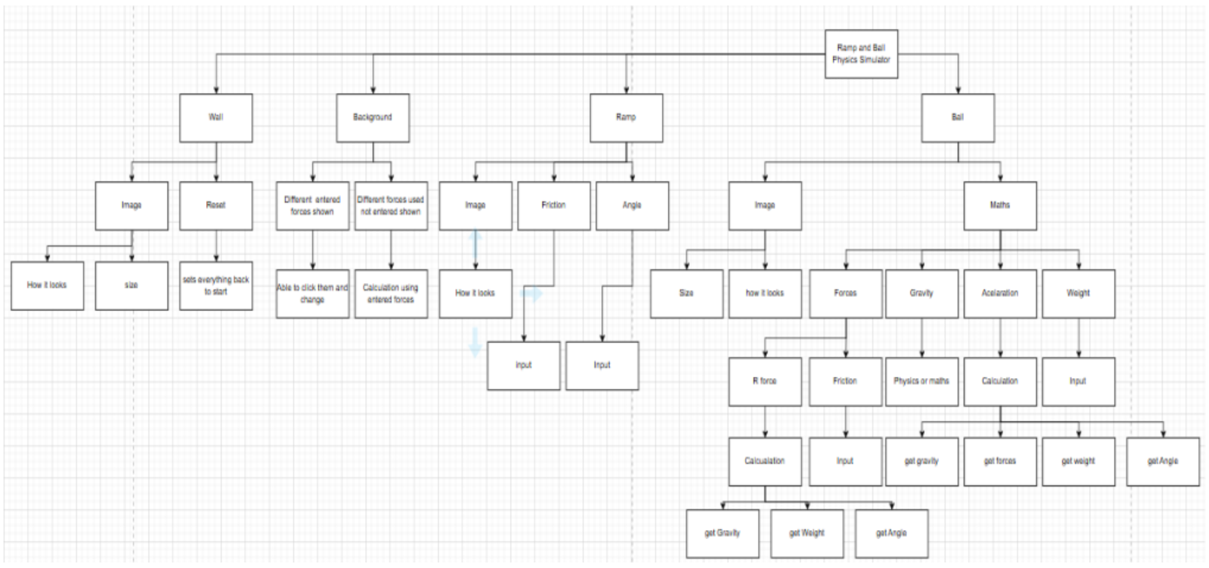
The code will use decomposition this is because i want each of my sections to be split up into many more different sections i want to do this so that i will know exactly what to code in each part meaning there is less chance i will miss things out and i will know when each section is finished this means that it will be more manageable it do examples in the code would be splitting the ball up into different areas so that there's a section for the gravity and another section for getting the speed of the ball.

There will be many tasks and calculations that will occur concurrently in the simulator as the ball will need to be falling down the ramp but also at the same time there will be calculations happening to workout what speed the ball should be going at and also it needs to be printing all of this information onto the screen so the user can see the forces on the ball and ramp and the speed which will be changing and others. Also the simulator will be checking to see if the ball has hit the wall yet so it can send it back up to the top.

I will also use abstraction in my code this is because there will be a lot of parts of the code that will not be necessary to the user an example of this is when the calculations are being made i wont need to show the exact calculation being used i will just show the end answer and what it is. I will not show for example 25cos(60) = 12.5 is force i will just have 12.5 is the force. This means that its easier for the user to understand and also it wont fill up the screen with unnecessary information.

<u>DESIGN</u>

<u>STRUCTURE DIAGRAM</u>



<u>STAGES OF DEVELOPMENT</u>

<u>Stage 1 Designing - Monday 22nd September</u>

Make the designs for the simulator

Creating images for the ramp, the ball, the wall.

Make a image for how the game will look.

I need to have a design for the simulator this is because when i make the simulator i dont need to spent time making the designs then

<u>Stage 2 Creating the ball – Monday 6th October</u>

Get the Image onto Greenfoot

Write the code so the ball wont just fall through the ramp

Write the code gravity and the calculations for the forces and speed so it can roll down the ramp.

Code the ball with its image, velocity, acceleration, force, gravity make it so i can input these. This is so that users can see it and have inputs so that users can customise it all to what they want

Make variables that will connect them with the code. This is so that the inputs can connect to the code to make the ball work

Code the maths so that its realistic. This is so that the ball has the right speed when moving.

Make the ball actually move. This is so that people can see what happens.

<u>Tests.</u>

Spawn in the ball at the top of the screen and see if it falls. This is to test the gravity on the ball.

Set Different Data onto the ball. This is to test to see if the inputs are working correctly.

<u>Stage 3 Creating the ramp – Monday 20th October</u>

Get the image onto Greenfoot

Write the code so the ball will stay on top of the ramp.

Write the code so that the ramp can change angle

Create code to make a image for a ramp. So that users can see it on screen.

Create inputs for the angle and friction of the ramp. So that the user can make it specific to what they want.

Create variable to connect input to code so the ramp can visually change angle. This is so that the ramp can change angle to anything inputted.

<u>Tests</u>

Set Angle to 0. The ramp should go flat

Set angle to 45 and spawn ball and set values. To see if the ball goes down the ramp and at the correct speed and acceleration.

Do one test of friction at 0 and one at 10 with a ball. To see if the ball is faster with no friction.

<u>Stage 4 Create the Wall – Monday 3rd November</u>

Get the image onto Greenfoot

Write the code so that the ball wont go through it. So that the ball wont go on forever.

Write code so that it knows when the ball has hit the wall. So that it can tell the rest of the code.

Write the code so that once the ball touches it everything is reset back to the beginning. So that the simulator can be ran again and the ball doesnt stay at the bottem.

<u>Tests</u>

Place a ball next to the wall at the bottom. The whole simulator should reset to the beginning.

Roll the ball down the ramp into the wall. The simulator should reset and the ball should not just pass through the wall.

<u>Stage 5 Create the background – Monday 17th November</u>

Add the code to make all of the objects appear on screen

Add the code to make the forces and there answers appear on screen.

Add the code to be able to click on the forces that are suppost to change. So that you are able to input the data.

Add the code for a start button. so that you can set up all the number without it already moving

<u>Tests</u>

Click on the changeable data. Screen should appear and let you enter it and should then appear on the screen where you clicked.

Click on the weight changeable data and run the simulator. It should be slower the higher the number.

Make the ball hit the wall. All numbers and the start button should reset back to 0.

Add all the data and click the start button. The simulator should run.

<u>Stage 6 Test – Monday 1st December</u>

Test each part of the code to see if the maths is correct, it outputs correctly, it inputs correctly.

Test it individually and all together.

Set all of the data then hit the start button the check if

Is all the data correct and is the calculations correct. Look at the unchangeable data and then use the changeable data and use a calculator to see if its all correct.

Did the ramp change angle. It should be at the same angle that user set.

Did the ball hit the wall and everything reset. Ball should go back to the top the changeable data should go back to 0 and the start button should be reset.

