Sarge's Siege: An Intuitive Design Interface

Connor McDowall

5309133986

Group 10

Department of Engineering Science

25th September 2017

Executive Summary

We were given the task to use a spreadsheet to create a tool for a user to explore the design space of a mechanical device. We choose a mangonel catapult. Live action role players and physics departments can order mangonels easily though an the user interface.

We planned the project at the beginning, setting goals and assigning tasks to team member, compartmentalising the project. The mathematics of a mangonel were broken up into two parts; the launching mechanics and projectile motion. We applied the mathematics to the design parameters of the mangonel. We decided on five design parameters; size, material, angle, wheel size and skin. These were quantised into three to five options per parameter.

We designed and implemented performance metrics based on the catapults design. These were; maximum height, distance, velocity, acceleration, impact force, force required, cost and mass. Dials, bar graphs and images that change relative to the maximum value were used to convey comparative information.

The invoicing, clearing and navigation features of the workbook are described.

Detailed instructions are given on how to navigate the workbook. The contents of each sheet in the workbook are explained with descriptions of the user forms and buttons.

An analysis of a medium sized mahogany mangonel with a 45˚ angle is given. The performance metrics are discussed, adding what impact a change in the input parameters would have on the performance.

Concluding the report, comments regarding my contribution to the group and a reflective discussion are described. The appendices with the layout of our design tool follow soon after.

Table of Contents

Table of Figures

Introduction

Spreadsheets have the functionality to be used as a creative tool. Our task was to use a spreadsheet to create a tool to explore the design space of mechanical device, referring to tables of data and specific qualities of our customisable parameters. We were to use Excel macros and forms controls to customise our worksheet. The results generated from the various design choices were to be displayed in a useful form e.g. report summaries, graphs or charts. A table of key information must be retrospectively generated, from our application, for use in Solidworks. There was to be choice for the user to explore the design space. The device was to be mechanical in nature.

We choose a catapult as our mechanical device. After discussing the mangonel, Ballista and Trebuchet variants, we decided to customise a Mangonel catapult in our design space. Our target users are live action role players who wish to add realism to their games and physics departments teaching projectile motion.

Our user interface (UI) is fluid and intuitive. Ordering a catapult in this age is difficult due to the historical nature of our product. Our UI guides the user from start to end. The user has control over customising five parameters of the Mangonel’s design. The UI also has the ability to quantifiable measure and compare different designs and order multiple mangonels. Our product enables the user to design a series our Mangonels which caters to their needs.

Our mangonel is a mechanical device. Our interface guides the user start to end linearly. We refer to parameter specific variables to determine design specific metrics. Forms Controls and Macro were written to customise the mangonel, measure the performance and navigate the UI. Mangalore specific performance was demonstrated through an animation, dials and bar graphs. Therefore, our product meets the design brief.

Development

Initially, I facilitated the planning stage of the project. Each team member outlined their goals for the project while identifying their strengths and weaknesses (Appendix 1). Tasks were assigned based on goals, strengths and weaknesses to improve motivation and team synergy. I outlined an estimated project schedule. I took the project objective and specifications, compartmentalising the project. I assigned a task portfolio to each team member. I oversaw the completion of the project from start to end. We compartmentalised the project into five portfolios; Customisation, Physics with Calculations and Animation, Performance and Comparative Metrics, Design and Invoicing. In addition to these portfolios, I worked on UI navigation and integration. I conceptualised each sheet at the beginning of the project with the team agreeing on a design (Appendix 18: Concept Sketches).

After planning, we investigated the mathematics of a mangonel to find a way to measure the mangonels performance objectively. The Mangalore stores energy in a torsion bundle. As the torsion bundle is wound, restrained from recoiling. This bundle is approximated as a spring. During this coiling, the arm is bent downwards. When released, the arm rotates around a pivot point at an angular acceleration with a rotational inertia. A cross beam is struck, converting halting the arm and releasing the loose projectile. Thus, we model projectile motion, assuming the projectile follows a two-dimensional path, and, subject to drag and the force due to gravity. Both drag and the force due to gravity depend on the catapults size.

The calculations were done in two stages. First, we calculated the combined rotational inertia of the projectile and catapult arm. The torsion bundle is assumed to be modelled as a rotating spring, then we calculate how much the arm will rotate due to the force in the spring.

This is done by summing the moments around the fixed point the arm is rotating around, and then using $\tau = I\alpha$ to calculate the angular acceleration where I is the rotational inertia. Heavier materials have a greater rotational inertia, therefore, a lower angular acceleration, thus, the projectile will have a lower launch velocity.

We then calculate how both the force of a spring ($\tau = k\omega$) changes and velocity of the arm changes as time progresses. We used a method similar to Euler's to calculate how the angle of the arm changes with time, using a time step. The angular velocity and length of the arm are multiplied together to determine the launch velocity of the projectile when the arm hits the cross beam.

All of these parameters, such as spring constant and length of the catapult arm depend on the input parameters. Once we had the catapult parameters (Figure 1), the angle of release and launch velocity, we have a projectile motion problem. We used another Euler method to calculate the projectile's position in the air. The forces of gravity and air resistance were taken into account with a friction co-efficient of 0.47 and gravity at $9.81m/s^2$. We allowed the projectile to bounce if the height of the projectile was less than the ground, in which case a co-efficient of restitution of 0.4 was applied to new vertical and horizontal velocities, and used the same ratio to put it back above the ground.

We chose to derive a numerical solution using Euler's method for the projectile motion. We deemed an analytical solution and improved Euler's method to be unnecessary as these methods were only slightly more accurate, thus, did not justify the greater computation strain. We decided to use spreadsheet manipulation to perform our calculations instead of function calls due to the spreadsheets greater efficiency and the production of smoother animations (Appendix 2: Calculations).

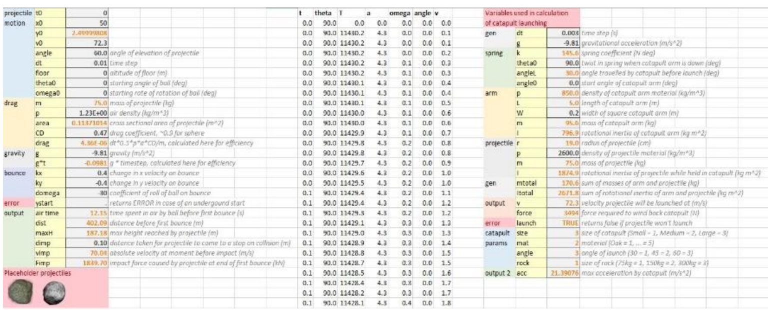


Figure 1: Spreadsheet Calculations

Once we determined an accurate method to influence projectile motion, we began integrating the maths into an UI. We decided to quantise our input parameters in a way to simplify the selection process. We decided to split our mangonel into five customisable parameters; our functional parameters size, material and angle of release, and, our cosmetic parameters wheel size and skin. Size was split three groups; small, medium and large. Material was split into five group; oak, birch, mahogany, willow and ash. The angle was split into three groups; 30˚, 45˚, 60˚. The size of the catapult controls the force stored in the torsion bundle. The material controls the inertia of the rotating arm. The size of the angle changes the position of the cross bar on the, mangonel, therefore, the angular distance, angular velocity and launch velocity of the projectile. The skin was split into five; wood, flames, camo, tiger stripes and jim. Wheel size was split into three groups; small, medium and large. We used the VLookup and HLookup excel functions to access the "Control Variables" and "Database" Sheets parameter properties (Appendix 12 and 13 respectively), to display changes and link the designed catapult to the "Performance", "CalcSheet", "ResultsPage", "Intermediary Invoice" and "Animation" pages (Appendices 7,14,15,8,6). We decided to use lists that always have the customisable parameters selected (Figure 2) instead of dropdowns to improve the user friendliness and aesthetic appeal of the UI (Appendix 5: Design Page).

Figure 2: Parameter Selection with Associated Design

After solidifying our parameter selection, we used the "Database" sheet to allocate unit costs and unit weights relative to the size of the catapult selected. We have not dimensioned our parameters; however, we have proportionality constants incorporated into IF statements (Figure 3) to change the catapult launching parameters in the "CalcSheet" and "ResultsPage" (Appendix 14: Calcsheet and Appendix 15: ResultsPage). We used a series of HLookup, VLookup and INDEX Functions to manipulate costs and weights, placing them in the reference table in the "ReferenceData" worksheet (Appendix 11: Reference Data). We use a series of subroutines to determine the unit costs and total cost of the design. We used GetCatapultFilename and SetCatapultImagesubroutines (Appendix 17: Sub Routines and Functions) to access the image associated the combination for parameters (Figure 4) which are located within the same directory as the workbook. The updated inputs are feed through all relevant pages to keep the workbook consistent with the current design.

spring	k	=112*IF(AG23=1; 0.75; IF(AG23=2; 1; IF(AG23=3; 1.3;"ERROR")))
	theta0	90
	angleL	=IF(AG25=1; 60; IF(AG25=2; 45; IF(AG25=3; 30;"ERROR")))
	angle0	0
arm	p	=1000*IF(AG24=1; 1; IF(AG24=2; 0.85; IF(AG24=3; 1.2;IF(AG24=4; 0.8; IF(AG24=5; 0.55;"ERROR")))))
	L	=IF(AG23=1; 2; IF(AG23=2; 3.5; IF(AG23=3; 5;"ERROR")))
	W	0.15
	m	=$AG$11^2*$AG$10*$AG$9
	I	=$AG$9*$AG$10^3*$AG$11^2/3
projectile	r	=IF($AG$26=1; 19.025; IF($AG$26=2; 23.9705; IF($AG$26=3; 30.201;"ERROR")))

Figure 3: Parameter values based on Selection

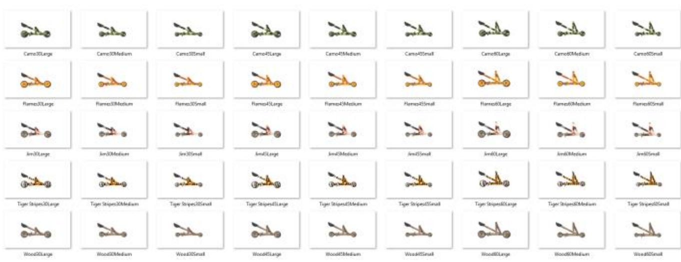


Figure 4: Catapult Image Database

After implementing how to assign and change parabolic performance metric based on design, we began producing methods to display performance. From planning, we agreed on two ways. The first an animation allowing the tracking the projectile's trajectory in real time. Using the calculations set by the parameters, users may select 1 of 3 projectile sizes. The animation uses the mathematics of projectile motion to model the trajectory. The animation scene is drawn each time using the DrawScene() subroutine. The projectile line is drawn and updated with a max height label added along the way. The DrawAnimation() sub routine draws the animation of the projectile moving. The projectile changes position with a DrawFrame() sub routine called iteratively (Appendix 17: Sub Routines and Functions). A keep_ prefix was used on images to prevent their deletion, as the animation resets each time. We decided to use an animation as gives the user an immersive experience, following the vision of an intuitive UI (Appendix 6: Animation Page).

We decided to dedicate a page to displaying key performance metrics based on the user's design (Appendix 7: Performance Page). We decided dials, images and filler bars were the best way to convey the performance data, as all users have a basic understanding of these displays. We choose air time, maximum velocity, impact force, maximum acceleration, maximum height and distance travelled as metrics. The maximum values for all metrics were derived from the parameter variations. The range of the metric (zero to maximum value) is converted into a co-ordinate system for the dials. Based on the metric values derived from the "CalcSheet" and "ResultsPage", a slither of a pie graph is shown overlying a dial display and labelled co-ordinate system (Figure 5). The SERIES function is used in the bar graphs, filled to the actual values derived in the "ResultsPage" between the minimum and maximum values for that metric (Figure 5).

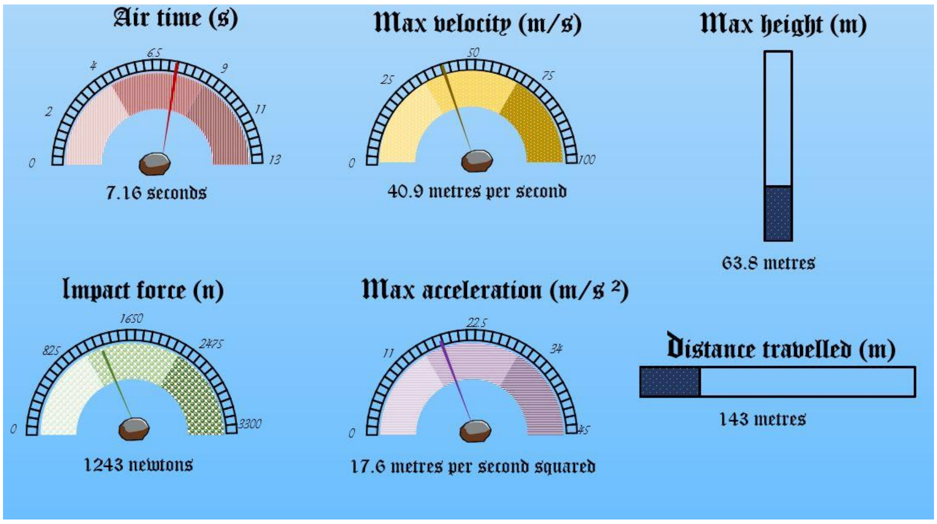


Figure 5: Performance Dial and Bars

We decided to display images relating to the cost, mass of the catapult and force needed to operate the catapult to simplify design comparisons. The subroutine Visible_Images() shows and hides images relative to the design catapults total cost, mass and force required to operate the catapult, using the Excel shape feature .visible = Boolean. We tried to make the display as simple as possible to not confuse the user.

The user reviews the design catapult in an individual invoice (Appendix 8: Intermediary Invoice) where the metric descriptions and user costs are displayed. The user can choose to save the catapult to the final invoice, saving up to a maximum of four catapults per deign, with up to four different designs, for a total of sixteen. We choose sixteen as deemed any more to be unrealistic for our target users. Designs can be reset by a clearing subroutine based on a conditional value in reference table. Discount and Delivery options are incorporated. GST is accounted for with the sub totals and final order totals displayed (Appendix 9: Final Invoice). You can reset your sheet and begin again by using the clear all function, resetting all reference values in the "ReferenceData" sheet (Appendix 11: Reference Data).

The navigator() function allows you to navigate the workbook seamlessly, allowing the user to revist pages previously visited but not pages yet visited. UpdateReference() changes the values in the reference table in "ReferenceData", enabling this functionality. The navigator was used to create a linear a linear and fluid experience for the user.

Submitting your order will bring you to the summary page (Appendix 10: Summary) where all catapults designed will be displayed using the Pedestal() function, showing images relative to the designs.

There are 14 user forms to aid the user. They pop up based on; your stage in the UI, Boolean variables, reset values, select options and error navigation (Appendix 16: User forms).

Instructions

Upon opening the workbook, you are presented with the home page. Click begin to start. You are taken to the design page. Select the parameters you wish to incorporate into the catapult. Your design will be displayed below parameter selection. If you wish to navigate back to the home screen, click the home button where a user form will pop up and take you home. If you don't know what to do, select the help button and the help pop up will show you. A display cost will instruct you on the individual cost of the parameter selected, and, the current design's cost and weight. Click the simulate button to move forward to the animation.

Drag and drop the castle to the desired target distance between 250m to 500m. Select one of the three projectile options in the top left-hand box and click animate. You will see the projectile path generated in front of you. If the projectile strikes the target, you will see an animation. You can select home to navigate home, help for advice on what the page does or results to move forward to see the performance metrics.

You will see four dials, three image sets and two bars indicating the performance of your design. You can adjust the projectile size in the drop box to change your catapults performance relative to the projectile. The metrics are; air time (s), maximum distance (m), height (m), velocity (m/s) and acceleration (m/s^2), impact force (N), force required (N), mass (kg) and Cost (GP). The images on the right approximate the mass, cost and force required to operate the catapult relative to the designs which max out these metrics. When you are ready to see the intermediary invoice, select the Invoice button.

The intermediary invoice gives you a summary of your individual catapult order. You will see a display of the parameters you selected, their associated unit cost and the sub total of the catapult before delivery, labour, GST and discount costs.

If content with your order, click invoice. Otherwise, use the navigator to go back to previously visited pages by selecting the button relating to the page you want to visit under the grey arrow. If the arrow above the button in red, you have not visited that page yet, therefore, you cannot move to that page. You can select the home button to navigate home or help if need a little guidance.

If you selected invoice, you will be presented with a variety of user prompts. First, select the quantity you wish to order and your design will be saved. After, you will be given the option to design another catapult. If click yes, you will be taken back to the design page to design another. If you click no, you will be given the option to choose delivery and payment option. After selecting both, you will be taken to the final invoice page.

The final invoice page displays your entire order. You will see; all parameters, the cost per unit, the delivery option, the payment method, labour costs, discount amount, the delivery cost, the GST charge, a subtotal and the total order cost. You can navigate back to home if you choose to do so or ask for help by clicking the help button. You can clear previous catapult designs by pressing the clear button. You can clear individual catapults by selecting the save slot to clear or clear them all by clicking the clear all button. If you try to clear an empty slot, you will be told you cannot and must select another. If you clear all or one catapult, and have no catapults saved, a prompt will come up asking if you want to start your order. Click Start Order to reset all data and you will be taken to the design page. Click Stop Order to reset all data and be taken back to the home page. If you have catapults saved and are happy with your order, select submit. A form will pop up. Click yes to confirm your order and move forward to the summary page where your designed catapults will be displayed on pedestals. Click no and you will be taken back to the final invoice page to reassess and adjust your order.

Analysis of Results and Outputs

For demonstration purposes, we will choose a medium sized catapult made of a mahogany, with a firing angle of 45˚. The wheel size is medium and skin camo do not affect the performance of the catapult.

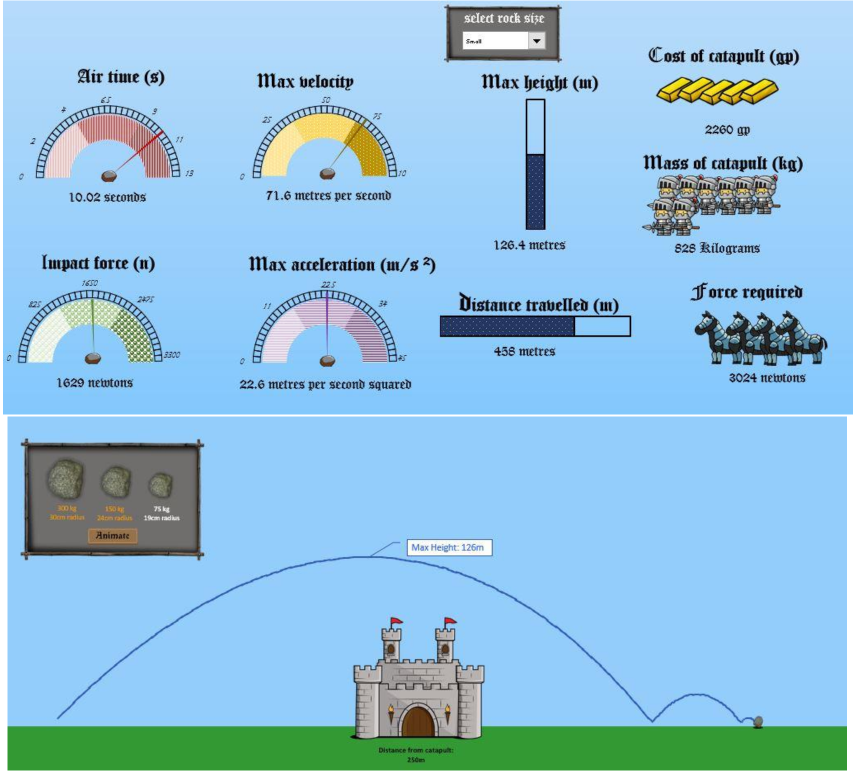


Figure 6: Animation and Performance Output

Here we see multiple performance metrics relating to the flight path of a small projectile fired by the user’s catapult. Firstly, we will discuss the dials. As you can see, air time was 10.02 seconds. The smaller angle would decrease the flight time and the max height reached. The maximum velocity is moderately high. The designed catapult is of a size and material light enough to generate a high launch and max velocity. Changes to the size and material of the catapult will change the metric. Impact force and maximum acceleration are proportional due to the relationship F = ma. Changing the size of the projectile, material and catapult size will adjust these metrics. Due to the angle and lightness of the catapult, the trajectory shows the projectile travels a good distance. This will change based on the angle, projectile size, catapult size and material. The maximum height on the animation screen matches the maximum height displayed on the bar in the performance page. On the right above, you will see the cost of this catapult is about half the design of the maximum cost, the mass is a little more than half the heaviest catapult possible and required slightly under half the force require as that of the design which requires the most. The size of the catapult, material, launch angle and the size of the projectile all contribute to the performance of the catapults. Different combinations of parameters will produce different results; therefore, a user will need to tailor an order to their specific needs. These metrics may be used to compare catapults and order a catapult which directly addresses the needs of the consumer. A customer could tailor their order to get catapults that shoot high and close by with a large angle and heavier materials and a catapult that shoots far and flat with a small angle and light materials. Our performance metrics allow the user to diversify their fleet of catapults for sieging.

Group Contribution

I was the group leader. I executed all the project planning at the start of the project, organising the project goals and timeline. I deconstructed tasks and assigned them to the team members. I oversaw the project, checking in on other team members. I tracked progress. I conceptually designed the pages. I planned and ran team meetings. I created the Intermediary Invoice, Final Invoice and ReferenceData sheets for the workbook. I designed and coded 11 of the user forms. I wrote the subroutines and functions in the following modules; Buttons, Calculations, Clearing, Formatting, Navigate, Performance and Saving. I designed the logo and navigator bar. I integrated the sheets into the workbook, creating a functioning user interface. I have personally spent 100+ hours on the project.