Make sure everything is running as it is expected to be.

Change parts of it if its not running correctly or want to make it better

Stage 7 Evaluate – Wednesday 10th December

Say what's good and bad about the simulator.

Say what could of gone better.

See how much it has changed from the first ideas.

Use the mark scheme to see where you could do better.

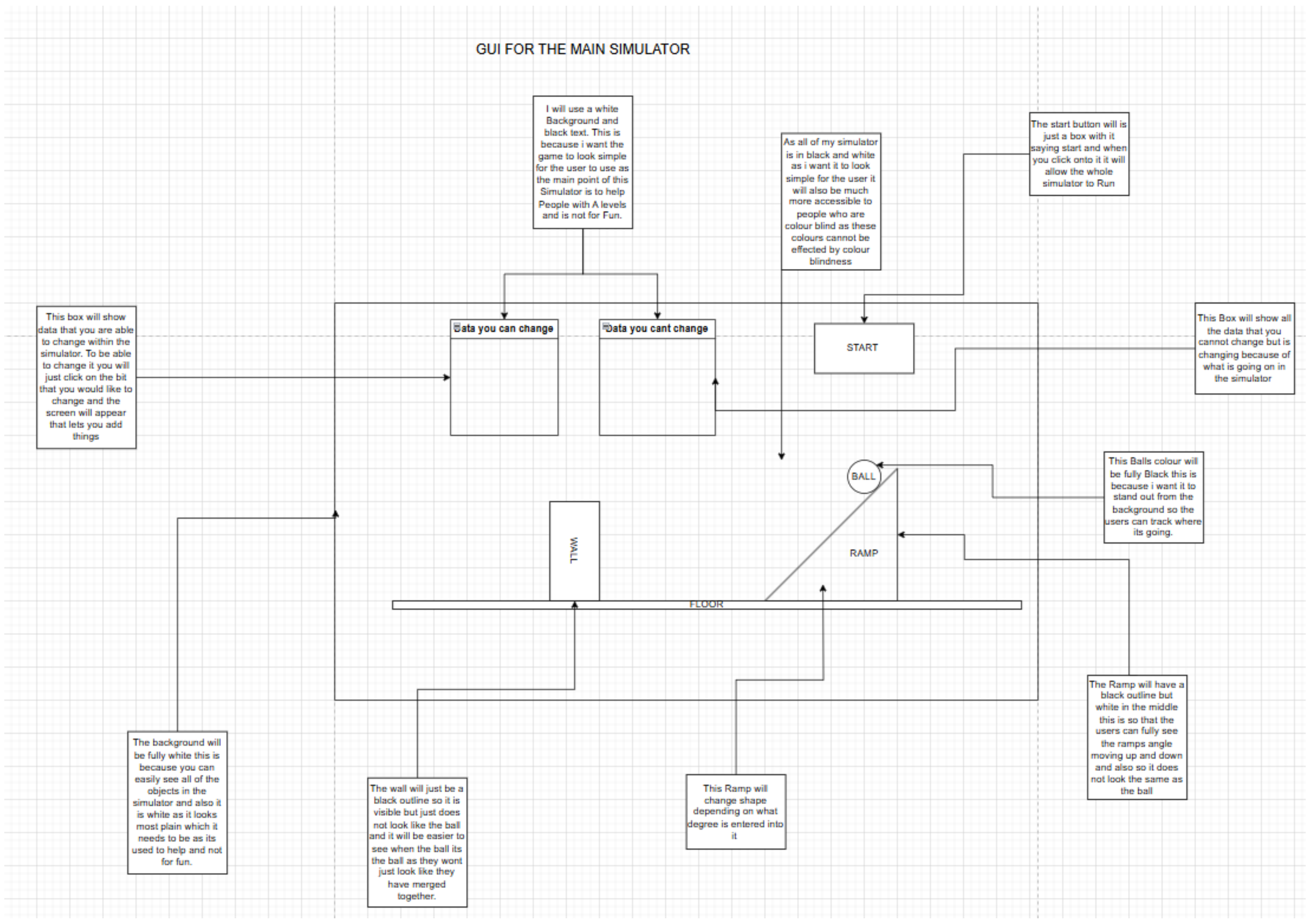
Go back through the writing to add more information or make it look better.

Make sure all the sections are finished.

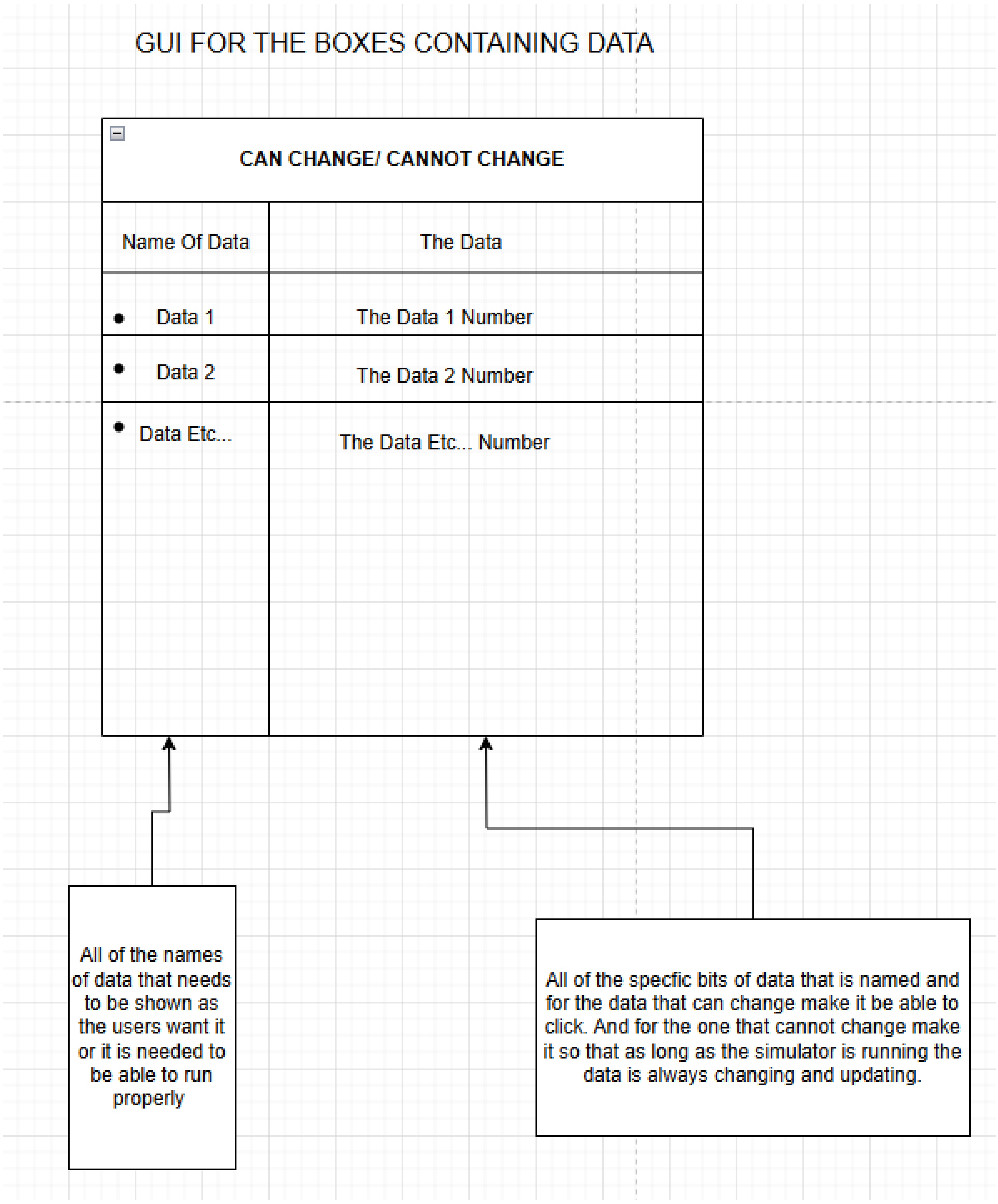
Submit it.