Reflective Discussion

Our group worked well together. We had little conflict. All members completed assigned tasks. We settled disagreements constructively. I am happy to work with each member again. Our initial project planning worked well. Outlining the goals, objectives and intentions of each team member. I was able to allocate tasks team members were motivated to complete. All team members completed assigned tasks and enjoyed doing so as suited their skillset and interests. The compartmentalisation of the project also worked well. Everybody had their own portfolio of work to be completed independently. There were no delays waiting for other team members to complete their portion of the project. Work was completed seamlessly. Deconstructing the project into small tasks was really useful. We had a clear direction with little disruption. One leader for everybody to report to worked well. Facebook messenger and face to face meetings were excellent communication tools. I was able to distribute new information very quickly. Transparent communication channels were crucial to product development.

There are a few things I would like to improve upon if given the opportunity. Integrating the final product was difficult as other team member either hard coded large proportions of their code, or, did not write down there thought process or methodology. A shared team workbook would be beneficial for sharing design decisions and cutting down integration time. We need to spend more time during the iterative design process at the start of the project. We jumped into the product very quickly with only a few concepts to go off. Seeing the features other groups incorporated, it would have been good exploring more ideas before deciding what to do. We will implement a more complete and comprehensive iterative design process in the future. I will need to get better at delegating work. As project lead, I took a lot on. I managed but tasks I did would have been best suited to the skillsets of other team members. We didn't fully explore the features of excel for example active x controls. We may have found features that would enhance our UI. We will spend more time researching tools and features in the future. To conclude, this project has been vastly enjoyable and a huge learning experience in design, coding, teamwork and project management.

Appendices

Appendix 1: Project Planning, Goals

Members	Connor	Matthew	Olivier	Uhalan	Frankie
Goals	• A+ in the Project and Course. • Build teamwork skills • Develop VBA Skills	• Learn how to produce useful and good-looking modelling tools. • Improve in ability to formulate models from real world constraints	• End product proud of • Good Grade	• A in 263 • Develop VBA Skills	• To produce a final result that I am proud of • To work well as a team • Ideally get an A or A+ grade in project/course
Strengths	• Project Management • Grit • Design Skills (Photoshop)	• Programming/simulation of projectiles • Fairly comfortable with photoshop/graphic design	• Programming • Maths • Physics	• Willingness to Learn • Adaptable • Flexible	• Capable User of Excel • Experience with photoshop • Design understanding • Good understanding of physics and equations
Weaknesses	• Working close to deadlines.	• Image Animation	• Design (Photoshop)	• Don't have significant coding strength	• Not quick at coding • Little VBA experience • Public speaking
Most Enjoyed	• Learning new skills • Increase competency with coding	• Satisfaction of making a well working & fully fleshed out tool • Seeing how intricate and accurate we can make our simulation	• Slow assembly of the product • Go deep into things if we want to	• Producing a finished project.	• Improving VBA skills • Producing atheistically pleasing product with good interface. • Linking background calculations with outputs from users
Least Enjoyed	• Rushing Project • Working close to deadlines	• Trying to consolidate programming from multiple different people with different styles		• How Finicky coding can be sometimes, but again, don't mind it too much.	• Research catapult info/entering in parameters and data. • Presentation

Project Timeline						
Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday
16th	17th	18th	19th	20th	21st	22nd
Create Project Deadline	Ideate	Ideate	Finish Research/ Finalise Performance Metrics	Ideate	Complete Template Calculations for all performance metrics	Module Formulation and Design Work
23rd	24th	25th	26th	27th	28th	29th
Module Formulation and Design Work	Module Formulation and Design Work	Module Formulation and Design Work	Module Formulation and Design Work	Module Formulation and Design Work	Module Formulation and Design Work	Module Formulation and Design Work
30th	31st	1st	2nd	3rd	4th	5th
Module Formulation and Design Work	Module Formulation and Design Work	Module Formulation and Design Work	Module Formulation and Design Work	Module Formulation and Design Work	Module Formulation and Design Work	Module Formulation and Design Work
6th	7th	8th	9th	10th	11th	12th
Module Formulation and Design Work	Module Formulation and Design Work	Module Formulation and Design Work	Module Formulation and Design Work	Module Formulation and Design Work	Module Formulation and Design Work	Module Formulation and Design Work
13th	14th	15th	16th	17th	18th	19th
Module Formulation and Design Work	All Design Images Complete		All Coding and Modules Complete	Final Check and Presentation Creation	Presentation Practise	Presentation Practise and Begin Report
20th	21st	22nd	23rd	24th	25th	26th
Presentation Due	Presentation Practise	Report	Report	Report	Report	Final Touches on Reports

Key Dates:

- Presentation 20th September
- Design Project A Report 27th September
- Design Project B Files/Spreadsheet 27th September

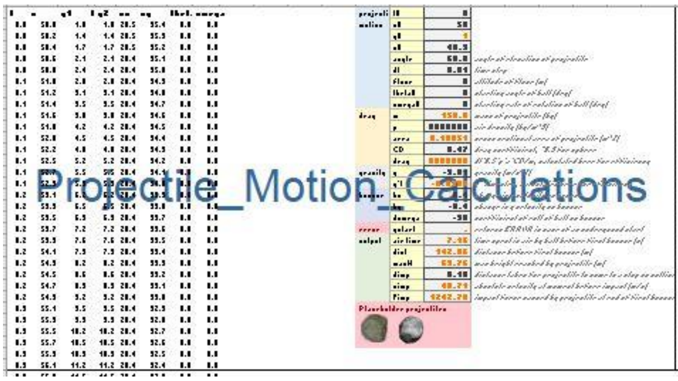
Key Tasks:

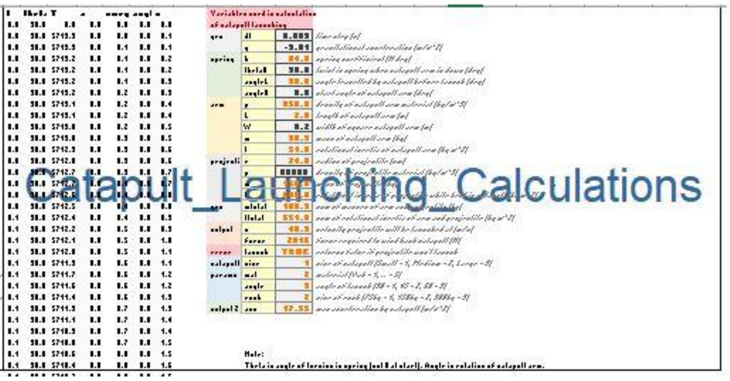
- Modules
  - Number
  - Specific Task of Modules
  - Subroutines versus Modules
- Design (Images, Aesthetics, Processes, Calculations)
  - Allocate one person to the task
  - Cover Page
  - Invoice Form
  - Graphs
  - Catapult Images
- Performance Metrics
  - Graphs, Animations, Relate to Physics
- Animations
  - Flight Path, Trajectory
- Dials, Controls, Performance Measurements
  - Change dependent on performance calculations
  - Previous Iterations comparison, Option to Save?
- Invoice Form
  - Design, Aesthetic

Roles

- Design (1)
  - Create images, graphs, displays for the graphs
- Physics (1)
  - Research and Calculate Performance based on inputs.
  - Investigate mathematics
    - Simple projectile motion, incorporating gravity and air friction co-efficients.
    - Calculations Based On Time Periods
- Model (3 to 5)
  - Organise the creation of modules.
  - Decide what needs to be created.
- Manager (1)
  - Manage Progression
  - Make decisions
  - Oversee entire project
- Performance (3 to 5)
  - Model and create performance metrics
    - Vertical and Horizontal; Distance, Speed and Acceleration?
    - Whether Target Hit

Appendix 2: Calculations





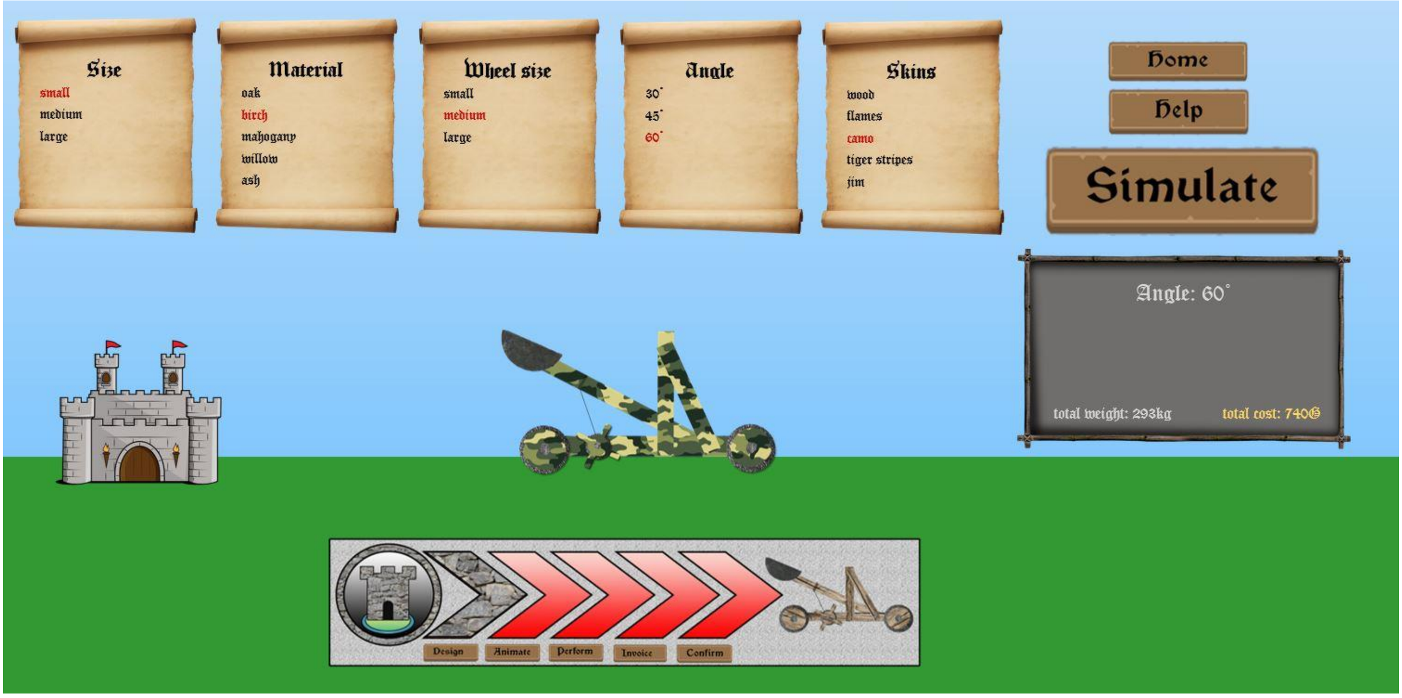
Appendix 3: Calculations with Formulas

Variables used in of catapult launch			
gen	dt	0.003	time step (s)
	g	-9.81	gravitational acceleration (m/s^2)
spring	k	=112*IF(AG23=1; 0.75; IF(AG23=2; 1; IF(AG23=3; 1.3;"ERROR")))	spring coefficient (N deg)
	theta0	90	twist in spring when catapult arm is down (deg)
	angleL	=IF(AG25=1; 60; IF(AG25=2; 45; IF(AG25=3; 30;"ERROR")))	angle travelled by catapult before launch (deg)
	angle0	0	start angle of catapult arm (deg)
arm	p	=1000*IF(AG24=1; 1; IF(AG24=2; 0.85; IF(AG24=3; 1.2;IF(AG24=4; 0.8; IF(AG24=5; 0.55;"ERROR")))))	density of catapult arm material (kg/m^3)
	L	=IF(AG23=1; 2; IF(AG23=2; 3.5; IF(AG23=3; 5;"ERROR")))	length of catapult arm (m)
	W	0.15	width of square catapult arm (m)
	m	=$AG$11^2*$AG$10*$AG$9	mass of catapult arm (kg)
	I	=$AG$9*$AG$10^3*$AG$11^2/3	rotational inertia of catapult arm (kg m^2)
projectile	r	=IF($AG$26=1; 19.025; IF($AG$26=2; 23.9705; IF($AG$26=3; 30.201;"ERROR")))	radius of projectile (cm)
	p	2600	density of projectile material (kg/m^3)
	m	=4/3*3.14159*($AG$14/100)^3*AG15	mass of projectile (kg)
	I	=$AG$16*$AG$10^2	rotational inertia of projectile while held in catapult (kg m^2)
gen	mtotal	=$AG$12+$AG$16	sum of masses of arm and projectile (kg)
	Itotal	=$AG$17+$AG$13	sum of rotational inertia of arm and projectile (kg m^2)
output	v	=INDEX($AC$2:$AC$5002;MATCH(TRUE; INDEX($AB$2:$AB$5002>$AG$7;0);))	velocity projectile will be launched at (m/s)
	force	=AG5*(AG6+AG7)/5	force required to wind back catapult (N)
error	launch	=NOT(ISNA(AG20))	returns false if projectile won't launch
catapult params	size	=SizeSelection	size of catapult (Small = 1, Medium = 2, Large = 3)
	mat	=MaterialSelection	material (Oak = 1, ... = 5)
	angle	=AngleSelection	angle of launch (30 = 1, 45 = 2, 60 = 3)
	rock	=MasterV5.xlsm!RockChanger	size of rock (75kg = 1, 150kg = 2, 300kg = 3)
output 2	acc	=Z3*AG10	max acceleration by catapult (m/s^2)

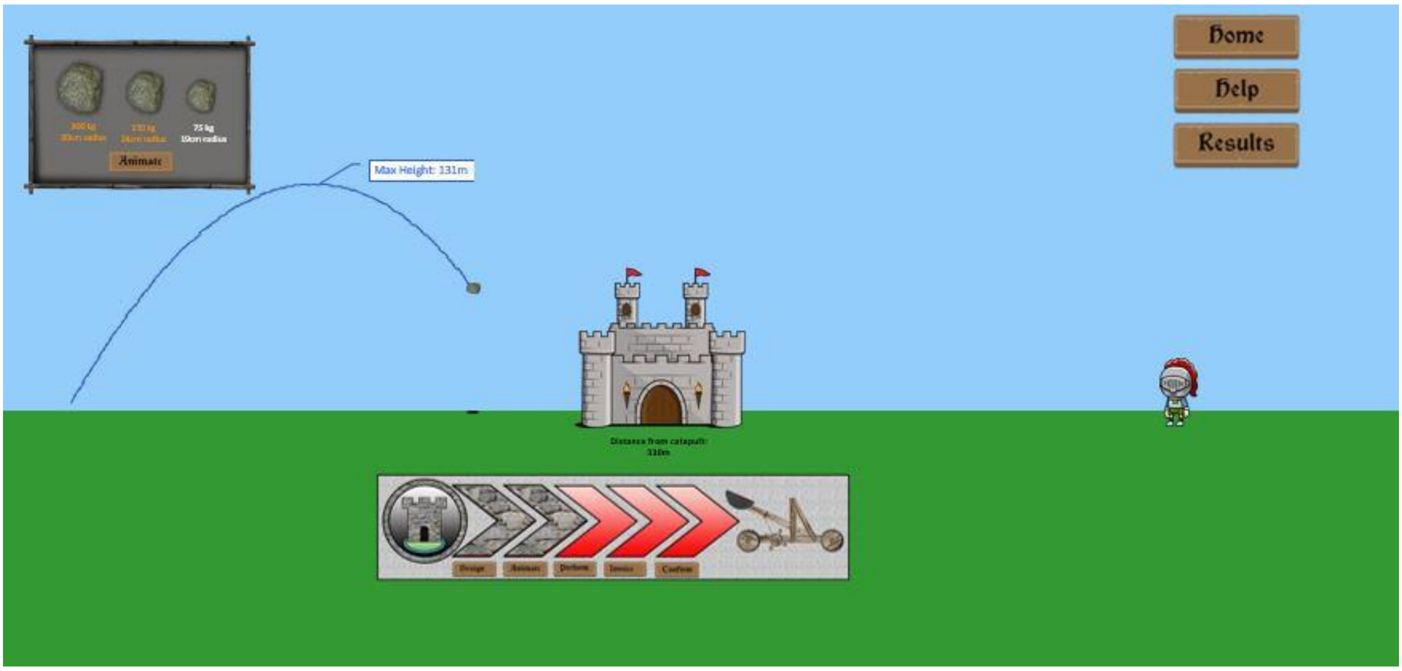
projectile motion	t0	0	
	x0	50	
	y0	=AG10*SIN(AG7*3.14159/180)	
	v0	=AG20	
	angle	=90-$AG$7	angle of elevation of projectile
	dt	0.01	time step
	floor	0	altitude of floor (m)
	theta0	0	starting angle of ball (deg)
	omega0	0	starting rate of rotation of ball (deg)
drag	m	=$AG$16	mass of projectile (kg)
	p	1.225	air density (kg/m^3)
	area	=($AG$14/100)^2*PI()	cross sectional area of projectile (m^2)
	CD	0.47	drag coefficient, ~0.9 for sphere
	drag	=P6*0.5*$P$11*$P$12*$P$13/$P$10	dt*0.5*p*a*CD/m, calculated here for efficiency
gravity	g	-9.81	gravity (m/s^2)
	g*t	=$P$15*$P$6	g * timestep, calculated here for efficiency
bounce	kx	0.4	change in x velocity on bounce
	ky	-0.4	change in y velocity on bounce
	domega	-30	coefficient of roll of ball on bounce
error	ystart	=IF($P$3<$P$7;"ERRORRRRR";".")	returns ERROR in case of an undergound start
output	air time	=INDEX(A3:A10002; MATCH(1;D3:D10002;0))	time spent in air by ball before first bounce (s)
	dist	=INDEX(B3:B10002; MATCH(1;D3:D10002;0))-P2	distance before first bounce (m)
	maxH	=MAX(E2:E10002)	max height reached by projectile (m)
	dimp	0.1	distance taken for projectile to come to a stop on collision (m)
	vimp	=SQRT(VLOOKUP(P21-P6; A4:G10002;6)^2 + VLOOKUP(P21-P6; A4:G10002;7)^2)	absolute velocity at moment before impact (m/s)
	Fimp	=0.5*P10*P25^2/P24/1000	impact force caused by projectile at end of first bounce (kN)
Placeholder proje			

Appendix 4: Home Page

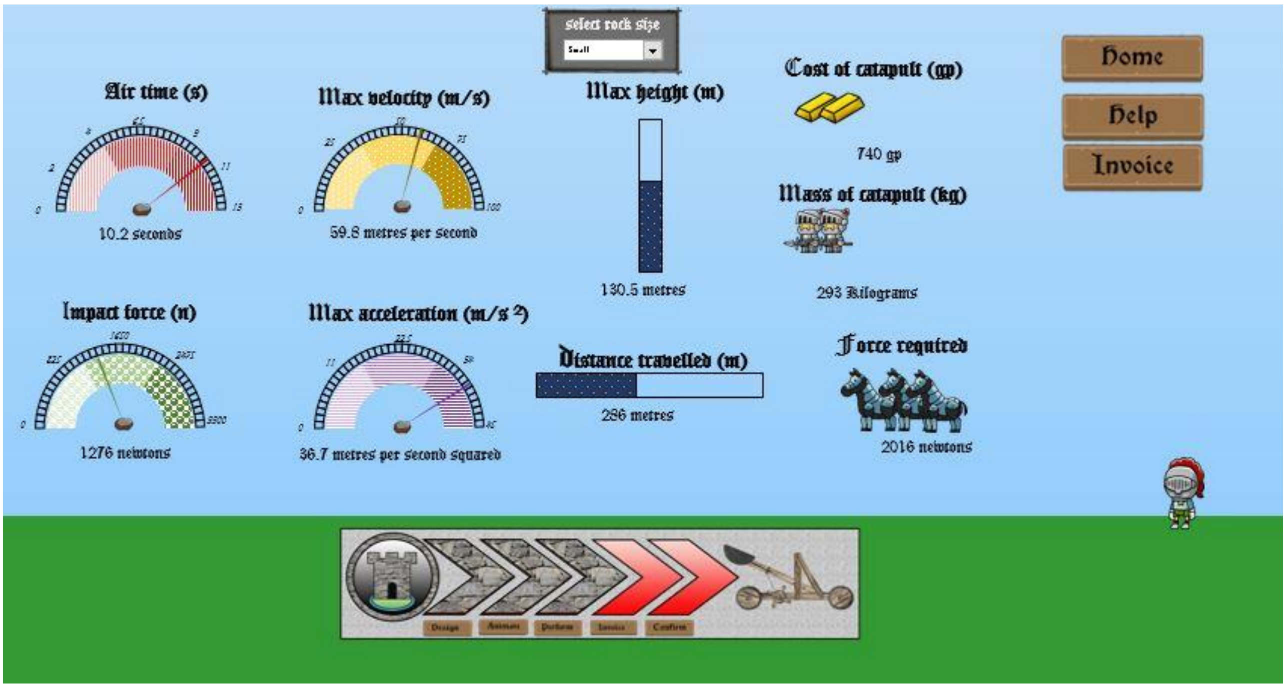
Appendix 5: Design Page



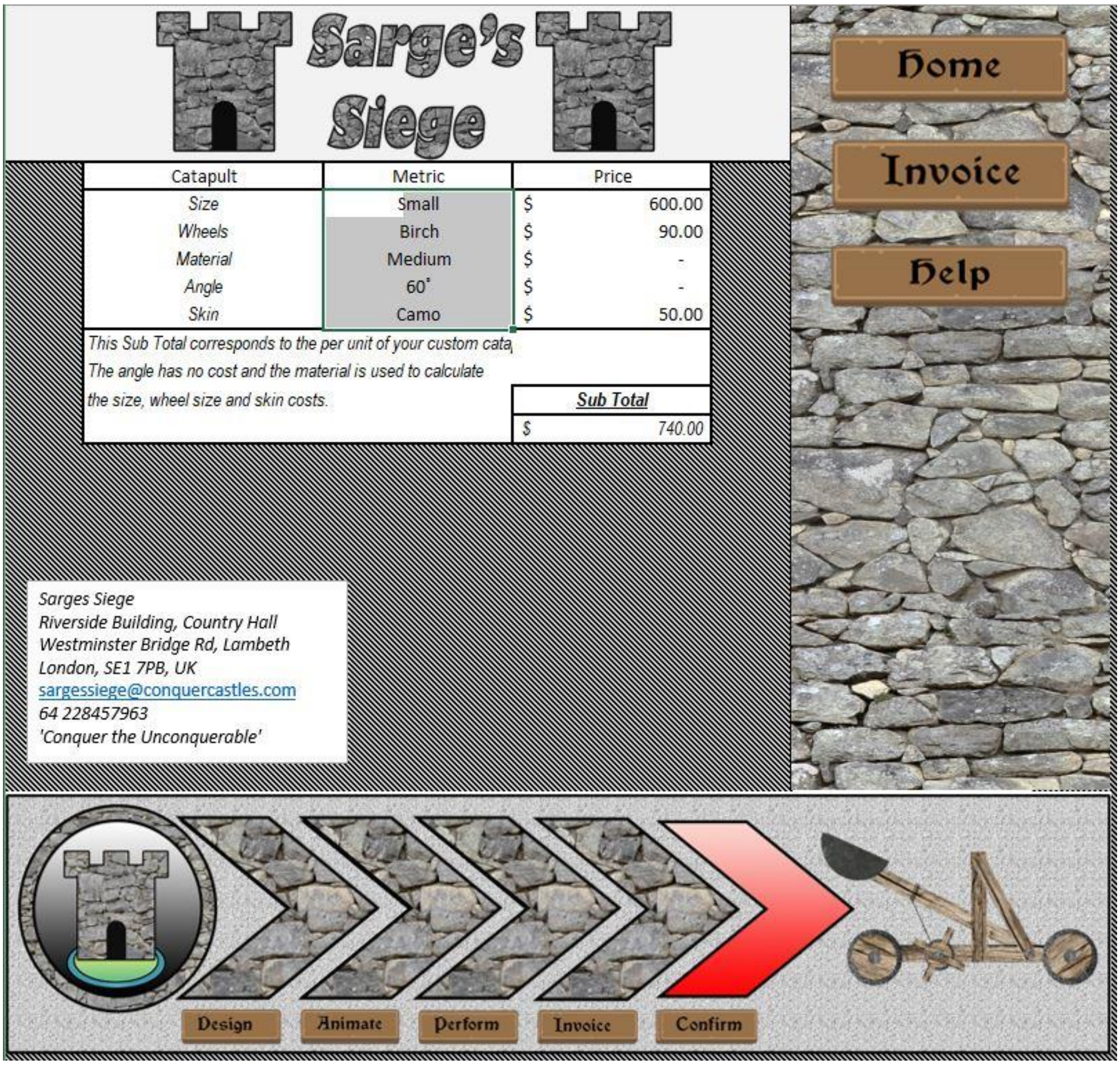
Appendix 6: Animation Page



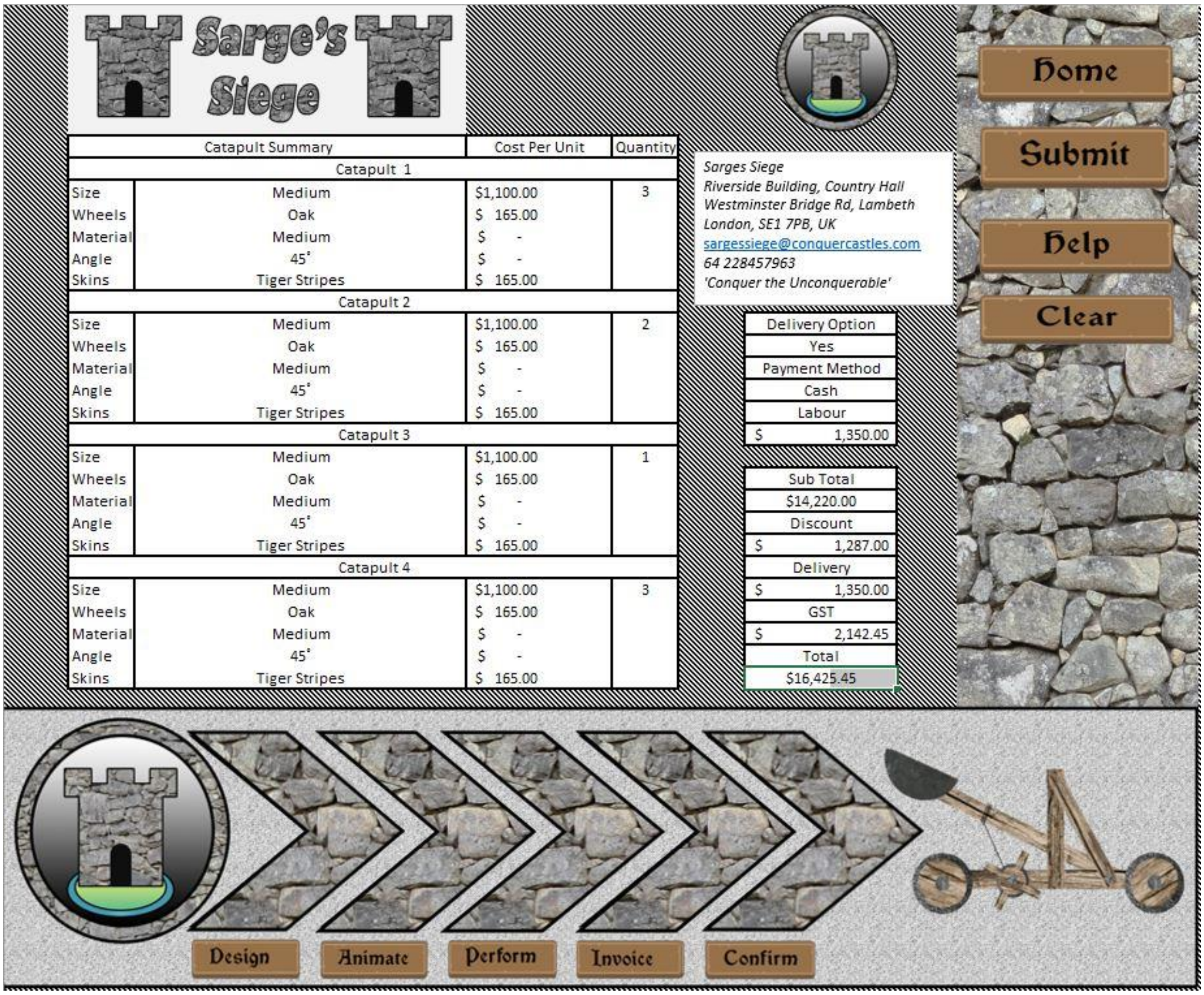
Appendix 7: Performance Page



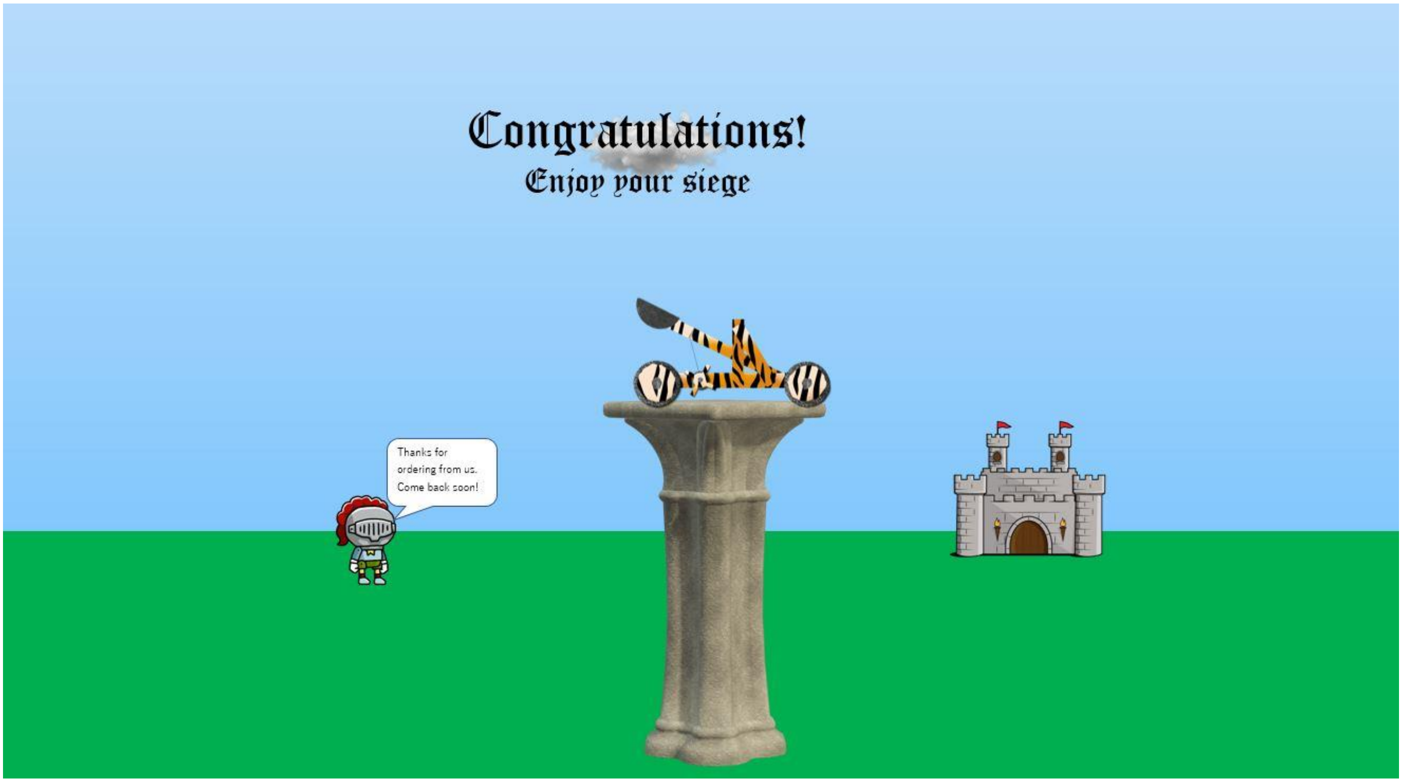
Appendix 8: Intermediary Invoice



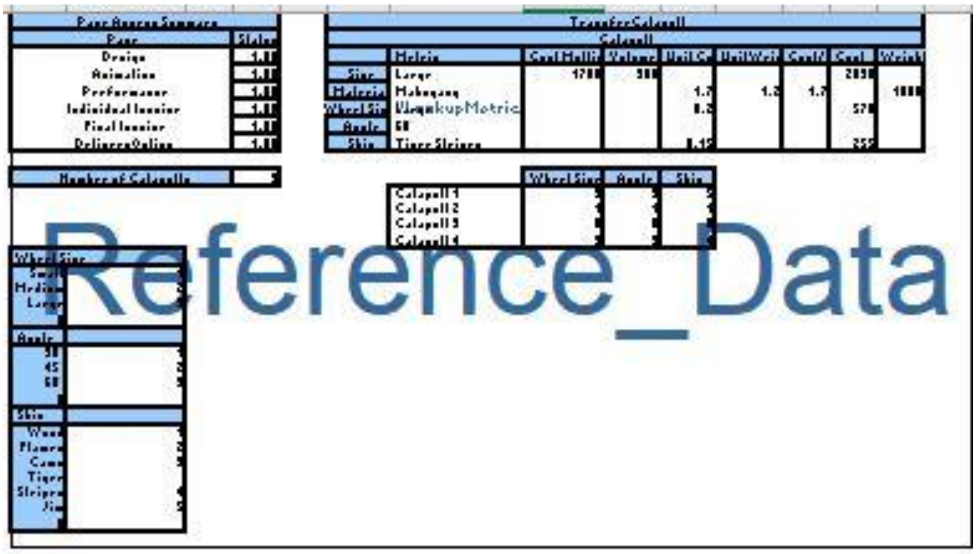
Appendix 9: Final Invoice



Appendix 10: Summary



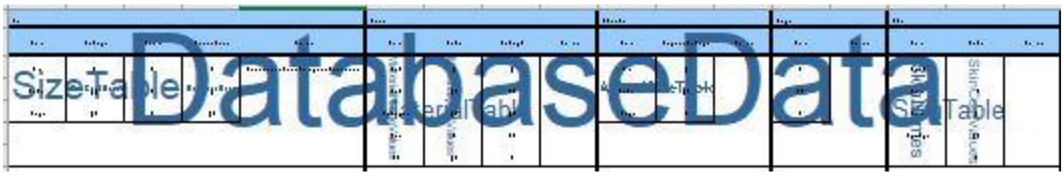
Appendix 11: Reference Data



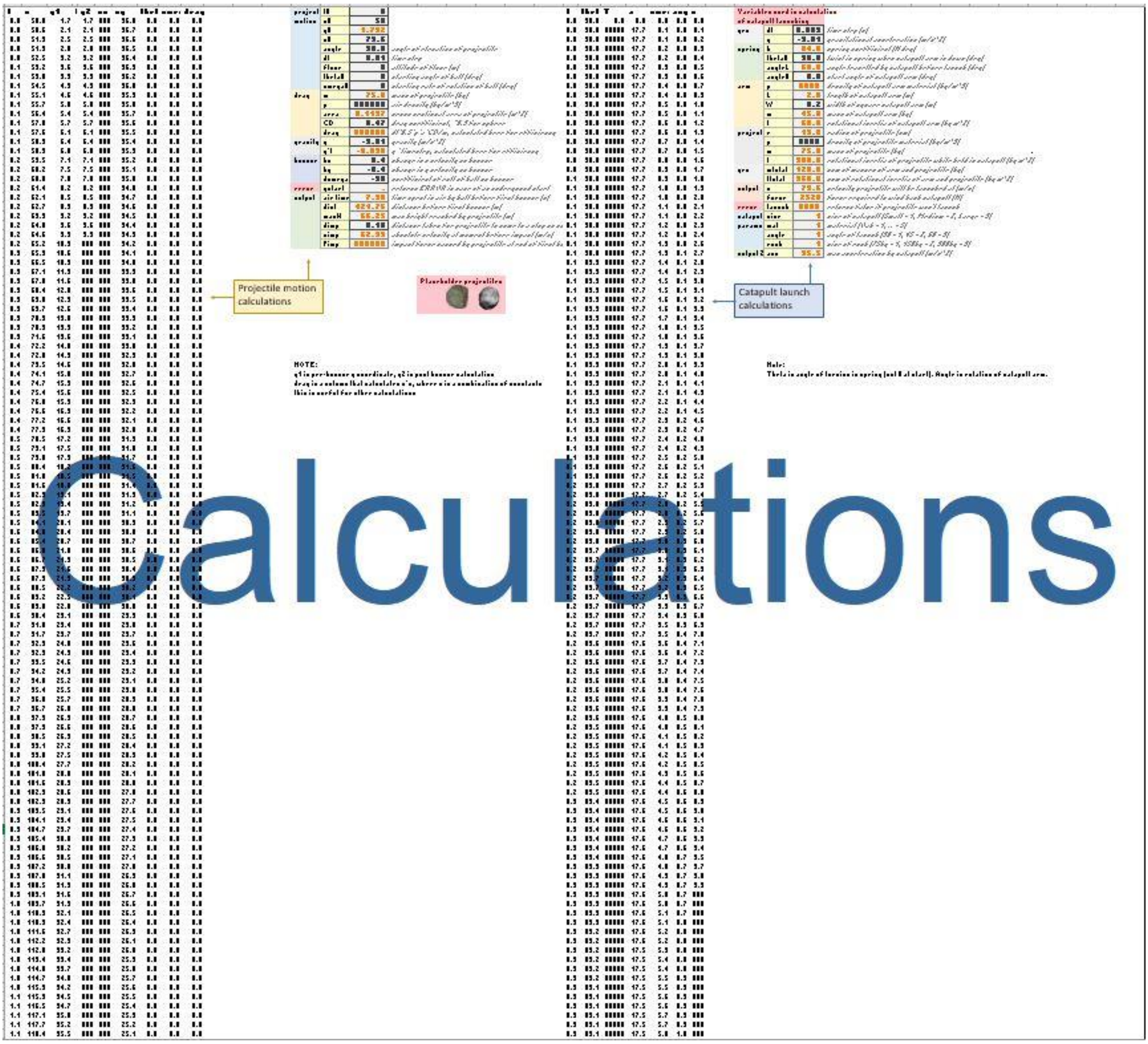
Appendix 12: Control Variables

Catapult Variable Selection		
Title	**Comment**	**Value**
Current Selections		
Size	The current selection for 'size'	1
Material	where 1 = small, 2 = medium etc	1
Wheel Size		1
Angle		1
Skin		1
Displays		
Latest Change Type	What the last selection change was	Skin
Latest Change Numerical Value	What choice the most recent selection was (i.e. change type is size, change value is 3 (large)	1
Latest Change Title	Font & font size is copied over so must be preset in this cell	Skin: Wood
Current Cost Display	These are done here as textboxes don't like concatenation	total cost: 550G
Current Weight Display		total weight: 330kg
Current Unit Cost		cost: 0G
Cost & Weight		
Current Unit Cost	i.e. how much the currently selected skin costs	0
Current Total Cost	Rounded down to the nearest integer	550
Current Weight	Rounded down to the nearest integer	330
Max Possible Cost		3978
Min Possible Cost		495

Appendix 13: Database



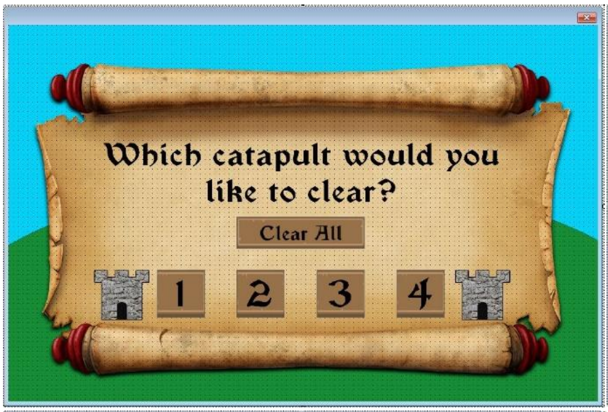
Appendix 14: Calcsheet

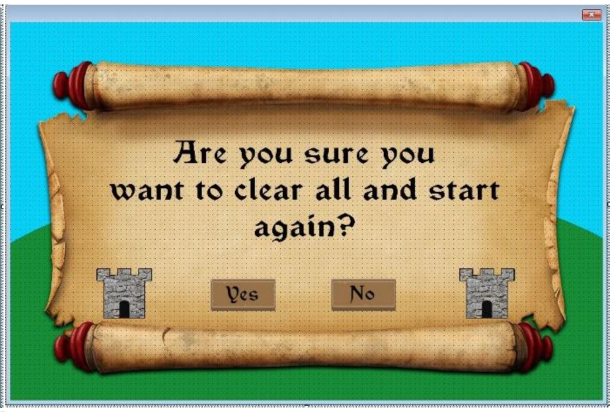


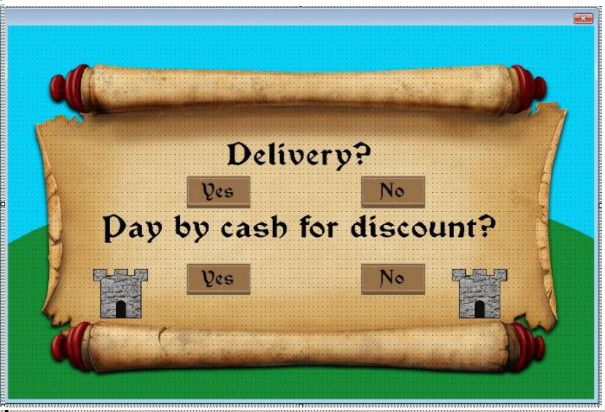
Appendix 15: ResultsPage

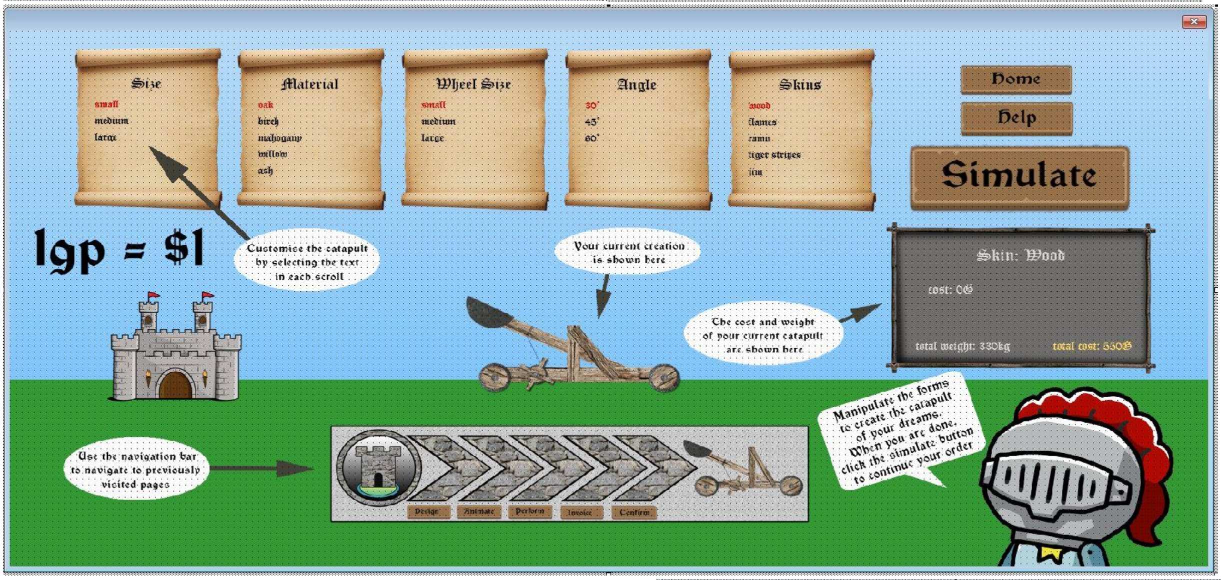
Gauge	Velocity	Chart Axis	Max Valu	185		73.6 metres per second	metres per second
90	133.4902	187	Min Valu	0		35.5 metres per second squared	metres per secoı
30	2	5	Mid Valu	95		1488 newtons	newtons
30	224.5098	5	Small			7.3 seconds	seconds
30	Air Time	5	Medium			422 metres	metres
Height	103.88462	5	Large			66.3 metres	gp
30.13636364	2	5	Other value, needed for dial to work (360-formula-pointer)			550 gp	Kilograms
69.86363636	254.11538	5				330 Kilograms	
Length	Force	5				2520 newtons	
64.92307692	83.418182	5					
35.07692308	2	5	Size of pointer.				
	274.58182	5					
	Acceleratic	5	Formula to convert values for use in dial.				
	145.94444	5					
	2	5					
	212.05556	5					
Max Velocity	102	5	Max Values are the maximum possible value these can be. For calculation purposes.				
Max Accel	45	5					
Max Impact Force	3300	5					
Max Air Time	13	5					
Actual Velocity	73.6	5	Actual Values change based on size of rock.				
Actual Accel	35.5	5					
Actual Force	1488	5					
Actual Air Time	7.30	5					
Max Distance	650	5					
Max Height	220	5	Max and Actual values for use in bar graph.				
Actual Distance	422.00	5					
Actual Height	66.30	5					
Actual Cost	550	5					
Actual Mass	330	5					
Actual Force Requi	2520	5					
Max Force Require	4700	5					
		5					
		5					
		5					
		5					
		5					
		5					
		5					