I have done these stages because it gives a structure of doing one section at a time meaning i can do one object then i can move onto the next and not then just doing one section on each then another and its not that ordered. The reason i done this order is because you need to design everything first then start on your code with the ball first because its the most important as you cant test the rest without the ball then the ramp because its the second most important and need that to test the ball then the wall as its not as important as you dont need to have it then i done the background last as its just making sure everything is in the right place and other things are added and doesnt mainly effect the actual simulator. Then i done the main test as i need so i can see if everything is running correctly then evaluation last as everything else needs to be done so i can evaluate.

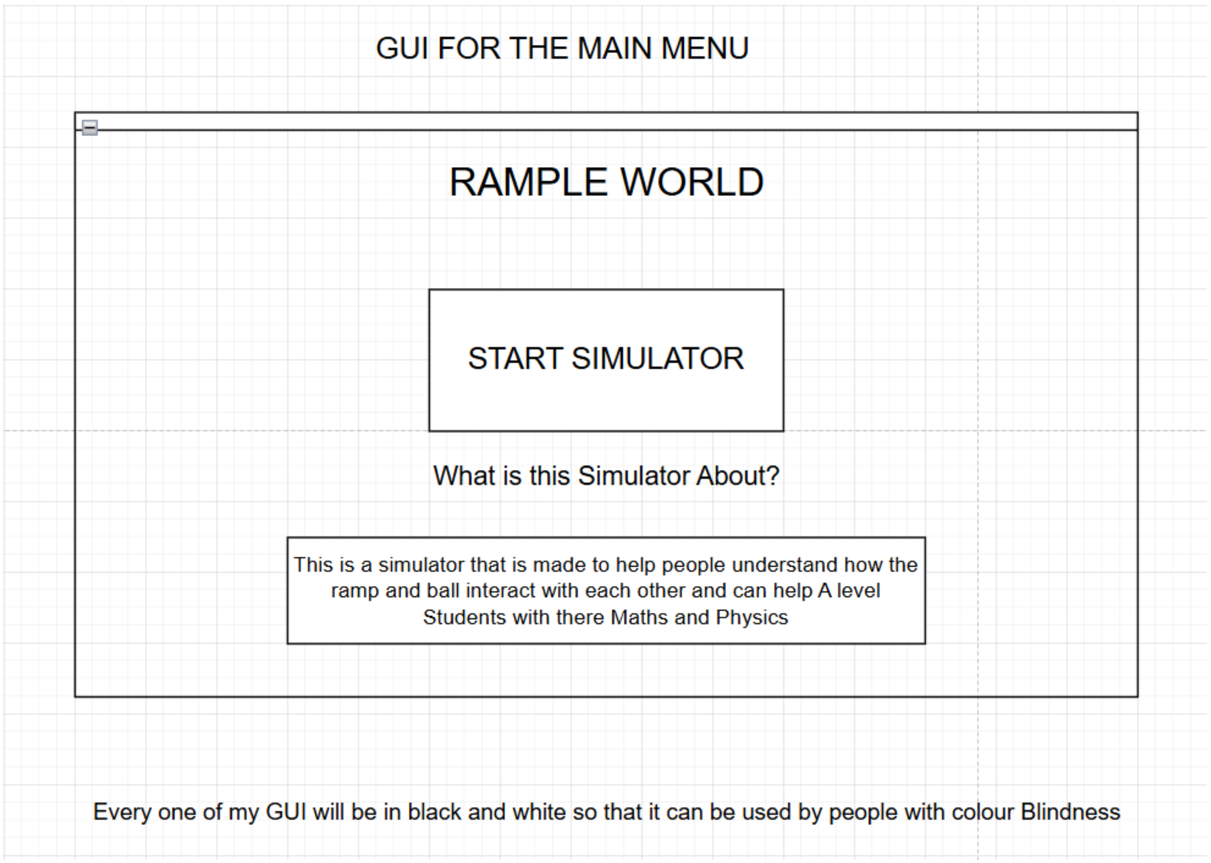
GUI DESIGNS


GUI FOR THE MAIN SIMULATOR
I will use a white Background and black text. This is because i want the game to look simple for the user to use as the main point of this Simulator is to help People with A levels and is not for Fun.
As all of my simulator is in black and white as i want it to look simple for the user it will also be much more accessible to people who are colour blind as these colours cannot be effected by colour blindness
The start button will is just a box with it saying start and when you click onto it it will allow the whole simulator to Run
This box will show data that you are able to change within the simulator. To be able to change it you will just click on the bit that you would like to change and the screen will appear that lets you add things
Data you can change
Data you cant change
START
This Box will show all the data that you cannot change but is changing because of what is going on in the simulator
BALL
This Balls colour will be fully Black this is because i want it to stand out from the background so the users can track where its going.
WALL
RAMP
FLOOR
The Ramp will have a black outline but white in the middle this is so that the users can fully see the ramps angle moving up and down and also so it does not look the same as the ball
The background will be fully white this is because you can easily see all of the objects in the simulator and also it is white as it looks most plain which it needs to be as its used to help and not for fun.
The wall will just be a black outline so it is visible but just does not look like the ball and it will be easier to see when the ball its the ball as they wont just look like they have merged together.
This Ramp will change shape depending on what degree is entered into it

I have set my GUI main screen out like this because it has all the different things in places to show how important it really is like the start button is in the top right corner as it is not that important to the actual simulator but is needed to make the game actually run. I have put the 2 boxes of where all of the data goes at the top in the centre this is because it is the main part of where you enter your code and also see if the data is coming out correctly but also not in the centre as it is not the main part of the screen. Then Finally I put the simulator in the whole of the middle of the screen this is because it needs to be long enough to actually show its all running correctly and its in the middle as it is the main part of the project so it needs to be the main thing the user focuses on.


GUI FOR THE BOXES CONTAINING DATA
CAN CHANGE/ CANNOT CHANGE
Name Of Data
The Data
• Data 1
The Data 1 Number
• Data 2
The Data 2 Number
• Data Etc...
The Data Etc... Number
All of the names of data that needs to be shown as the users want it or it is needed to be able to run properly
All of the specfic bits of data that is named and for the data that can change make it be able to click. And for the one that cannot change make it so that as long as the simulator is running the data is always changing and updating.

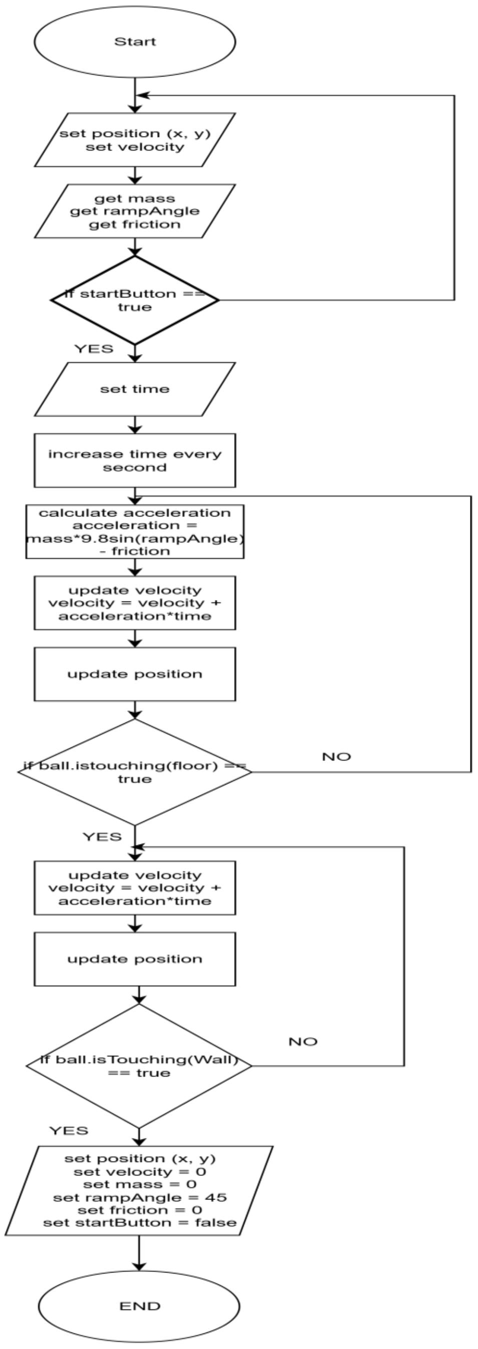
This GUI is just a zoomed in on the data boxes from the main screen. I have a box at the top to indicate if it is the data you can change or cannot change this is because there are 2 different boxes and one will contain data that only changes on what you enter into the other box. This makes it easy for any user to understand what is happening in that box. I then split it up into 2 different sections one is the name of the data and then next to it there will be the actual data that will be gotten from the other classes. This is good as it is easy for a new user to understand what data goes with what and if there in the box that you can change they will easily know what they are updating.


GUI FOR THE MAIN MENU
RAMPLE WORLD
START SIMULATOR
What is this Simulator About?
This is a simulator that is made to help people understand how the ramp and ball interact with each other and can help A level Students with there Maths and Physics
Every one of my GUI will be in black and white so that it can be used by people with colour Blindness

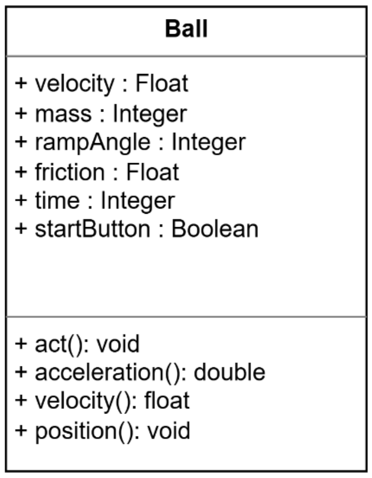
This is the Start screen from where you can click onto the simulator. I have the start simulator button in the middle of the screen because it's the main part as all this pages is for is to say what the simulator is about and who it can help so that bit is just a small section at the bottom of the screen.

<u>WRITE UP</u>

BALL
<u>Flowcharts</u>



<u>Class Diagram</u>

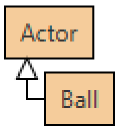


Data Dictionary

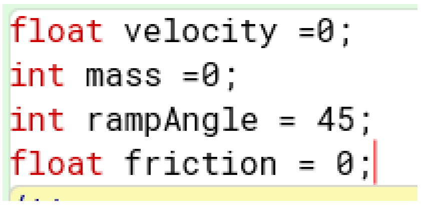
Name of Data	Type of Data	The Use of the Data.
velocity	Float	To get how fast the ball is going so i can update the position
mass	Integer	To get how heavy the ball is so i can get the force of the ball going down the ramp
rampAngle	Integer	To get the steepness of the ramp so i can work out the acceleration 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.

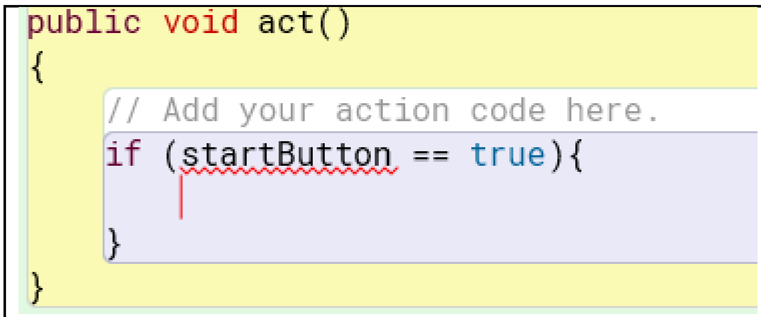


Firstly I created a new subclass called ball this is so that it has its own section that I can put all of the code for it in there.