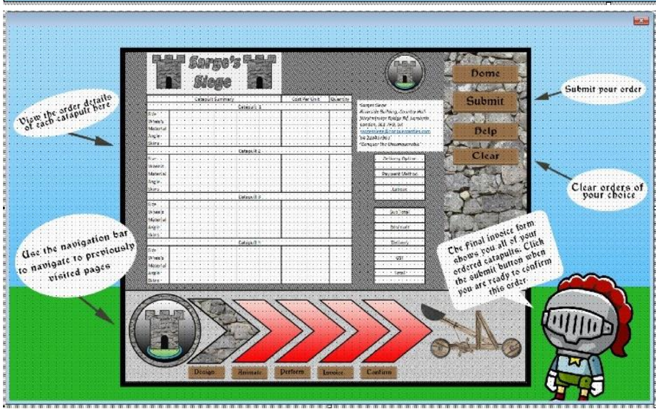
Appendix 16: User forms

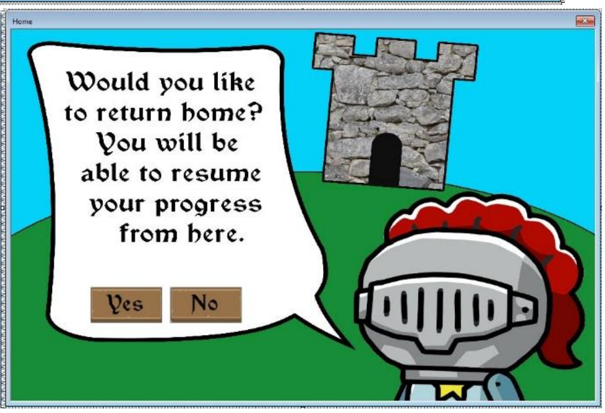


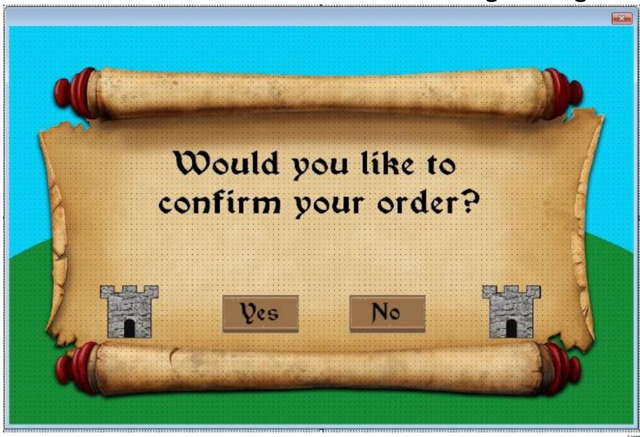




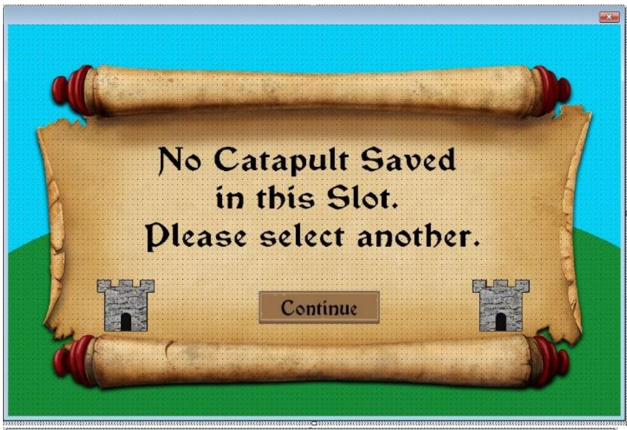


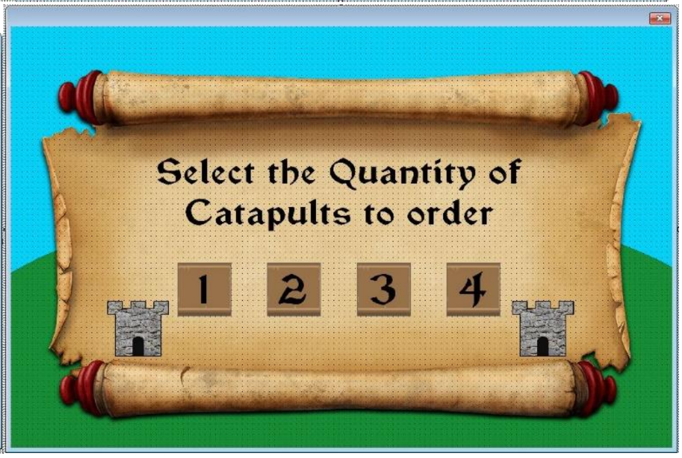


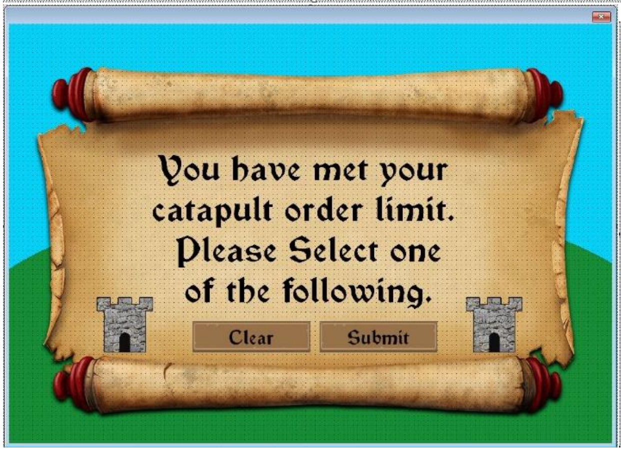


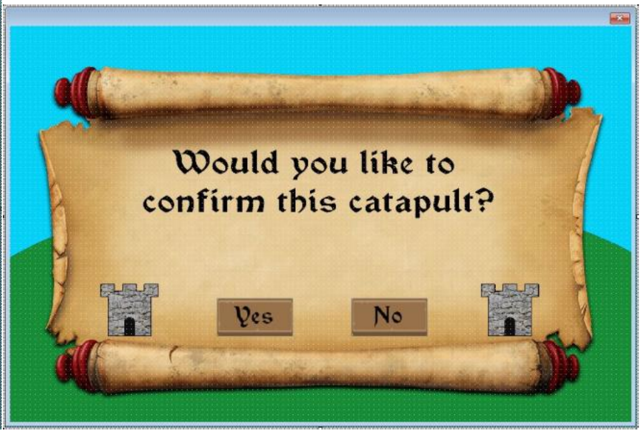

Would you like to confirm your order?
Yes
No

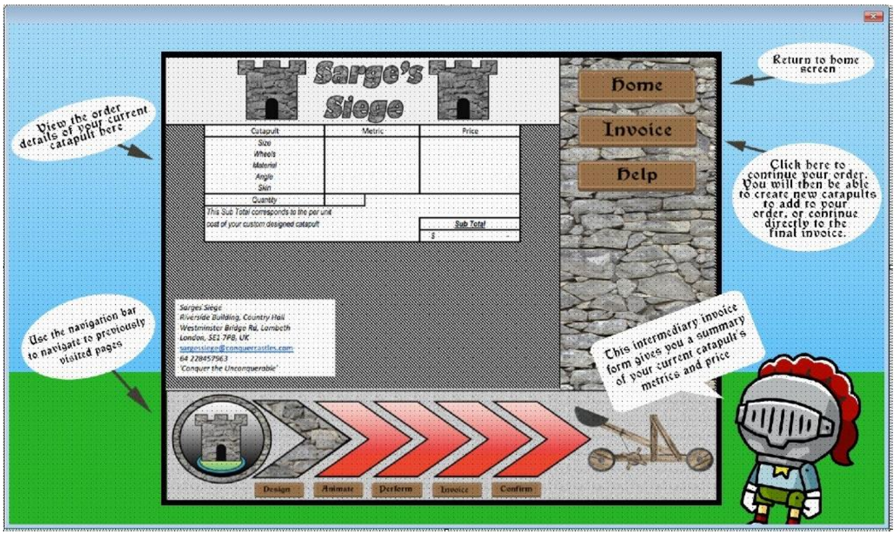

You have no saved catapults
Start Order
Stop Order


No Catapult Saved in this Slot. Please select another.
Continue


Select the Quantity of Catapults to order
1
2
3
4


You have met your catapult order limit. Please Select one of the following.
Clear
Submit


Would you like to confirm this catapult?
Yes
No


View the order details of your current catapult here
Sarge's Siege
Catapult
Metric
Price
Size
Wheels
Material
Angle
Skin
Quantity
This Sub Total corresponds to the per unit cost of your custom designed catapult
Sub Total
$
Sarges Siege
Riverside Building, Country Hall
Westminster Bridge Rd, Lambeth
London, SE1 7PB, UK
sargessiege@conquercastles.com
64 228457963
'Conquer the Unconquerable'
Use the navigation bar to navigate to previously visited pages
Home
Invoice
Help
Return to home screen
Click here to continue your order. You will then be able to create new catapults to add to your order, or continue directly to the final invoice.
This intermediary invoice form gives you a summary of your current catapult's metrics and price
Design
Animate
Perform
Invoice
Confirm

Appendix 17: Sub Routines and Functions

Module	Sub Routine and Function
Animation	Animate() ClearSheet() DrawScene(sC As String, sA As String, size As Double) DrawAnimation(sC As String, sA As String) DrawFrame(sC As String, sA As String, r As Integer, proj As Variant, shad As Variant, curt As Variant, cast As Variant, fire As Variant) Sleep(Duration As Integer) Small_rock_click() Medium_rock_click() Large_rock_click() PositionCastle()
Buttons	Begin() HelpDesign() HomeButton() Begin() SimulateButton() ResultsButton() InvoiceFromPerformance() InvoiceButton() HelpAnimationButton() HelpPerformanceButton() HelpIndividual() HelpFinal() SubmitButton() ClearFinal() NavigateDesignButton() NavigateAnimationButton() NavigatePerformanceButton() NavigateIndividualInvoiceButton() NavigateFinalInvoiceButton() UpdateReference(CellReference)
Calculations	Function GST(Value) Function Cost(Quantity, Rate) Function Discount(Check, Value, DiscountRate) Function TotalCostPerUnit(Sheet, CellReference) As Double Sub UpdateAllCosts() Sub UpdateCount()
Clearing	Sub ClearRange(Sheet, Range) Sub ResetReferenceData() Sub ClearFinalInvoice(Counter) Sub ClearAll()
DesignAndInputs	Sub ProtectSheet() Sub SetButtonColours(selectedVariable, selectedButton) Function GetCatapultFilename() As String Sub SetCatapultImage(selectedVariable) Sub ButtonSize1_Click() Sub ButtonSize2_Click() Sub ButtonSize3_Click() Sub ButtonMaterial1_Click()

	Sub ButtonMaterial2_Click() Sub ButtonMaterial3_Click() Sub ButtonMaterial4_Click() Sub ButtonMaterial5_Click() Sub ButtonWheelSize1_Click() Sub ButtonWheelSize2_Click() Sub ButtonWheelSize3_Click() Sub ButtonAngle1_Click() Sub ButtonAngle2_Click() Sub ButtonAngle3_Click() Sub ButtonSkin1_Click() Sub ButtonSkin2_Click() Sub ButtonSkin3_Click() Sub ButtonSkin4_Click() Sub ButtonSkin5_Click() Function GetUnitCost()

Formatting	Sub FormatFinalInvoice(Sheet, Counter)
	Sub FormatReferenceData(Sheet, Counter)
	Sub FormatFinalInvoiceTotals(Sheet)
	Sub CentreAccrossSelection(Sheet, CellRange)
	Sub CreateBorder(Sheet, CellReference)

HomeAndSummary	Sub Pedestal() Sub castle() Sub CloudMovement(SheetName, CloudName) Sub timeout(duration_ms As Double) Sub Help() Sub Help2() Sub Help3() Sub Help4() Sub WaitFor(NumOfSeconds As Long) Sub CreateZoomWindow(ByVal ZoomThisRange As Range, _ ByVal PreserveRows As Boolean) Sub SetCatapultImage1() Sub SetCatapultImage2() Sub SetCatapultImage3() Sub SetCatapultImage4() Function GetCatapultFilename1() As String Function GetCatapultFilename2() As String Function GetCatapultFilename3() As String Function GetCatapultFilename4() As String
Navigate	Sub NavigateYes(UserForm, SheetObject) Sub NavigateNo(UserForm) Sub ShowTime(UserForm) Function CatapultCounter() As Integer Sub NavigateButton(SheetDestination) Sub ShowImageCondition(SheetDestination, SheetReference, ImageReference, CellReference) Sub Navigator(SheetDestination, SheetReference, ImageReference, CellReference) Function CheckReference(SheetReference, CellReference) Function CheckAllocation() As Integer Sub UpdateReference(CellReference) Function CheckSaveSlot(CellReference) As Integer Function CountCatapults()

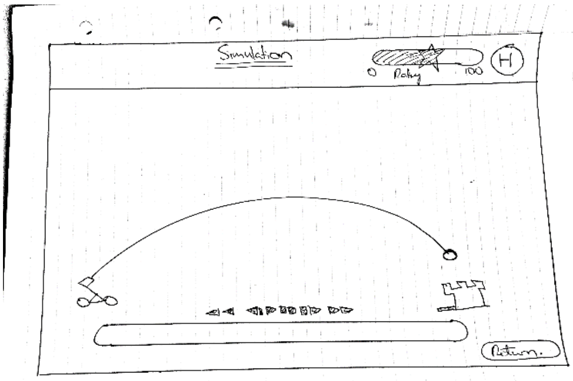
Performance	Sub rockSize() Sub visible_Images()
Saving	Sub SaveIndividualCatapult(SheetTo, SheetFrom, Counter, Quantity) Sub TransferCatapultImage(Count) Sub TranferCatapultDesign(SheetTo, SheetFrom, ValueRange, ReferenceRange) Sub Save(Quantity)

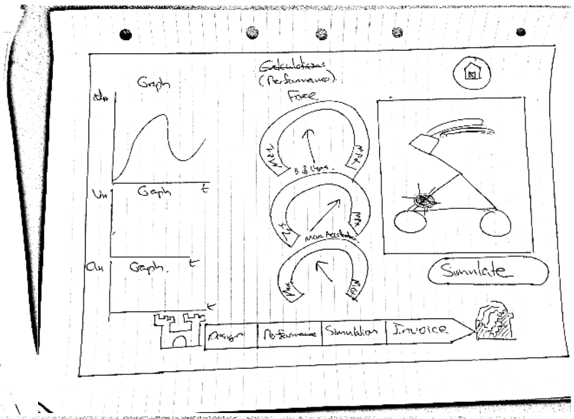
```
Function GetCatapultFilename() As String
    ' Gets the filename for the image on disk based on the current selection
    Dim WheelSize As String, Angle As String, Skin As String

    ' Get the values and names of the current selection
    CurrentSkinSelection = Sheets("Control Variables").Range("SkinSelection").Value
    Skin = Application.WorksheetFunction.Index(Sheets("Database").Range("SkinNames"), CurrentSkinSelection)
    CurrentAngleSelection = Sheets("Control Variables").Range("AngleSelection").Value
    Angle = Application.WorksheetFunction.Index(Sheets("Database").Range("AngleNames"), CurrentAngleSelection)
    CurrentWheelSizeSelection = Sheets("Control Variables").Range("WheelSizeSelection").Value
    WheelSize = Application.WorksheetFunction.Index(Sheets("Database").Range("WheelSizeNames"), CurrentWheelSizeSelection)

    ' Concatenate to get the filename
    GetCatapultFilename = Skin & Left(Angle, 2) & WheelSize
End Function
```

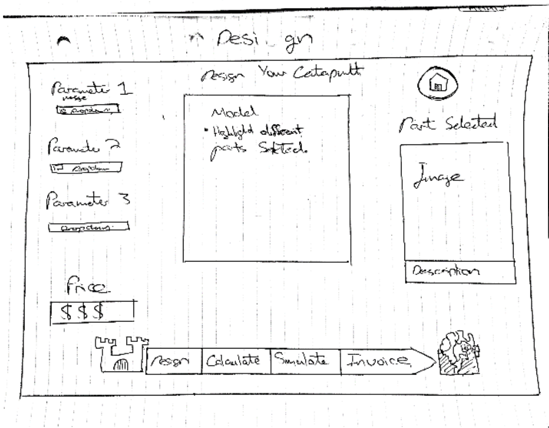
```
Sub SetCatapultImage(selectedVariable)
    ' Updates the image of the catapult based on the current selection
    Dim defaultWidth As Integer
    ' The defalt image width
    defaultWidth = 900
    'Set sheet'
    Set Sheet = Worksheets("Design")

    ' Get the current size selected (small/medium/large) and find the relative scale (multiplier for the default width)
    CurrentSizeSelection = Sheets("Control Variables").Range("SizeSelection").Value
    CurrentSizeScale = Application.WorksheetFunction.Index(Sheets("Database").Range("SizeScaleValues"), CurrentSizeSelection)

    ' Calculate the new width
    newWidth = defaultWidth * CurrentSizeScale

    ' Get the image filename and path to it
    Filename = GetCatapultFilename()
    Directory = ThisWorkbook.Path & "\CatapultImages\" & Filename & ".png"

    ' If a catapult image already exists, delete it
    On Error Resume Next
    If Sheet.Shapes("CatapultImage").Name <> "CatapultImage" Then
        ' Catapult image doesn't exist on sheet
    Else
        Sheet.Shapes("CatapultImage").Delete
    End If
    On Error GoTo 0

    ' Check if the directory is valid
    If Dir(Directory) <> "" Then

        ' Insert a new image of the selected size and reposition it
        With Sheet.Pictures.Insert(Directory)
            .ShapeRange.PictureFormat.CropTop = 130
            .ShapeRange.PictureFormat.CropBottom = 130
            .ShapeRange.LockAspectRatio = msoTrue
            .ShapeRange.Width = newWidth
            .ShapeRange.Left = Sheet.Range("R40").Left - .ShapeRange.Width / 2
            .ShapeRange.Top = Sheet.Range("R33").Top - .ShapeRange.Height / 2
            '.ShapeRange.ZOrder msoSendToBack
            .Name = "CatapultImage"
        End With
    End If

End Sub
```

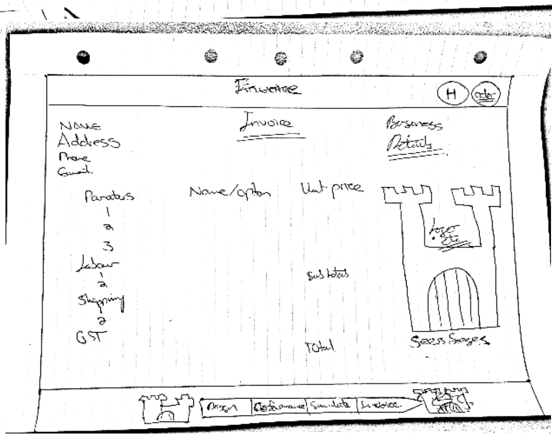
For practical purpose, we listed all sub routines and function names and not the functions themselves. The code for each sub routine and function may be viewed in the Microsoft Visual Basic Window.

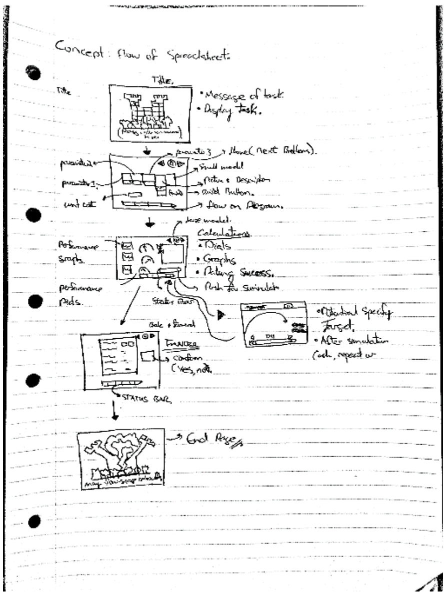
Appendix 18: Concept Sketches











Sarge's Siege: A Solidworks Suitability Investigation

Connor McDowall

5309133986

Group 10

Department of Engineering Science

16th October 2017

Executive Summary

Solidworks is an Computer Aided Design Program with Visual Basic for Application Libraries and an extensive Application Programming Interface. An investigation was undertaken to investigate the suitability of implementing Solidworks within the workplace of Sarge's Siege.

Solidworks is Intuitive and easy to learn. Dassault Systems installed comprehensive tutorials and has set up an online community, enabling fast learning and adoption of Solidworks.

The modelling of an axial, a component of a catapult, was used to investigate the suitability of the product. The component of an axial was split into five components; Wheel Rim Scalable, Tire, Swivel Mechanism, Wheel Fastener and Drive Axial. Each part was constructed using the functionalities of Solidworks. A combination of sketches, extrudes, revolves and patterns were used to model each part. Each component was assembled using standard, advanced or mechanical 'mating' relationships in an assembly. Solidworks can produce drawing to convey information, demonstrated by the Swivel Mechanism and Axial drawings.

Solidworks API and VBA libraries allow model parameters to be changed by assigning pointers to the Solidworks application from the excel workbook. Parameters are modified through replacing values or strings. The global variables manipulated where; the radius of the wheels linked to the wheel size, the length of the axial, linked to the catapult size and the material of the wooden components.

Solidworks was a great learning experience with smart, intuitive features and comprehensive guidance.

Solidworks is intuitive and comprehensive. The functionalities of the program are easy to use. Model parameters are easy to manipulate external from an excel workbook using subroutines. It is recommended Sarge's Siege adopt Solidworks.

Table of Contents

Table of Figures

1 Introduction

The market for Computer Aided Design (CAD) products is saturated. There are many alternative products capable of producing three dimensional models, sketching and performing analysis. Solidworks is a CAD program specifically targeted towards engineers. Solidworks has an extensive list of Visual Basic for Application (VBA) libraries and an extensive Application Programming Interface (API). Sarge's Siege have used Solidworks to develop a model and manipulate the model in excel based on the users input, where the API is used to access the VBA programming libraries. The end goal was to use excel to manipulate the Solidworks model's parameters. Solidworks was the focus for our suitability investigation to reach a conclusion on whether the CAD package would meet our scopes of use.

2 Assessment

2.1 Ease of Use

CAD program must be intuitive and easy to use to aid firm wide implementation. Solidworks meets this classification. The application is easy to navigate and design with. Solidworks has a series of in-built tutorials, introducing new users to the functionalities of Solidworks. Tutorials include; basics and advanced techniques, productivity tools, design evaluation and more. See Figure 46 in 6.11. The gradient of the learning curve and time invested in learning the package is significantly reduced, excellent for educating draftsman quickly and a fast adoption. The tools in sketching are smart and easy to use. The 'Smart Dimension' tool is the most intuitive. You can dimension any sketch in relation to another by creating a driving, named dimension by two or more simple clicks, dragging, and then placing the dimension. You can create three-dimensional object from two-dimensional sketches through extrudes, revolves, lofts, sweeps and other functions. Assembling different components is easy through 'mating' different parts using standard, advanced or mechanical mates. The drawing wizard within Solidworks is powerful and intuitive. You can create drawings with multiple views, magnified sections and dimensioned. These drawings are useful to convey information to suppliers and inhouse compartments. Solidwork's graphical interface makes modelling, assembly and drawing very easy. There is also an extensive online community with support and learning opportunities (Dassault Systems, 2017).

2.2 Model Creation

A model had to be created to assess the programs suitability. We focused on the front axial component of the catapult due to the time constraints of this report. The findings from the investigation may be apply to the full product. The complexity of our axial led us to build the product in a series of discrete steps. The axial was split into 2 two sets of five components: Drive Axial, Swivel Mechanism, Tire, Wheel Fastener and Wheel Rim Scalable. The dimensions of each part were driven off the two global variables, radius and length. 'Radius' relates to radius of the wheel and 'Length' the width of the axial. The wooden material property is changed in the components as well.

2.2.1 Wheel Rim Scalable

This part was the most complicated to model (Figure 1). We sketched the rims cross section, with all dimensions derived from the radius. The sketch was revolved 360˚ to form the framework of the rim. A second cut extrude made slots on the edge of the rim. A revolve features created the hub cap from a sketch incorporating splines. A cut extrude features made slots in the circular hub cap. Mirroring the previous cut extrude, and using a circular pattern, helped formed the resemblance of a hub cap. The centre of the wheel rim was made from a series of circular cut extrudes and another central revolve around the central axis. The edges of the rim were filleted to increase realism. The dimensions were derived from ratios to scale accordingly. The design of the wheel rim was inspired by one produced by CAD CAM Tutorial (CAD CAM Tutorial, 2016).See Figure 11 through to Figure 16 in 6.1.

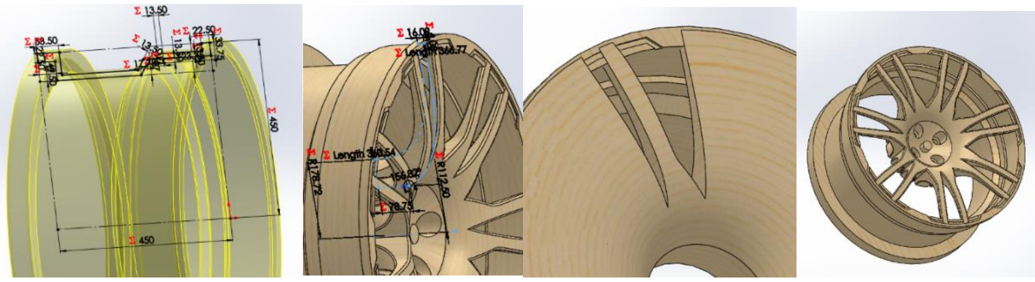


Figure 1: Wheel Rim Scalable Development

2.2.2 Tire

The tire part (Figure 2) was simple relative to the Wheel Rim Scalable. The sketch of the tire's cross section mirrors the Wheel Rim Scalable parts cross section to aid mating in assembly later. Tolerances were not accounted for as unnecessary for determining the suitability of Solidworks. However, tolerances in this part and all parts in the assembly will need to be accounted for use in machining. The sketch was revolved around the central axis. The edges of the tires were filleted to make the tire more realistic. After establishing the base of the tire, the tread was added by sketching the tread on a plane tangential to the tire's edge, using a boss extrude up until the surface of the tire and using a circular pattern to repeat the extrude around the edge of the tire. See Figure 17 through to Figure 20 in 6.2 for an in-depth walkthrough of the tire's creation.

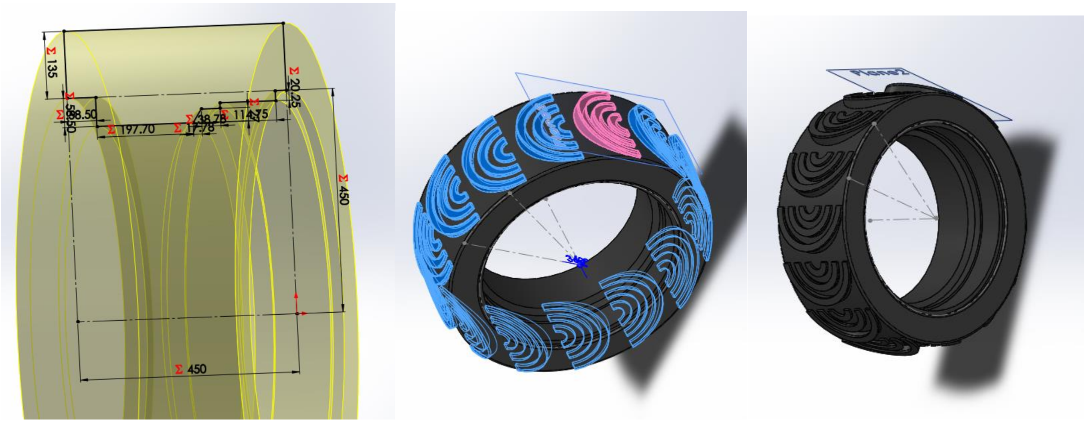


Figure 2: Tire Development

2.2.3 Drive Axial

The next part created was the drive axial, another simple part (Figure 3). First, the basic dimensions were sketched. Instead of using a standard extrude, we chose to use a series of loft extrudes where surfaces are formed between two parallel sketches. Four additional planes were made, each a quarter of the length apart. Circular and spline incorporating sketches were made on the planes with four separate loft extrudes connecting the sketches. To complete the part, two equal-distance circular cut extrudes were made on the axial. An walkthrough of the drive axial's modelling is shown in 6.3, Figure 21 through to Figure 23.

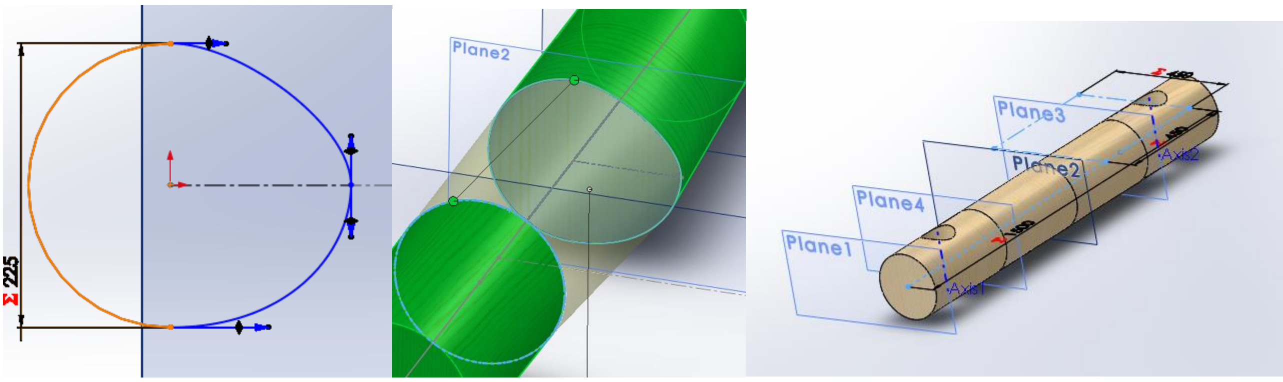


Figure 3: Drive Axial Development

2.2.4 Wheel Fastener

The wheel fastener part connects the wheel rims to the subsequent swivel mechanism (Figure 4). Like the swivel mechanism, this part is a series of simple extrudes. First, a circular extrude is made with four, equally dimensioned, equidistance circular extrudes protruding from on face. On the adjacent, parallel face to the four circular extrudes, another circular extrude is made, hollowed out by an internalised cut extrude. This part connects with the swivel mechanism and turns 360˚ in theory. The model is for demonstration purposes, and, due to time constraints, bearings which enable the turning mechanism were not designed. The bearings and there housing will sit in the cut extrude if this part was to be machined. See 6.4, Figure 24 though to Figure 26.

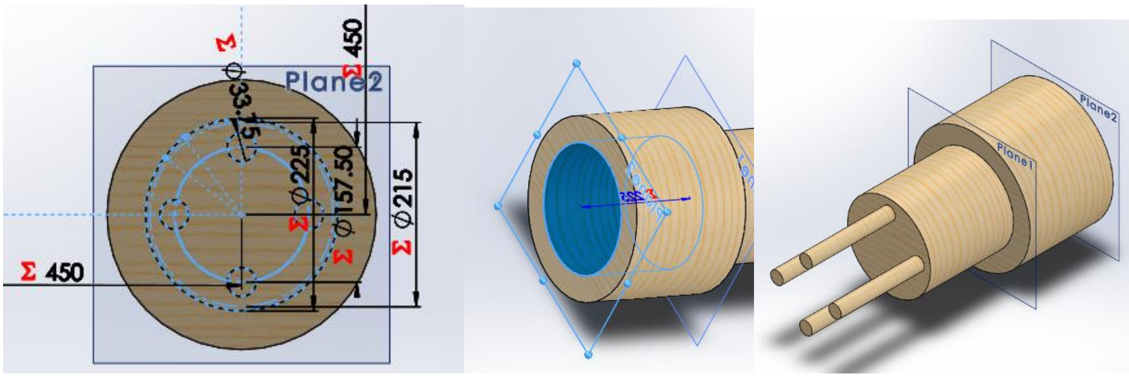


Figure 4: Wheel Fastener Development

2.2.5 Swivel Mechanism

The swivel mechanism connects the drive axial to the wheel fastener. This part was fabricated computationally using several complex functions (Figure 5). First, two circular extrudes with different radii were made off each other. A pin like component was to be incorporated into one of these extrudes. A square cut extrude was made to make room for this pin. After room was made, a circular extrude with two directions was put in place, extruding up to until the surface. The swivel mechanism has an arc shaped extrude, cut, with another pin adjacent to the first. This is a complex feature and was difficult to do at first and define with a set of equations. Two additional planes were made to enable a swept boss. A square sketch is swept along a spline to create the arc. The spline was difficult to define via an equation at first. We initially used one quarter of an ellipse's parameter. Determining the equations for the perimeters of an ellipse is very difficult as is calculated by approximations or two sets of infinite series (Maths is Fun, 2017). To circumnavigate this complexity, a simple spline equation was used. To complete the part, the Swept blend was cut out and the second pin added. See Figure 27Figure 29 in 6.5 from an in depth walkthrough.

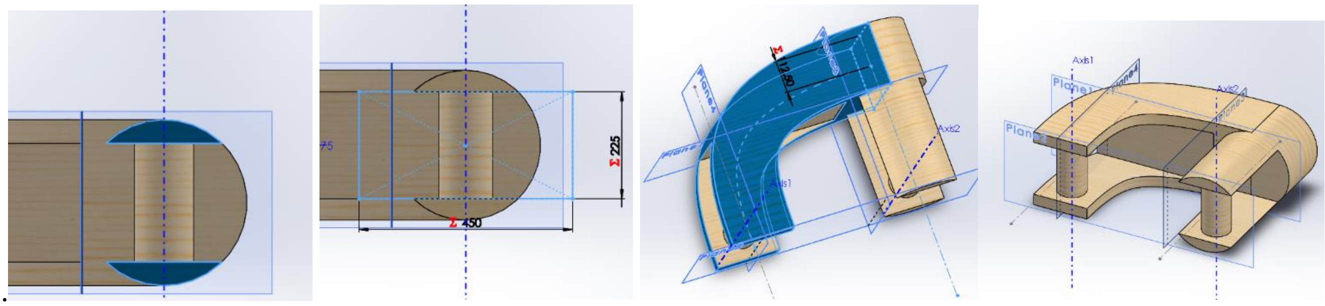


Figure 5: Swivel Mechanism Development

2.2.6 Assembly

After finalising the model components, they can each be individually placed in an assembly (Figure 6). Solidworks allows you to mate individual components together. There are standard, advanced and mechanical mates. The wheel rims, tires and wheels fasteners are mated using standard co-incident and concentric mates. Mechanical hinge mates were used to mate the wheel fasteners and swivel mechanisms. The hinge constraint combined the concentric, co-incident and rotational constraints to enable the wheel fastener and swivel mechanism to rotate with one degree of freedom. If the axial was to be machined and assembled, a slot mate would be used. The swivel mechanism and drive axials were mated using a combination of hinge and concentric constraints. The final assembly is under defined due to the rotational permissions from the hinge mates, allowing the wheels to rotate 360˚ and swivel 40˚ (Figure 7). See Figure 30 through to Figure 34 in 6.6 for the final images of all individual parts and Figure 35 through to Figure 39 in 6.7 for pre assembly, multiple assemblies and a mating pair.

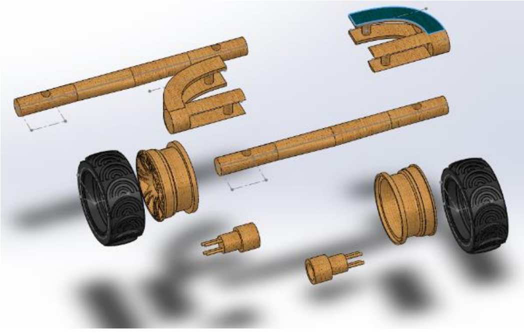
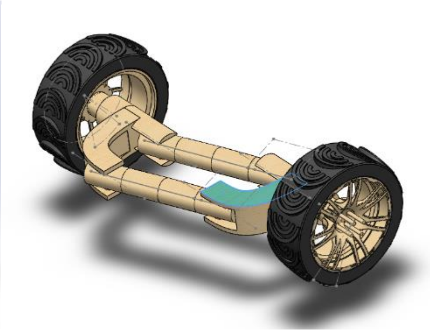



Figure 6:Individual Components placed in Assembly (Left)

Figure 7: Completed Assembly (Right)

2.2.7 Drawings

Once the assembly is complete, Solidworks has the capability of producing drawings of our parts or assemblies. Drawings are an excellent way to convey the part or assembly information. Different views of the part may be displayed on the drawing. We have displayed a top, right, front and isometric view of the Axial assembly and the Swivel Mechanism part. Dimensions may be added to individual views or the entire model. You can also produce section and detailed view to drawings. Solidworks allows you to add reference and machining information to the drawings files, enabling the accurate filing and revision of drawings. Assemblies have many dimensions from aggregate parts, cluttering the sheet. It is better to display the assembly without drawings and dimension the individual parts. See Figure 40 for the axial drawings and Figure 41 for the swivel mechanism drawings in 6.8.

2.2.8 Additional Features

Solidworks is a complex program with many other features. There are; productivity tools, animation, simulation, costing, flow analysis, dimension expert tools, mould designs and more (Figure 46,6.11). After the modelling of your product, you can test its suitability though simulation. Solidworks also has rendering capabilities to present your model in a high presentation quality in Dassault System's premium product. In addition to rendering, background images may be set in the assembly planes (Figure 45, 6.11). Solidworks has the ability to install add ins to use new features, properties and increase the functionality of the product.

2.3 Implementation within the Excel Workbook

After the assembly was finalised and the dimensions were appropriately referenced global variables, the next step was to integrate the excel workbook with Solidworks and the model. To begin with, the Visual Basic Solidworks API was investigated, learning the capabilities. Excel is able to send, receive and manipulate information to, from and in the powerful API. The Solidworks Application is accessed as a VBA object, using pointers in sub routines to point to, activate and manipulate objects in the API. This functionality allows parameters within Solidworks to be manipulated. The relevant file in Solidworks must be open to make changes, enabling the use of 'pointers'. The exert of code to establish pointers to the Solidworks application, assembly files, part files and the equation manager is in Figure 8.

```
'Set Assembly Name'
assembly = "Axial.SLDASM"
'Set pointer to the Solidworks Application'
Set swApp = CreateObject("Sldworks.Application")
'Make the assembly the active document model'
swApp.ActivateDoc (assembly)
'Set pointer to make changes to the assembly'
Set objFile = swApp.ActivateDoc(assembly)
'Set pointer to Equation manager tool'
Set swEqnMgr = objFile.GetEquationMgr
'Change the Radius and Length Global variables'
```

Figure 8: VBA pointers to Solidworks Application, Parts, Assembly and Equation Manager

After the code assigns the assembly and equation manager as the current objects, adjustments are made to the radius and length global variables saved in the assembly. The modifications to the global variables are based on the user's input in the excel worksheet. Excel VLOOKUP functions create a Solidworks manipulation table, depicted in Figure 9. The table resides in the "ReferenceData" sheet in the excel workbook. After selection, the user can press the Solidworks button to the right of the table to manipulate the model. The use of pre-determined sizes and materials matches the functionality of our previous excel workbook.

Solidworks Dimensions and Parameters (mm)						
	Size	Wheel Size	Material		Solidworks Parameters	
Small	2000	300	Oak	Oak	Size	3000
Medium	2500	450	Mahogany	Mahogany	WheelSize	300
Large	3000	600	Birch	Pine	Material	Mahogany
			Willow	Teak		
			Ash	Cedar		

Figure 9: Solidworks Modification Table

The methodology of updating the model in Solidworks is very simple. The global variables 'Radius' and 'Length' can be manipulated in the equation manager (Figure 42, 6.90. The global variables are stored as strings. The values stored can be converted to string, split removing the old value and replacing the old value in the global variable newly converted string. The index position of each global parameter is known. The snippet of code is shown in Figure 10.

```
'Convert Values in the excel spreadsheet into strings'
length = CStr(Worksheets("ReferenceData").Range("SolidworksWheelSize"))
radius = CStr(Worksheets("ReferenceData").Range("SolidworksSize"))
'Split the global variable storage equation strings at the equals sign'
strSplit1 = Split(swEqnMgr.Equation(0), "=")
strSplit2 = Split(swEqnMgr.Equation(1), "=")
'Append the length and radius value onto the end of the strings'
swEqnMgr.Equation(0) = strSplit1(0) & "= " & length
swEqnMgr.Equation(1) = strSplit2(0) & "= " & radius
```

Figure 10:VBA Updating Global Variables

The global variables 'Radius' and 'Length' are linked to the 'Radius' and 'Length' global variables in each part. Every dimension, excluding angles, is referenced to these variables. 'Length' adjusts the length of the axial, attributable to the size of catapult. 'Radius' adjust the radius of the wheels, attributable to the wheel size. The dimensions scale according to the global variables, retaining assembly mates and definitions. The material parameters are updated on each part in the assembly by resetting the material name in the material properties manager (See 6.9, Figure 43), one at a time. Solidworks had oak and mahogany as material options but did not have Birch, Willow and Ash. I used Cedar, Teak and Pine as placeholders to import later. Once all updates have been made, the assembly is rebuilt with the user able to view their custom-built axial related to their catapult design.

The entirety of the code was input into the "SolidworksIntegration" module as one subroutine, "SolidworksManipulator". A copy of this subroutine may be found in Figure 44, 6.10. When the user reaches the "Individual Invoice" sheet and confirms their catapult design, a create more catapults userform will show. If the user chooses no, the "SolidworksManipulator" subroutine will be called. The user will be presented with a prompt, asking the user to have all relevant Solidworks files open before continuing. This is to guide the user to open all files and avoid a confusing error message. The user follows the design process in the excel workbook before confirming the catapult. The user will only see updates to their model if all files are open and they selected yes in the prompt. If the user selects no, the subroutine to update the model will not run.