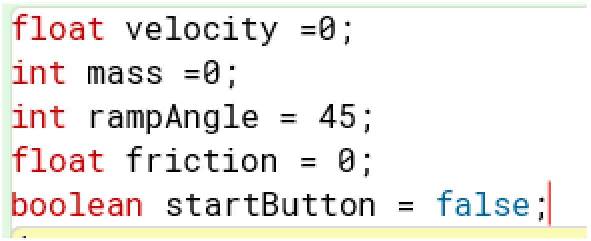
```
float velocity =0;
int mass =0;
int rampAngle = 45;
float friction = 0;
```

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.



```
public void act()
{
    // Add your action code here.
    if (startButton == true){
        
    }
}
```

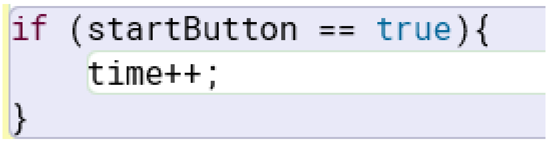
In the act method i put an if statement for if startButton == true then it will run some code. I done this because when i have my code fully working if there isnt a button to start it the whole of the code will run endlessly and the ball will be always rolling down.



```
float velocity =0;
int mass =0;
int rampAngle = 45;
float friction = 0;
boolean startButton = false;
```

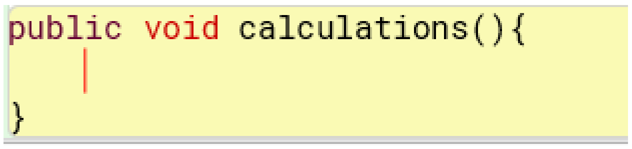
This also means that i need to change the top area where i put the global variables because i need to add another one called startButton and it needs to be a boolean because it has to be either true and it can run or false and it wont run yet. It will be set to false because you will want to be able to set all your data in first before you are able to start running the ball down the ramp.

```
int time;
```



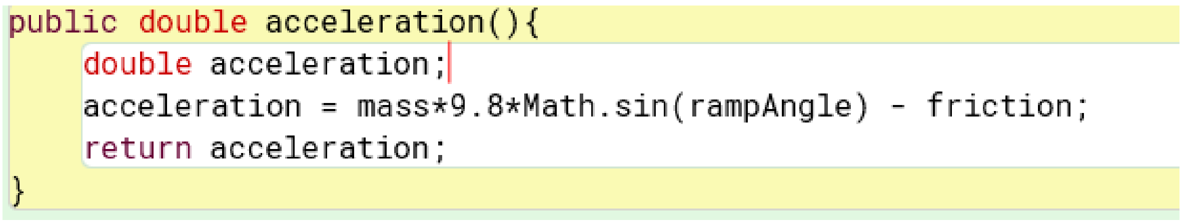
```
if (startButton == true){
    time++;
}
```

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.



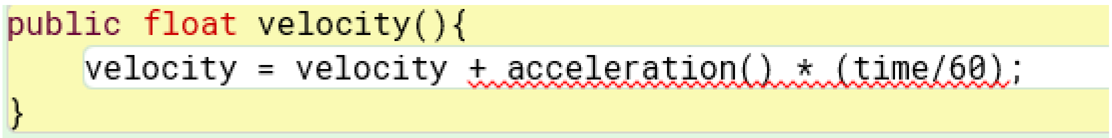
```
public void calculations(){
    
}
```

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.



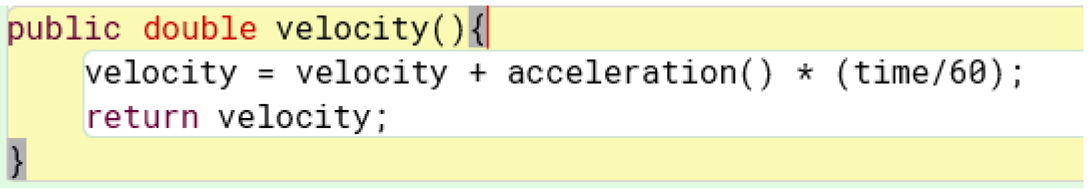
```
public double acceleration(){
    double acceleration;
    acceleration = mass*9.8*Math.sin(rampAngle) - friction;
    return acceleration;
}
```

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.

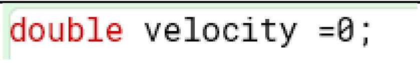


```
public float velocity(){
    velocity = velocity + acceleration() * (time/60);
}
```

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.

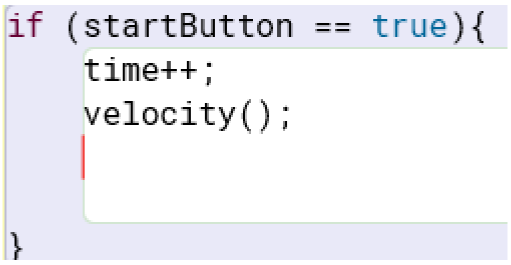


```
public double velocity(){
    velocity = velocity + acceleration() * (time/60);
    return velocity;
}
```



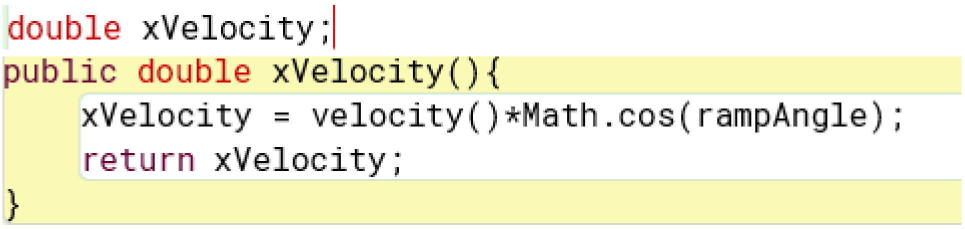
```
double velocity =0;
```

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.



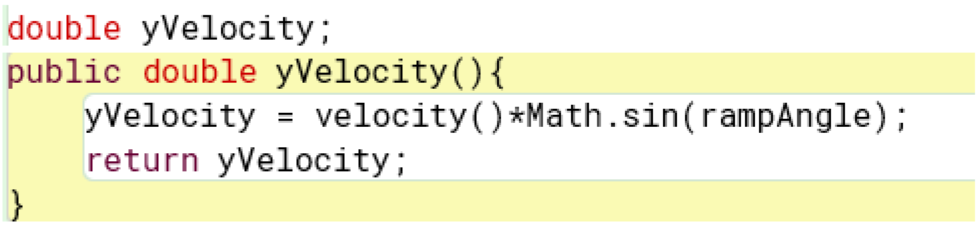
```
if (startButton == true){
    time++;
    velocity();


}
```

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.



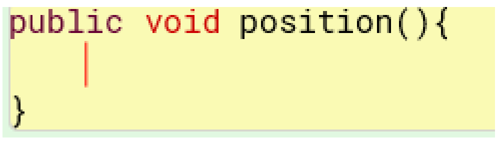
```
double xVelocity;
public double xVelocity(){
    xVelocity = velocity()*Math.cos(rampAngle);
    return xVelocity;
}
```

When trying to figure out how to get the ball to actually move on screen I realised that I need to get 2 separate numbers that one of the velocity in the X direction and one in the Y direction. So to do this I needed to make a new variable called xVelocity and then another method where I got the original velocity and then used cos to then get it in the x coordinate this is good as before I could not move the ball correctly but now with when I need to make it move I can by getting it to move across and then down. I used double as I will need a return as I will need to use the xVelocity in another method when changing the location of the ball.



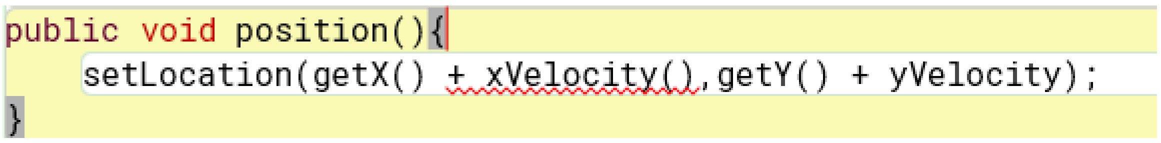
```
double yVelocity;
public double yVelocity(){
    yVelocity = velocity()*Math.sin(rampAngle);
    return yVelocity;
}
```

This method was made for the same reason of the last one as I need to get the yVelocity so that I know how much the ball needs to move down by so to do this I need to get the original velocity then use trigonometry to get the y side so I need to use sin to get there then I return it as a double so that I can use it to move the ball in a future method.



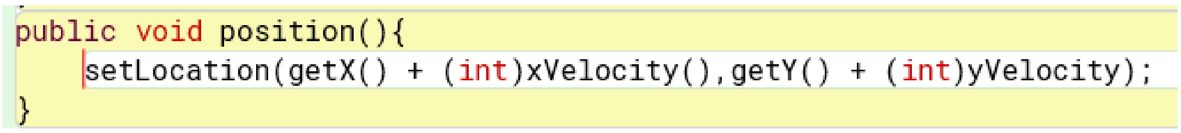
```
public void position(){
    
}
```

I have created this method because I will need a way to get the 2 different velocities and then actually use them to move the ball. The reason that it is a void method is because it doesn't need to return anything as it only needs to run the code within it and then stop.



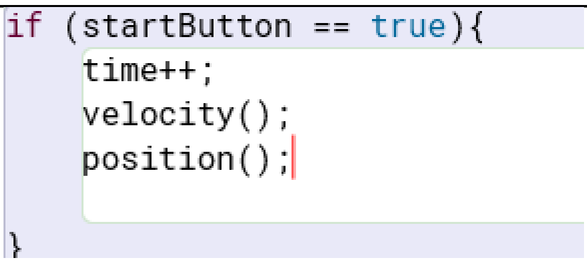
```
public void position(){
    setLocation(getX() + xVelocity(),getY() + yVelocity);
}
```

This code would allow it to move the balls position it does it by setting a new location by finding the location of the ball at this moment in time using getX() and getY() then i call the separate velocity methods and add them onto there respective ones. The reason I done this is because doing this instead of just getting it to roll down by making it not touch the ramp is because this allows me to change the speed of how much its going down as with the other way it would stay at one speed.



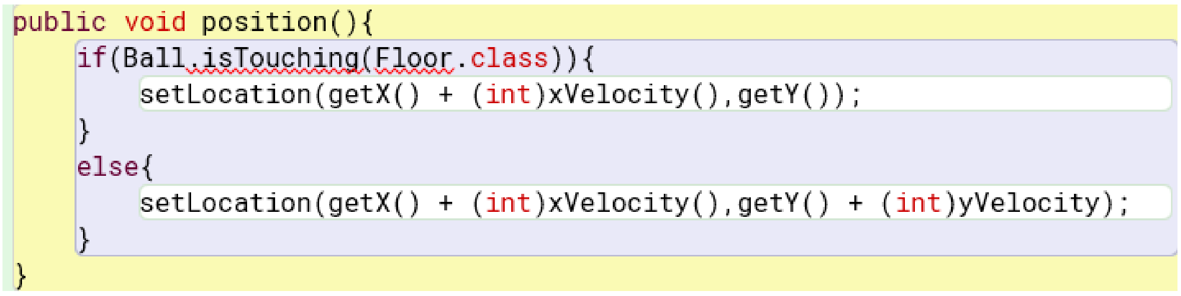
```
public void position(){
    setLocation(getX() + (int)xVelocity(),getY() + (int)yVelocity);
}
```

I then realised that it was not going to work as both of the velocity methods were doubles and in greenfoot it does not allow you to add a double and a integer together and the getX() and getY() have to be integers as they are exact positions in the world. So to fix this I have to put (int) in front of the velocities as what this does is changes the double into an integer which will then allow me to add them together then allowing me to change there positions. This way is good as it can move the ball at an increasing speed as the velocity is always increasing going down the ramp.



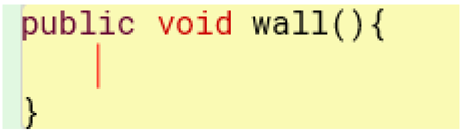
```
if (startButton == true){
    time++;
    velocity();
    position();

}
```

I then added it to my act method and put it in the if statement this is because the ball is only allowed to start moving once the start button has been clicked.



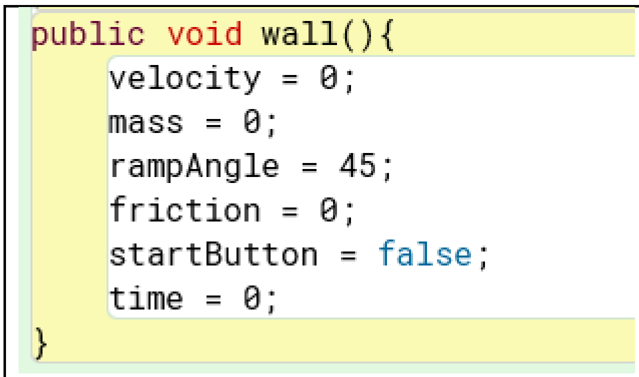
```
public void position(){
    if(Ball.isTouching(Floor.class)){
        setLocation(getX() + (int)xVelocity(),getY());
    }
    else{
        setLocation(getX() + (int)xVelocity(),getY() + (int)yVelocity);
    }
}
```

I then changed the method and added a if statement into it this is because if there wasn't one and it done the setLocation adding y and x the whole time once it reached the floor of the simulator it would just go through it and carry on forever but with this it makes it so when the ball finally gets to the bottom of the ramp and touches the floor it will change it so that the y coordinate doesn't change anymore and only the x does this is good as it means the ball will now go along the floor once it reaches it and it makes it look more like a realistic simulator. The if statement will not work for now this is because I have not made a floor class yet.