3 Learning Experiences

Overall, this investigation proved useful and informative. The Solidworks tutorials are informative, requiring less external exploratory based learning experienced with other CAD packages. The creator functions in Solidworks are intuitive and straightforward to use. Part creation, the assembly window and the drawing environment were simple to learn. The drag and drop functionality for all tools was fun and engaging. Solidworks was helpful with error messages, guiding you to the right methods such as fully dimensioning drawings and defining assemblies. The use of equations to drive dimensions is brilliant. I had some difficult with fully dimensioning drawings and accessing the correct indices and value of global variables in the equation manager. All other errors were quick fixes, either adjusting syntaxes or changing dimensions to reference equations. The VBA was intuitive, apart from the concept of global variables stored as strings. Using VBA to access the API of Solidworks was an enlightening experience. I look forward to investigating and experimenting with accessing other software packages using VBA. The online forum set up by Dassault Systems, the creator of Solidworks was insightful on the rare occasion the Solidworks user interface could not help. Solidworks is a great product. I planned out my parts in the beginning, performing sketches of the axial before modelling. I was set on what I wanted to design in the beginning. Next time, I will spend more time at the beginning iterating concepts to find the best possible solution. I used my knowledge from PTC CREO, assuming Solidworks had the same functionality. This worked well however I will look to investigate other features in Solidworks first to ensure I don't miss using a more appropriate tool. The way I integrated excel and VBA was very simple. I will look into thoroughly completing the VBA and API tutorials for future products to gain a better understanding of the interface, whilst, expanding my capability on what I can do with VBA and the Solidworks product. I have loved my experience with Solidworks and cannot wait to use it in other academic and personal projects.

4 Conclusions

- Solidworks is an intuitive and engaging product, straightforward to learn. The adoption of Solidworks would not improve productivity immediately. However, Solidworks is quick to learn, improving efficiency in the long term.
- Modelling parts, assembly and producing drawings is an easy and engaging process. Creativity and engineering specifications are your only constraints to what you can create, exhibited through the creation of the catapult axial.
- Using VBA to update dimensions from an Excel subroutine is easy, as well as other Visual Basic for Application functions.
- The subroutine required to update dimensions and material properties fits seamlessly with the current version of *Sarge's Siege Catapult Workbook.*
- It is recommended our company adopt Solidworks to aid design.

5 References

CAD CAM Tutorial. (2016). Solidworks Tutorial | Sketch Wheel Rim in Solidworks. Retrieved from https://www.youtube.com/watch?v=_C8gy-OI1GU

Dassault Systems. (2017). SOLIDWORKS Community. Retrieved from http://www.solidworks.com/sw/community.htm

Maths is Fun. (2017). Perimeter of an Ellipse. Retrieved from https://www.mathsisfun.com/geometry/ellipse-perimeter.html

6 Appendices

6.1 Appendix 1: Wheel Rim Scalable Development

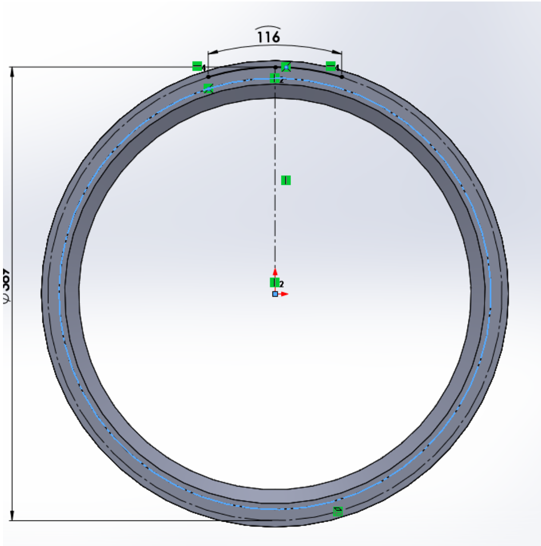


Figure 11: Initial Revolve

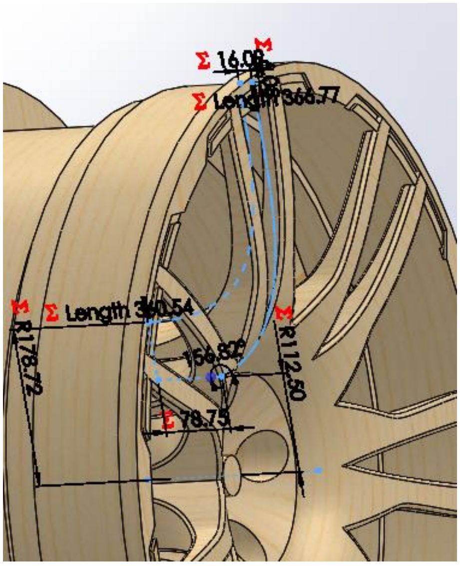


Figure 12: Sketch for Hub Cap

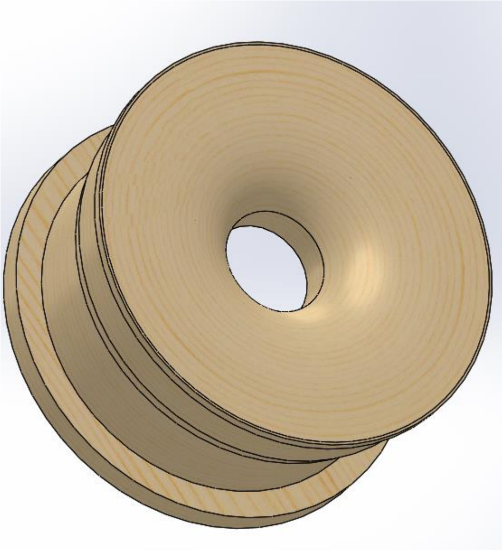


Figure 13: Revolve of Complex Hub Cap Sketch

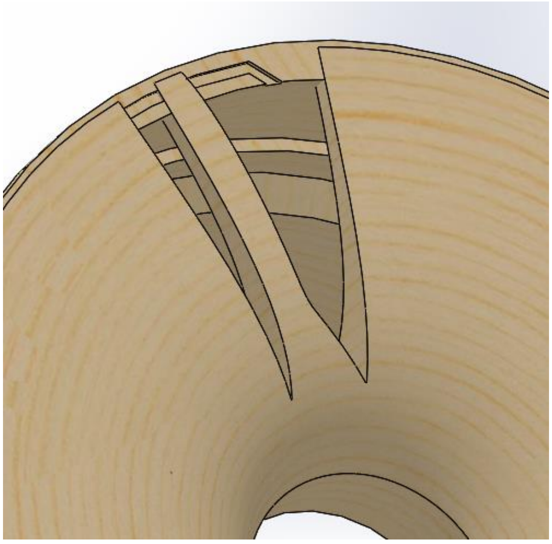


Figure 14: Wheel Rim Hub Cap Cut

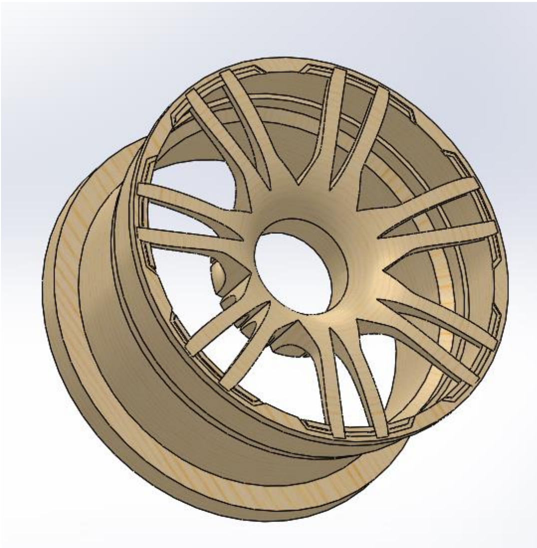


Figure 15: Using the Circular Pattern Tool on the Hub Cap Cut Extrude

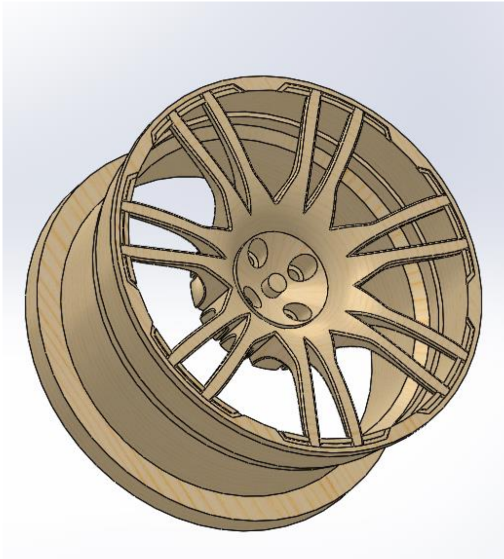


Figure 16: Finishing the Wheel Rim with Circular Extrudes

6.2 Appendix 2: Tire Development

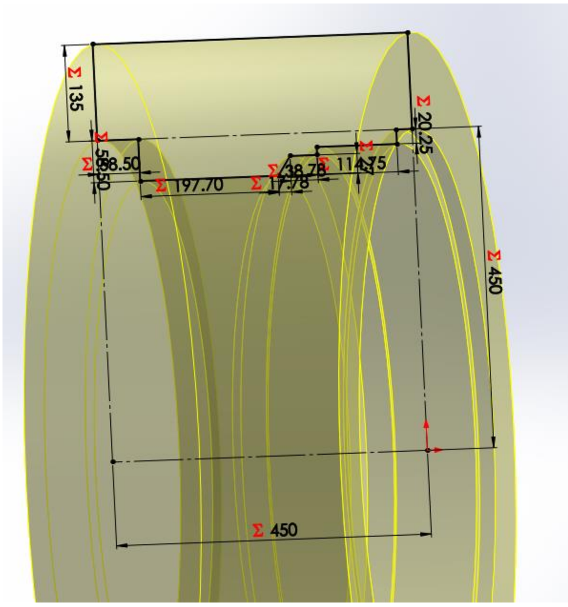


Figure 17: Sketch of Tire Thickness mirroring the Wheel Rim Thickness

Figure 18: Using the Revolve feature around a Circular Axis

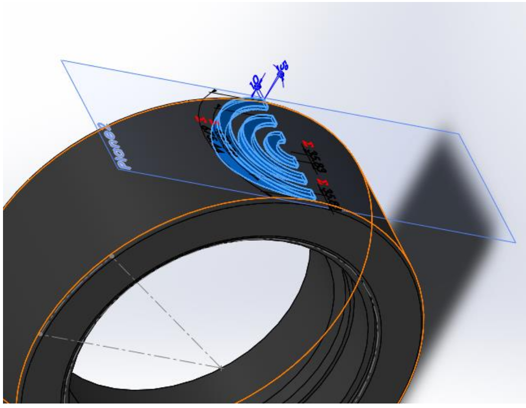


Figure 19: Creating the Tire Tread on a New Plane

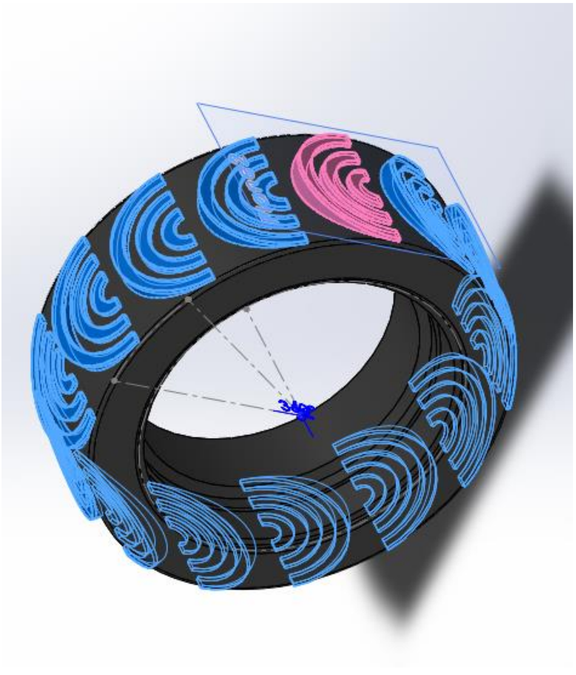


Figure 20: Using a Circular Pattern to repeat the Tire Pattern

6.3 Appendix 3: Drive Axial Development

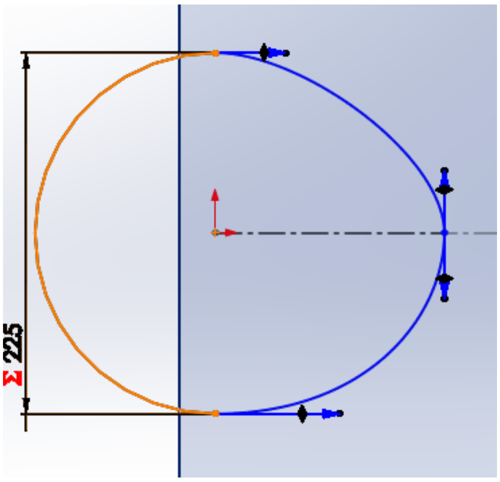


Figure 21: Sketch for Loft Mid-Section

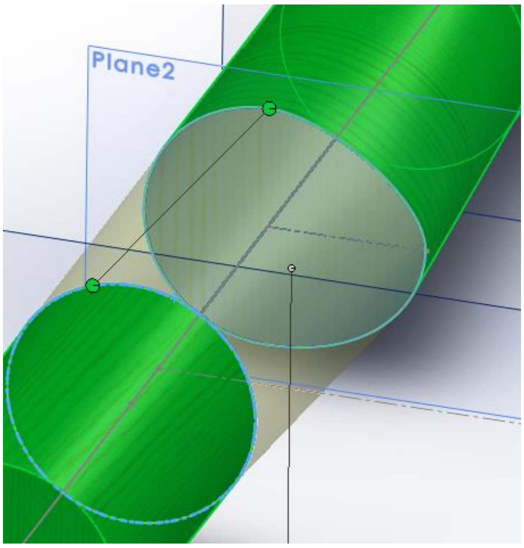


Figure 22: Using the Loft Feature to create an Extrude

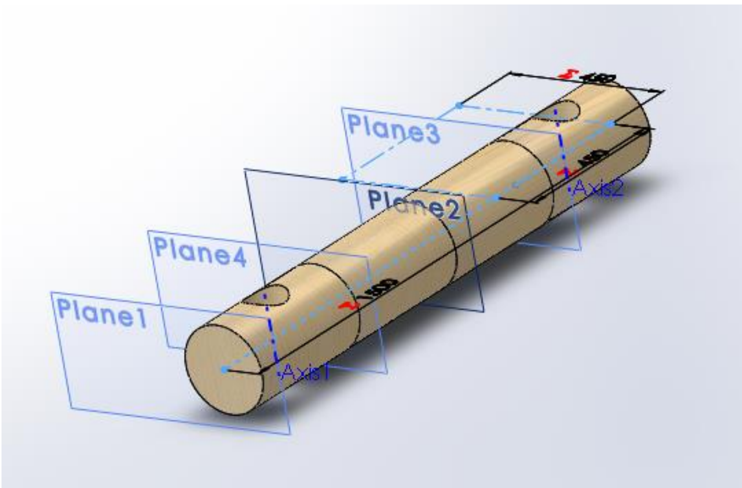


Figure 23: Creating Multiple Planes to create the Loft

6.4 Appendix 4: Wheel Fastener Development

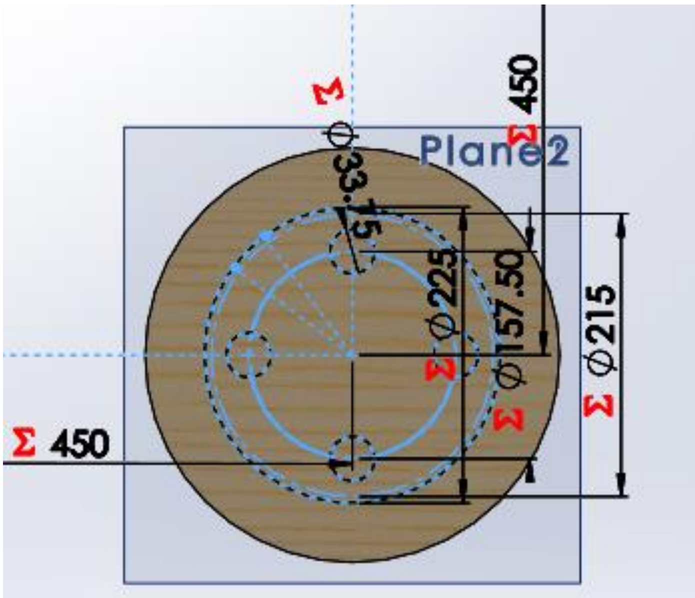


Figure 24: Sketch for first extrude with referenced dimensions

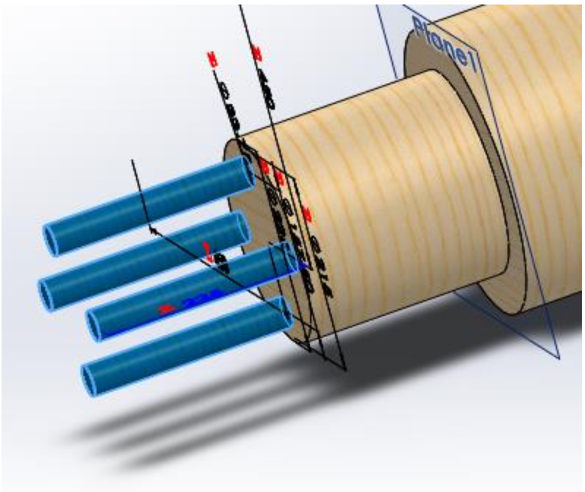


Figure 25: Four Circular extrudes to connect with the wheel

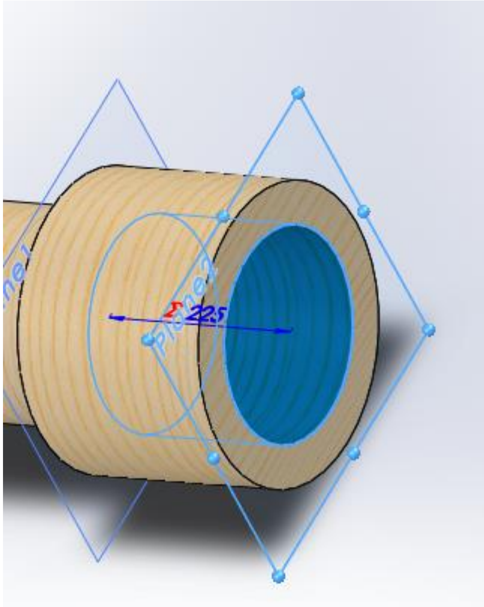


Figure 26: Hole features created using a circular cut extrude

6.5 Appendix 5: Swivel Mechanism Development

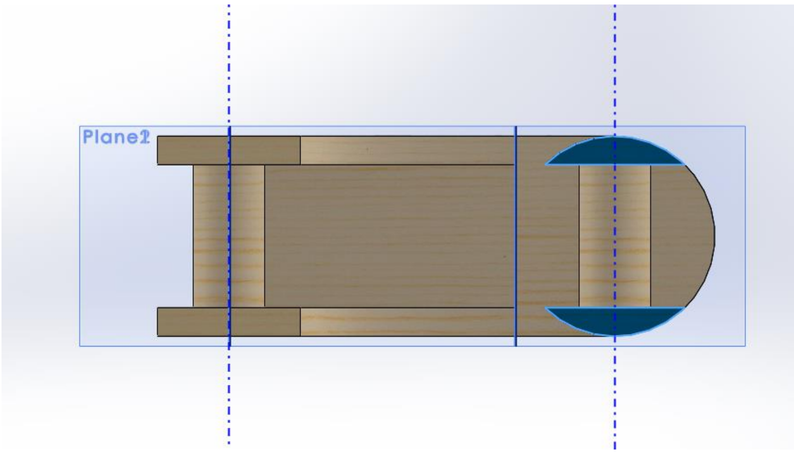


Figure 27: Front View of the First Circular Extrude

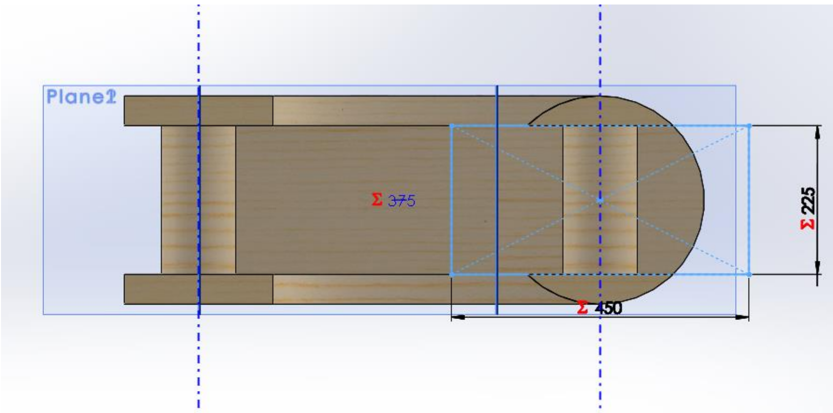


Figure 28: Using a Cut Extrude to hollow out the face

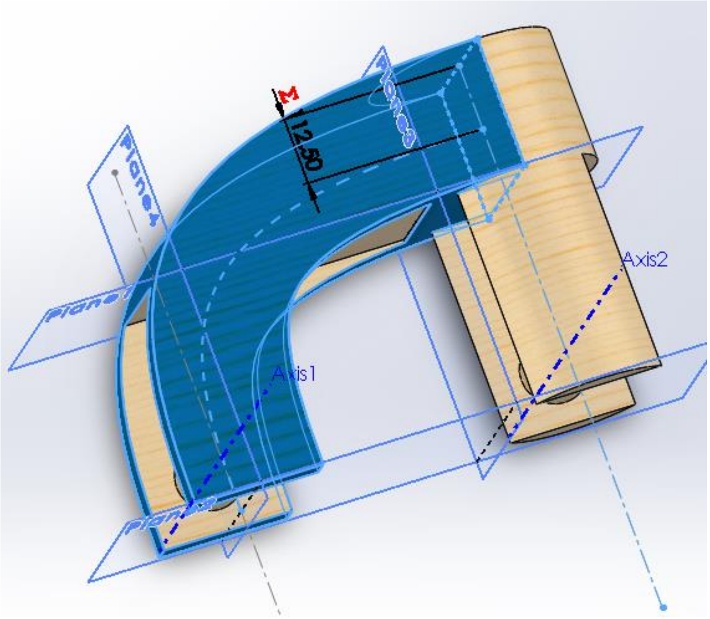


Figure 29:Using a Swept Blend to make a complex feature

6.6 Appendix 6: Completed Parts

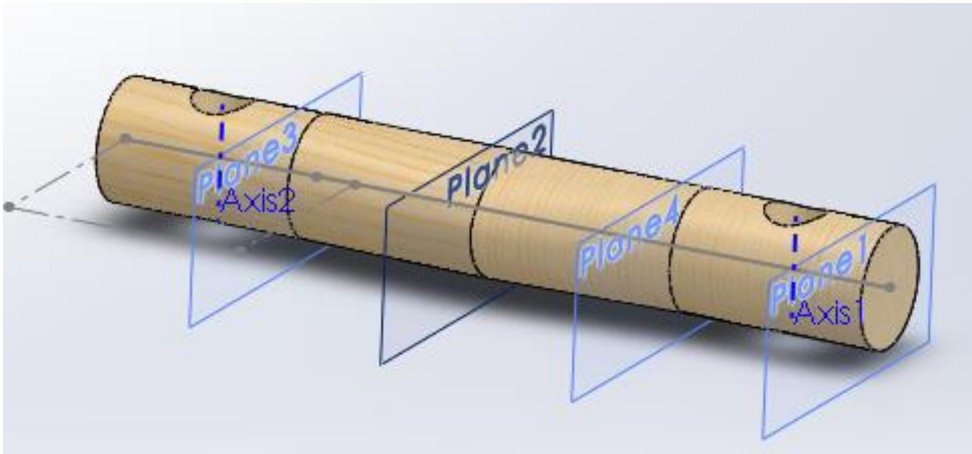


Figure 30: Drive Axial Final

Figure 31: Tire Final

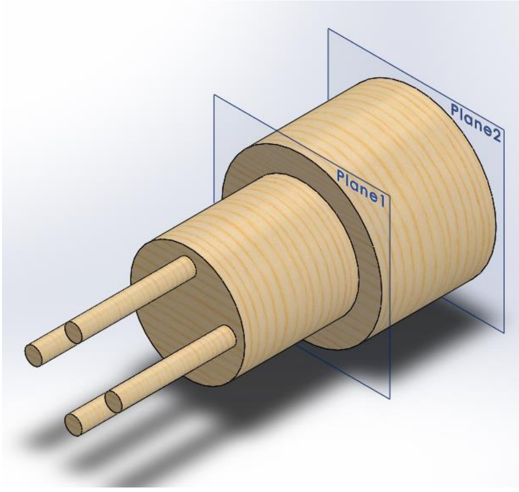


Figure 32: Wheel Fastener Final

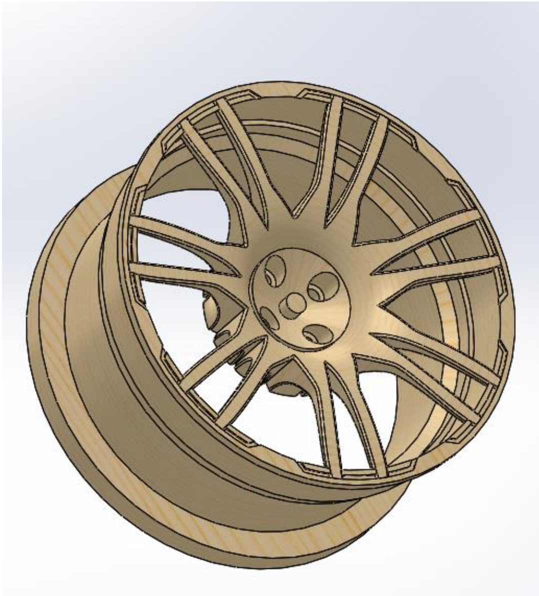


Figure 33: Wheel Rim Scalable Final

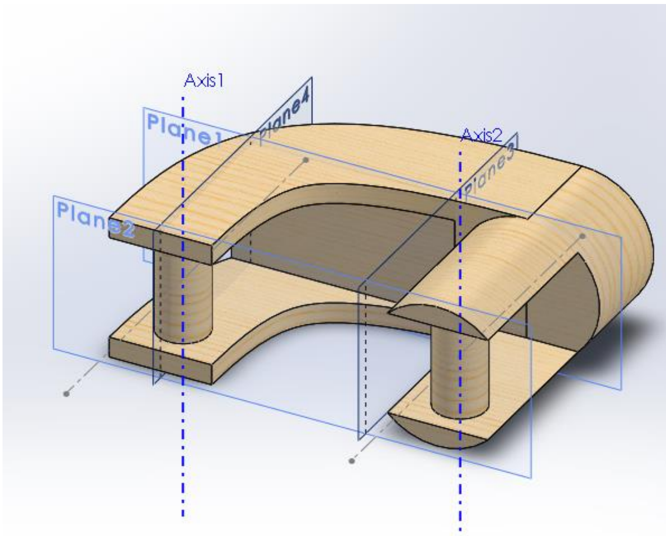


Figure 34: Swivel Mechanism Final

6.7 Appendix 7: Final Assembly

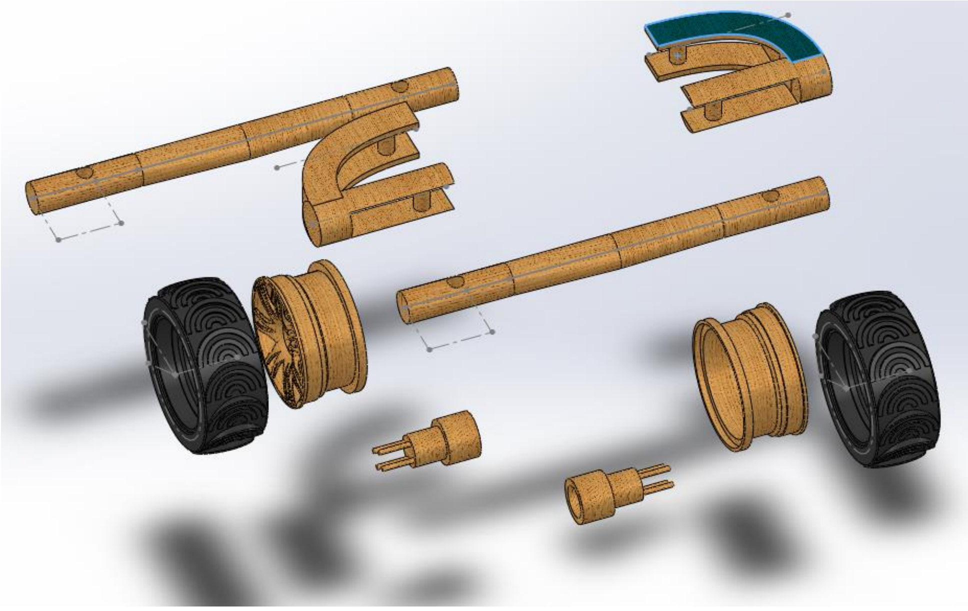


Figure 35: Pre- Assembly with Parts Individually Placed

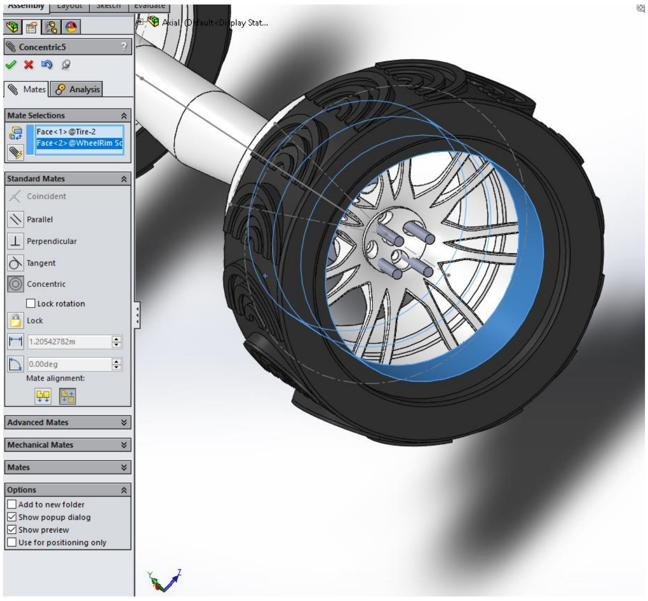


Figure 36: Using a concentric mate to place the wheel

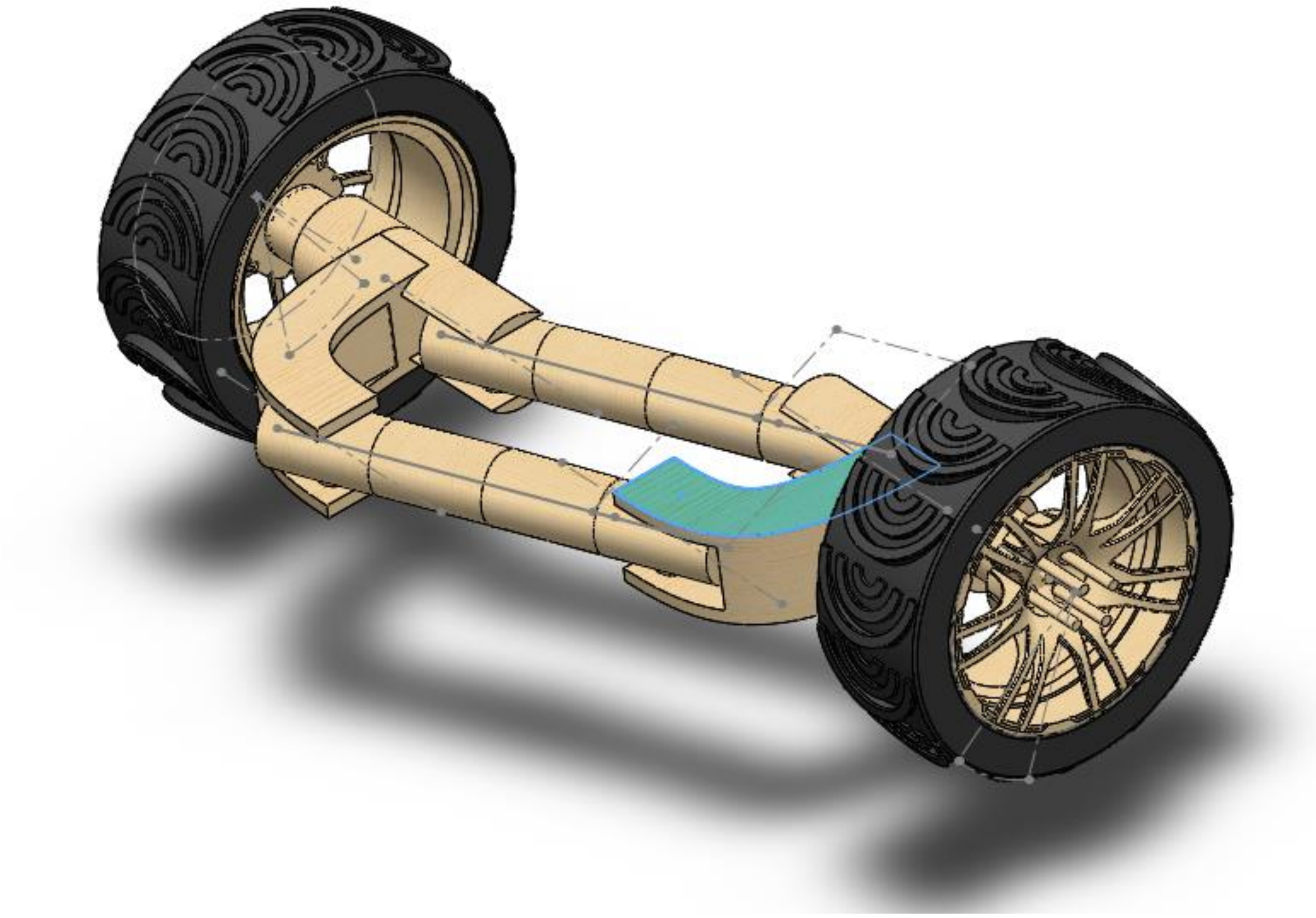


Figure 37: Final Assembly with spinning wheels and swivelling wheels

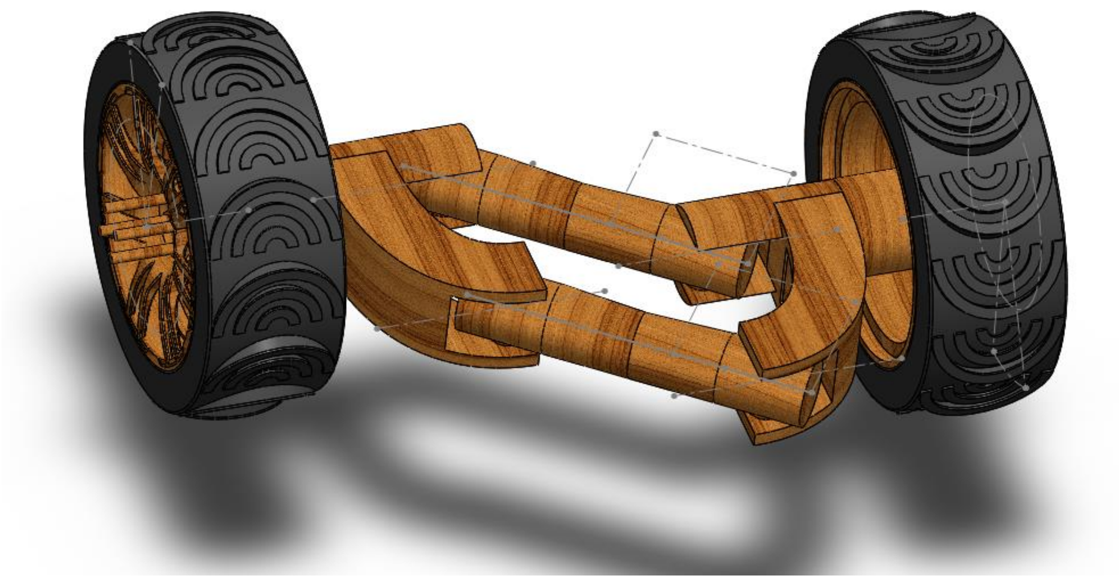


Figure 38: Axial Assembly with mahogany material, large wheel size and small catapult size