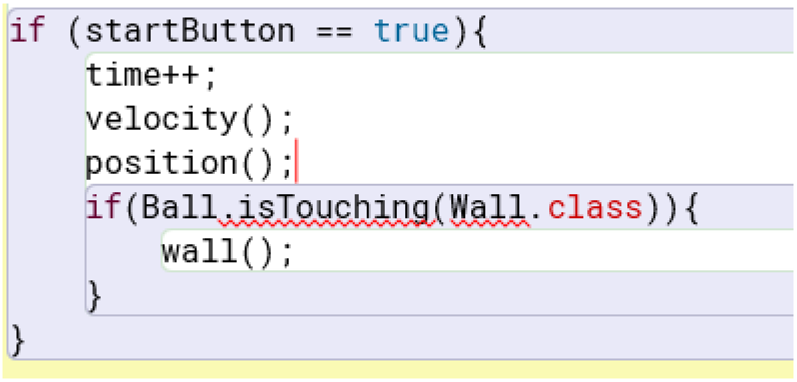
```
public void wall(){
    
}
```

This method is created to finally see if the ball has reached the wall as when this happens my simulator needs to reset and go back to how it was at the beginning this is because when the ball has reached the bottom of the simulator it is not doing anything any more so there needs to be a way for the ball to go back to the top so that is why there is a wall so that it can detect it and then reset. It is a void statement because all of the code is happening inside the method and nothing needs to be returned.



```
public void wall(){
    velocity = 0;
    mass = 0;
    rampAngle = 45;
    friction = 0;
    startButton = false;
    time = 0;
}
```

This is the full code of the wall method it has all of the different variables being set back to what they started at and also it sets the startButton back to false so that it can be ran again. I still need to add the reset position in because I have not yet set up the starting position for everything so I cannot add that until I have got everything set up in the correct position.

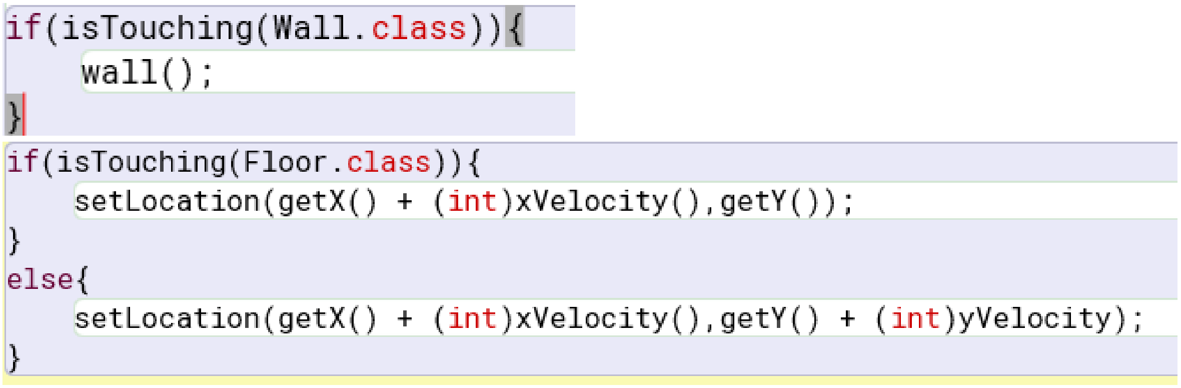


```
if (startButton == true){
    time++;
    velocity();
    position();
    if(Ball.isTouching(Wall.class)){
        wall();
    }
}
```

I have added the wall method into my act method and I have added a method which what its doing is that its getting the ball and then checking to see if the ball has touched the wall class and once that is true it will the use the wall class and reset everything. This is done because the simulator needs to be able to reset so must have a way to do it and it cannot be reset all the time so needs to have a specific way for it to be triggered. The reason why it is currently red is because the wall class has not been created yet so it will say it is wrong.

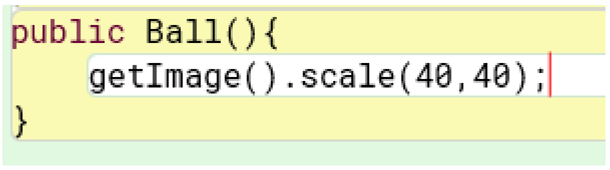
https://www.shutterstock.com/image-illustration/crystal-image-3d-rendering-600nw-2264170947.jpg

I added my image for my ball this is just an image from google that I found that is the correct colour I wanted it to be.



```
if(isTouching(Wall.class)){
    wall();
}
if(isTouching(Floor.class)){
    setLocation(getX() + (int)xVelocity(),getY());
}
else{
    setLocation(getX() + (int)xVelocity(),getY() + (int)yVelocity);
}
```

I added the 2 classes Wall and Floor into the simulator to see if the code was working correctly and I went back onto the code in ball and I saw that my code still had an error within it so I went back through my old notes to see what I had done wrong and I found that I don't need to put the Ball. Before isTouching as we are already in the ball class. I then fixed this and there are now no errors within my code.



```
public Ball(){
    getImage().scale(40,40);
}
```

I created a constructor for the ball class this is because I want to make sure that the image of the ball is the correct size and will not fill up the whole screen so to do this I used getImage and scaled it to a normal size however I am able to come back to this and change it when I am connecting all of the classes together.

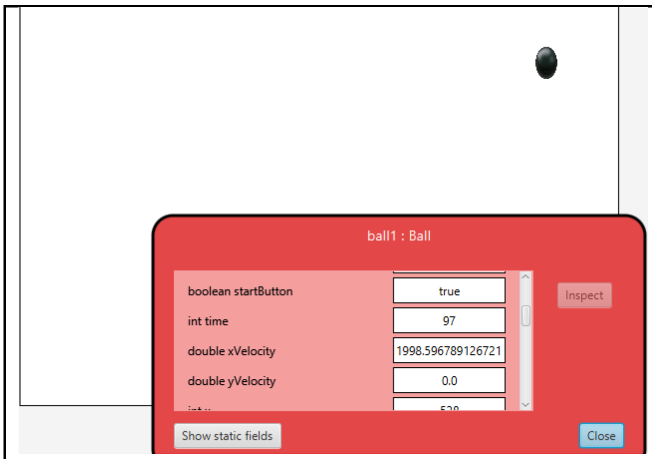
Test Using the Test Plan in that section and mark them as working when they are.

Fix any Problems.

Show screenshots of the section fully working.

<u>TEST 1</u>

Spawn in the ball at the top of the screen and see if it falls. This is to test the gravity on the ball.


ball1 : Ball

```
boolean startButton    true
int time               97
double xVelocity       1998.596789126721
double yVelocity       0.0
```

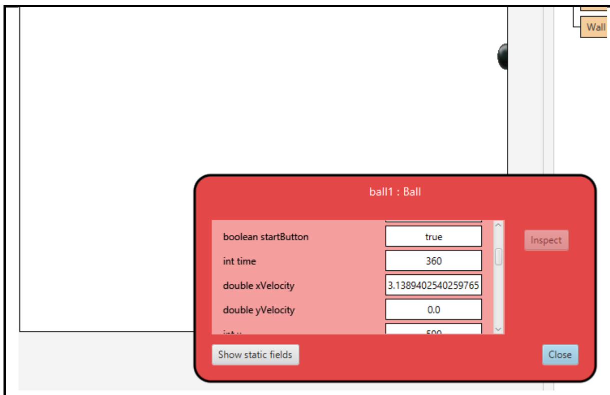
This test was ran using the mass of 6 ramp angle of 45 friction 0. This test has partly worked as the ball moved right instead of down however this is correct as in my code if there isn't a ramp there the ball will move to the side instead however the problem is that after 97 seconds the velocity on the x axis is going at a speed of 1998 and even after 97 seconds it should not be going that fast.

```
public double acceleration(){
    double acceleration;
    double force = mass*9.8*Math.sin(rampAngle) - friction;
    acceleration = force / mass;
    return acceleration;
}
```

To fix this problem I asked a friend in maths and looked in my maths book and I realised that I had forgotten to divide the acceleration by mass this needs to be done because I am using the equation F= ma and before I was just getting the force in the equation then I wasn't dividing the mass so it was giving me too big of a number which then caused it to go off the screen.

```
public double velocity(){
    velocity = velocity + acceleration();
    return velocity;
}
```

Another Problem I fixed with this is that I don't need to multiply the acceleration by the time this is because where I am adding the acceleration on every time it is just doing the same thing so there is no reason for it to be adding on time every time.


ball1 : Ball

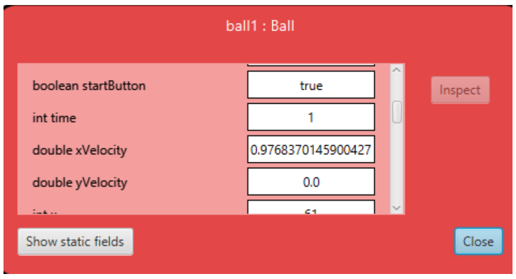
boolean startButton	true
int time	360
double xVelocity	3.1389402540259765
double yVelocity	0.0

This was the end result of them 2 fixes which now the xVelocity is going at the correct speed and it slowly takes time to increase the speed instead of very quickly as it was before. This means that Test 1, Spawn in the ball at the top of the screen and see if it falls. This is to test the gravity on the ball, is correct as there is forces pushing on the ball shown in how the xVelocity is increasing due to them.

<u>TEST 2</u>

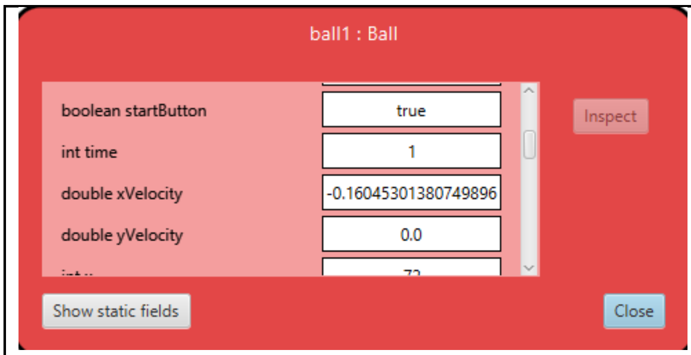
Set Different Data onto the ball. This is to test to see if the inputs are working correctly.

The first set of data I entered is Velocity at 2, Mass at 8, rampAngle at 20, friction at 70


ball1 : Ball

boolean startButton	true
int time	1
double xVelocity	0.9768370145900427
double yVelocity	0.0

This in the first second then gave out an xVelocity of 0.4667. I looked at this using the inspect button in greenfoot which allows you to look at all the different variables that are connected to an object. I then used a calculator using all of the same data variables and the answer to it was correct this is good as it means my calculations in my code are all correct to check this I will use new variables. The second set of data I entered is Velocity at 2, Mass at 11, rampAngle at 40, friction at 60



ball1 : Ball

boolean startButton	true
int time	1
double xVelocity	-0.16045301380749896
double yVelocity	0.0

I then put all of the same data into the my calculator again to see if it's the same and when I worked it out it gave the same xVelocity as it does in the code. This all then means that my calculations are fully working as they should meaning that my test 2, Set Different Data onto the ball. This is to test to see if the inputs are working correctly, is correct. Finally meaning that my Ball class I working how I would like it to work.

Write up how the section went.

Overall I think that this section of my development went well this is because I was able to fix all of the problems that showed up within my code and then by the end of it all of the test were running correctly. As this is a main part of my simulator it is good that I have gotten my code to work in a way that should work with the other sections once I create them. I was able to go through each stage of the flow diagram doing every part in the order it is on there and I finished all of the sections on the flowchart. The problem that I kept making while writing the code is that when I needed to make the method one with returns within it so that it can then be used in another method getting the return from that one so then for the future sections I need to understand more on what type of method each needs to be before hand.

Ramp
Flowcharts

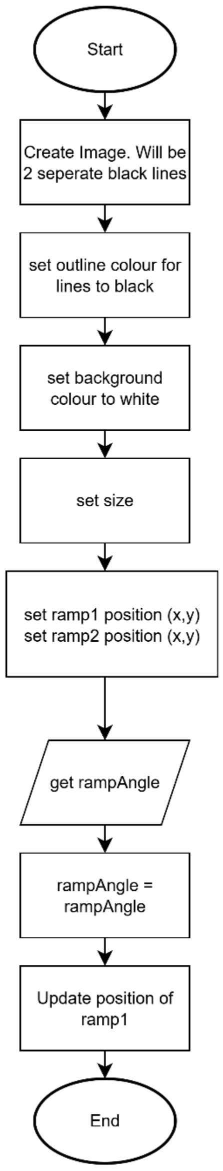


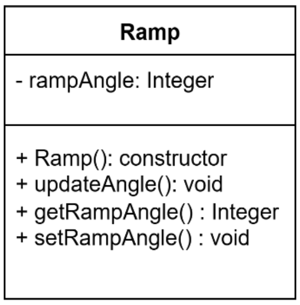
Diagram of each section of code in the stage showing how the code should Run.

Data Dictionary

Name of Data	Type of Data	The Use of the Data.
rampAngle	Integer	It is used to change the image of the ramp so that the angle used to effect the ball is the same as what the ramp image actually looks like this

		is good as it makes it look more realistic.

Class Diagram



Ramp

- rampAngle: Integer

+ Ramp(): constructor
+ updateAngle(): void
+ getRampAngle() : Integer
+ setRampAngle() : void

Write part of the code.

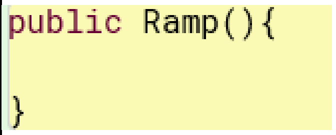
Screenshot it and say what its done and justify it.

Run tests you made.

Fix Problems.

Repeat.

I have created a new class for ramp this class is will have a ramp image for to put on the screen and it will be able to change the angle of the ramp by using the input to change the direction the ramp is facing.



```
public Ramp(){

}
```

I have created a constructor for the ramp class this is because in this class I would like to actually make the image instead of just looking one up the image will be just a black line this is because then I can use these images to then make an outline of a ramp as I will need to images to be separate so I can make it so only one of the images changes when I change the angle.

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.