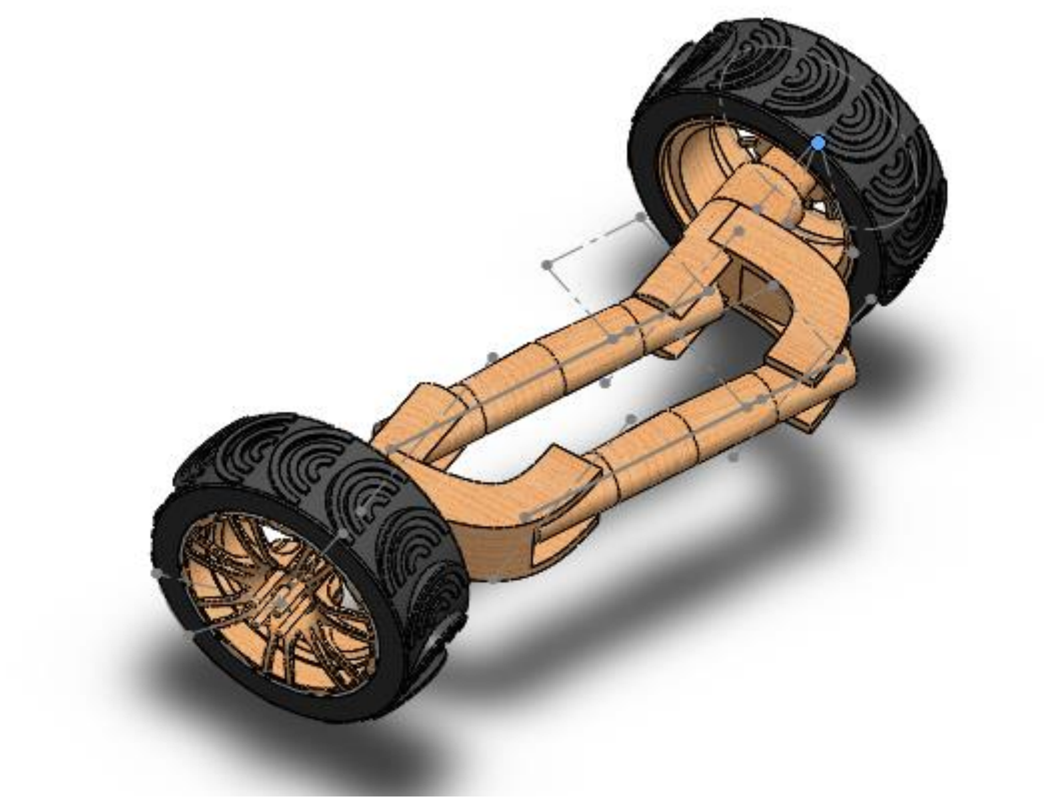


Figure 39: Axial Assembly with a cedar material

6.8 Appendix 8: Drawings

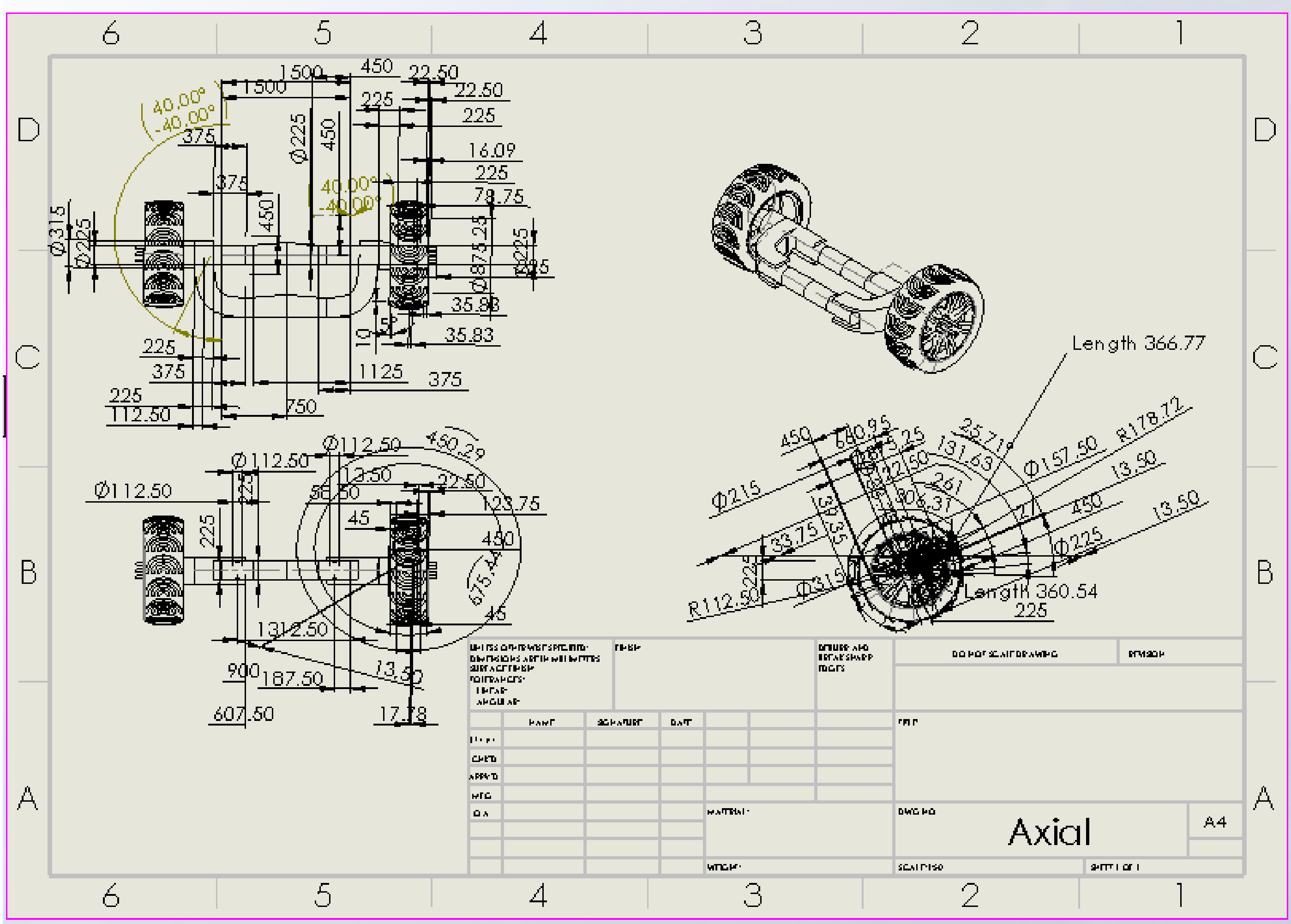


Figure 40: Axial Assembly Drawing

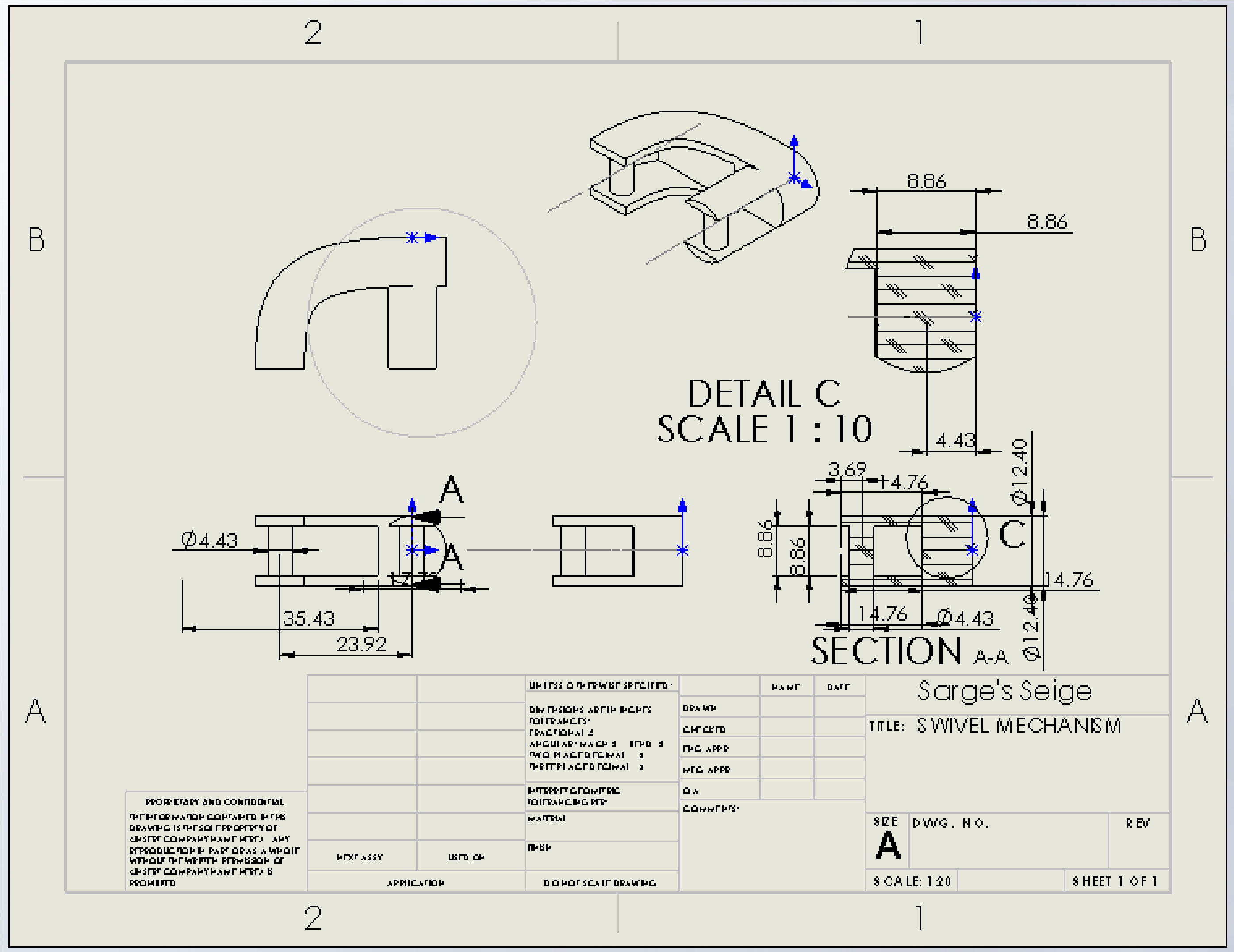
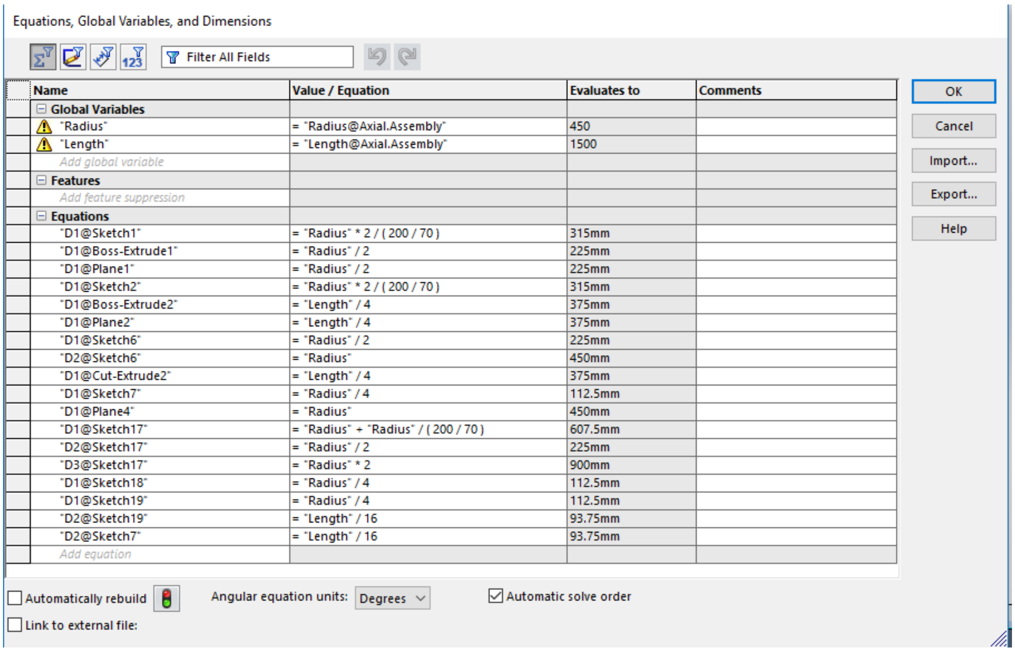


Figure 41: Swivel Mechanism Drawings with a top, front, right, isometric, segment and detailed views

6.9 Appendix 9: Equation Manager, Material Properties and Referenced Equations


Equations, Global Variables, and Dimensions

Name	Value / Equation	Evaluates to	Comments
Global Variables			
"Radius"	= "Radius@Axial.Assembly"	450	
"Length"	= "Length@Axial.Assembly"	1500	
Add global variable			
Features			
Add feature suppression			
Equations			
"D1@Sketch1"	= "Radius" * 2 / (200 / 70)	315mm	
"D1@Boss-Extrude1"	= "Radius" / 2	225mm	
"D1@Plane1"	= "Radius" / 2	225mm	
"D1@Sketch2"	= "Radius" * 2 / (200 / 70)	315mm	
"D1@Boss-Extrude2"	= "Length" / 4	375mm	
"D1@Plane2"	= "Length" / 4	375mm	
"D1@Sketch6"	= "Radius" / 2	225mm	
"D2@Sketch6"	= "Radius"	450mm	
"D1@Cut-Extrude2"	= "Length" / 4	375mm	
"D1@Sketch7"	= "Radius" / 4	112.5mm	
"D1@Plane4"	= "Radius"	450mm	
"D1@Sketch17"	= "Radius" + "Radius" / (200 / 70)	607.5mm	
"D2@Sketch17"	= "Radius" / 2	225mm	
"D3@Sketch17"	= "Radius" * 2	900mm	
"D1@Sketch18"	= "Radius" / 4	112.5mm	
"D1@Sketch19"	= "Radius" / 4	112.5mm	
"D2@Sketch19"	= "Length" / 16	93.75mm	
"D2@Sketch7"	= "Length" / 16	93.75mm	
Add equation			

Figure 42: Equation Manager with Reference Equations

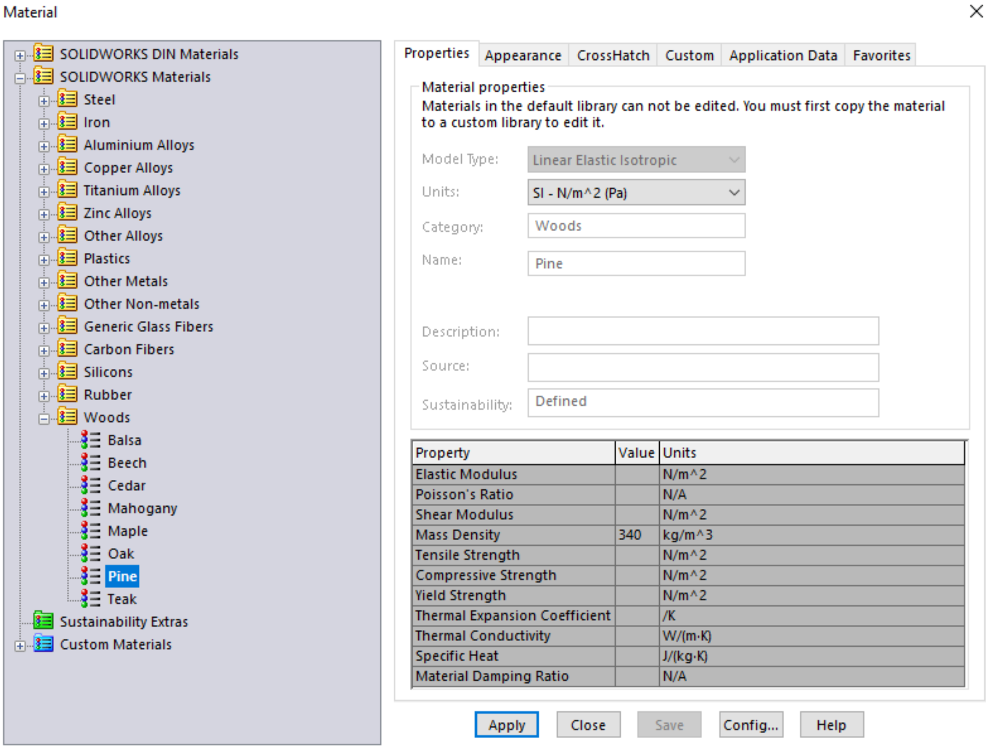


Figure 43:Material Property Manager

6.10 Appendix 10: Solidworks Manipulator Sub Routine

```
Option Explicit
Sub SolidworksManipulator()
    'Connor McDowall'
    'This sub changes the parameters of the Axial components based on the users input'
    'The drive axial, Wheel Fastener, Tire, WheelRim Scalable and Swivel Mechanism
    'Radius, Length and Material Parameters are Changed'

    'Dimension Variables'
    Dim swApp As Object 'Solidworks Application Pointer'
    Dim objFile As Object 'Solidworks Part or Assembly Pointer'
    Dim assembly As String 'Name of the Assembly you are changing'
    Dim part As String 'Name of the part changing'
    Dim response As Integer 'Response from message box'
    Dim swEqnMgr As Object 'Equation Manager for part or assembly'
    Dim i As Integer 'Counting Variable'
    Dim strSplit1 As Variant 'Splitting Variable '
    Dim strSplit2 As Variant 'Splitting Variable '
    Dim material As Variant 'Material variable'
    Dim length As String 'length string value'
    Dim radius As String 'Radius String Value'

    'Make sure the user does the right thing'
    response = MsgBox("Select yes if wish to modify model and have all relevant files open. Select no otherwise", vbYesNo)
    If response = vbNo Then
        Exit Sub
    End If

    'Set Assembly Name'
    assembly = "Axial.SLDASM"
    'Set pointer to the Solidworks Application'
    Set swApp = CreateObject("Sldworks.Application")
    'Make the assembly the active document model'
    swApp.ActivateDoc (assembly)
    'Set pointer to make changes to the assembly'
    Set objFile = swApp.ActivateDoc(assembly)
    'Set pointer to Equation manager tool'
    Set swEqnMgr = objFile.GetEquationMgr
    'Change the Radius and Length Global variables'

    'Convert Values in the excel spreadsheet into strings'
    length = CStr(Worksheets("ReferenceData").Range("SolidworksWheelSize"))
    radius = CStr(Worksheets("ReferenceData").Range("SolidworksSize"))
    'Split the global variable storage equation strings at the equals sign'
    strSplit1 = Split(swEqnMgr.Equation(0), "=")
    strSplit2 = Split(swEqnMgr.Equation(1), "=")
    'Append the length and radius value onto the end of the strings'
    swEqnMgr.Equation(0) = strSplit1(0) & "= " & length
    swEqnMgr.Equation(1) = strSplit2(0) & "= " & radius
```

```
    'Change the material of each wooden part
    'Drive Axial'
    part = "Drive Axial.SLDPRT"
    'Make the part the active document model'
    swApp.ActivateDoc (part)
    'Set the pointer to the object'
    Set objFile = swApp.ActivateDoc(part)
    'Change the material'
    material = objFile.SetMaterialPropertyName("SOLIDWORKS Materials.sldmat", Worksheets("ReferenceData").Range("SolidworksMaterial"))

    'WheelRim Scalable'
    part = "WheelRim Scalable.SLDPRT"
    'Make the part the active document model'
    swApp.ActivateDoc (part)
    'Set the pointer to the object'
    Set objFile = swApp.ActivateDoc(part)
    'Change the material'
    material = objFile.SetMaterialPropertyName("SOLIDWORKS Materials.sldmat", Worksheets("ReferenceData").Range("SolidworksMaterial"))

    'Swivel Mechanism'
    part = "Swivel Mechanism.SLDPRT"
    'Make the part the active document model'
    swApp.ActivateDoc (part)
    'Set the pointer to the object'
    Set objFile = swApp.ActivateDoc(part)
    'Change the material'
    material = objFile.SetMaterialPropertyName("SOLIDWORKS Materials.sldmat", Worksheets("ReferenceData").Range("SolidworksMaterial"))

    'Wheel Fastener'
    part = "Wheel Fastener.SLDPRT"
    'Make the part the active document model'
    swApp.ActivateDoc (part)
    'Set the pointer to the object'
    Set objFile = swApp.ActivateDoc(part)
    'Change the material'
    material = objFile.SetMaterialPropertyName("SOLIDWORKS Materials.sldmat", Worksheets("ReferenceData").Range("SolidworksMaterial"))

    'Rebuild the assembly'
    Set objFile = swApp.ActivateDoc(assembly)
    objFile.EditRebuild
End Sub
```

Figure 44: SolidworksManipulator Sub Routine

6.11 Appendix 11: Additional Tools, Tutorials and Features

Figure 45: Using Different Enviroments as Backgrounds

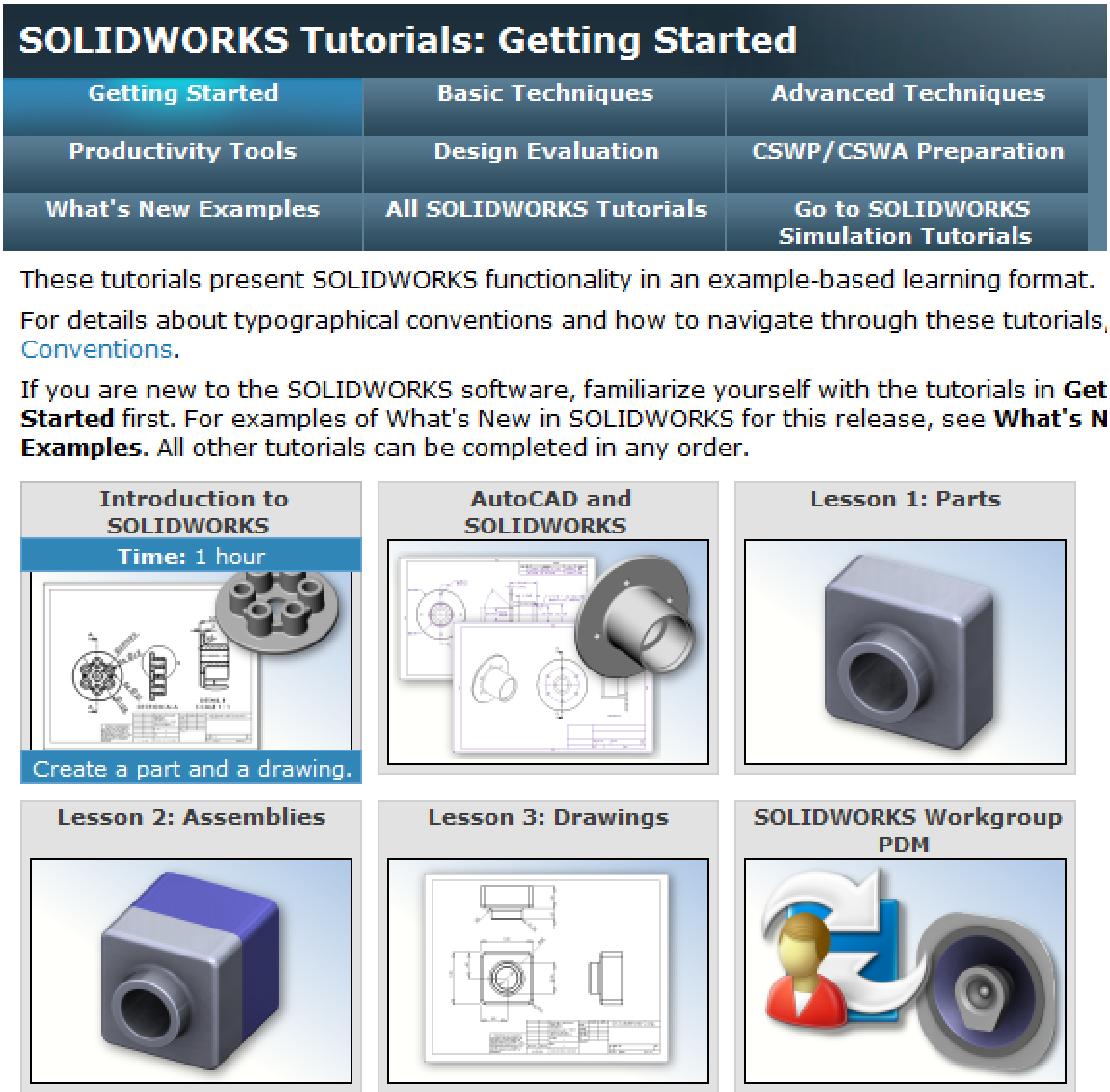
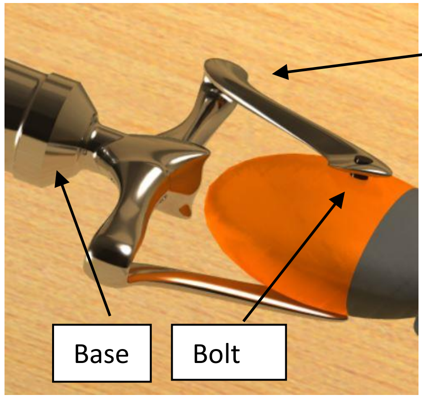
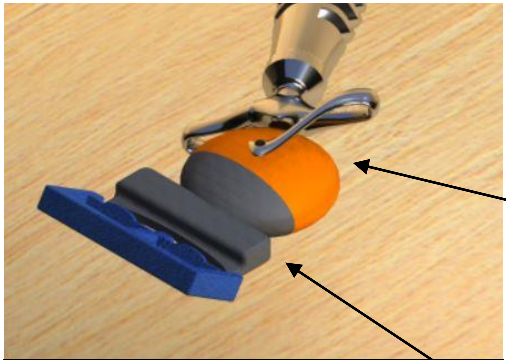


Figure 46: Solidworks Tutorials

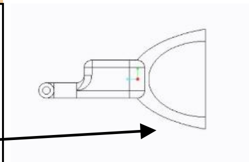


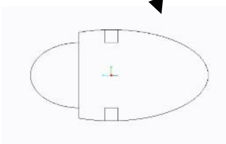
The S-Hold was made using two Sweep Blends with varying sections (a circle in the centre, points at the end and ellipses in the middle). The combination of sweep blends was warped to create the shape able to hold the swivel mechanism. Extrudes and rounds where then added to form the base of the hold. A cylindrical extrude was used to create the inserts for the bolts.



Two revolves and a removal extrude made the shape of the swivel mechanism. The initial revolve formed the flattened sphere. A cut was removed with another revolve added to create the revolve connector, similar to the product I am designing.

The shape on the Hinge Connector that connects to the Swivel Mechanism was made using the same process as the Swivel mechanism. Addition extrudes and revolves were added to complete the part.



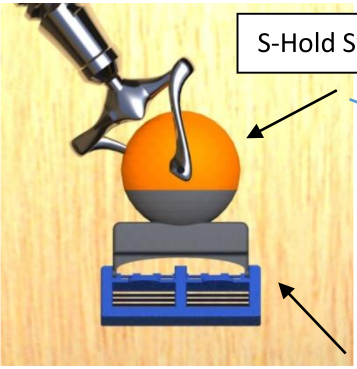


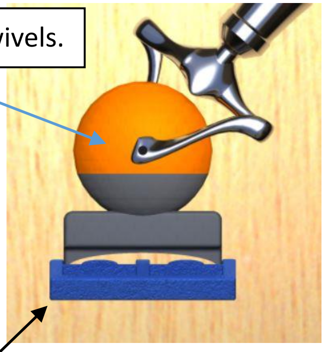
The Shaft is one of the most complex features. The initial structure was composed of a tapered cylindrical extrude. A helical sweep runs down the shaft. The cylindrical extrude was added to a plane tangential to the tapered extrude then wrapped the tapered surface. The aggregate part was warped to flatten and produce varying ridgelines to serve as the grip. Apart of the grip is a warped power symbol. Theoretically, the power button uses touch sensitive technology, similar to a touch lamp, to activate.

Frame with five blade attachments. Frame created via simple extrudes, geometric patterning and rounds. Multiple Datum planes used. Blade produced via sweep blends. Placed on frame via geometric patterning. Rendering uses bump mapping.

Isometric view of complete assembly. I used Ash wood as my backdrop as resembled my workspace.

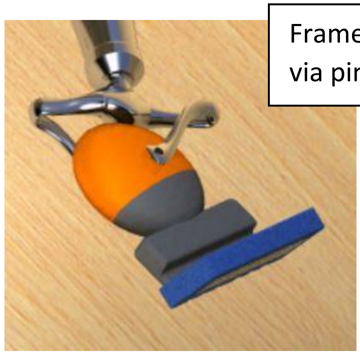
S-Hold Swivels.

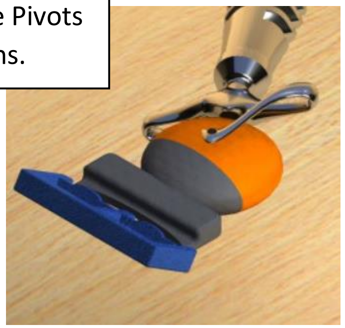




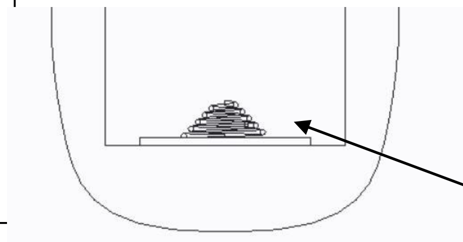
Right side of the complete assembly. Note: The constraints state the major parts dimensions may not be less than 1mm. Minor parts are not included within this constraint.

Frame Pivots via pins.

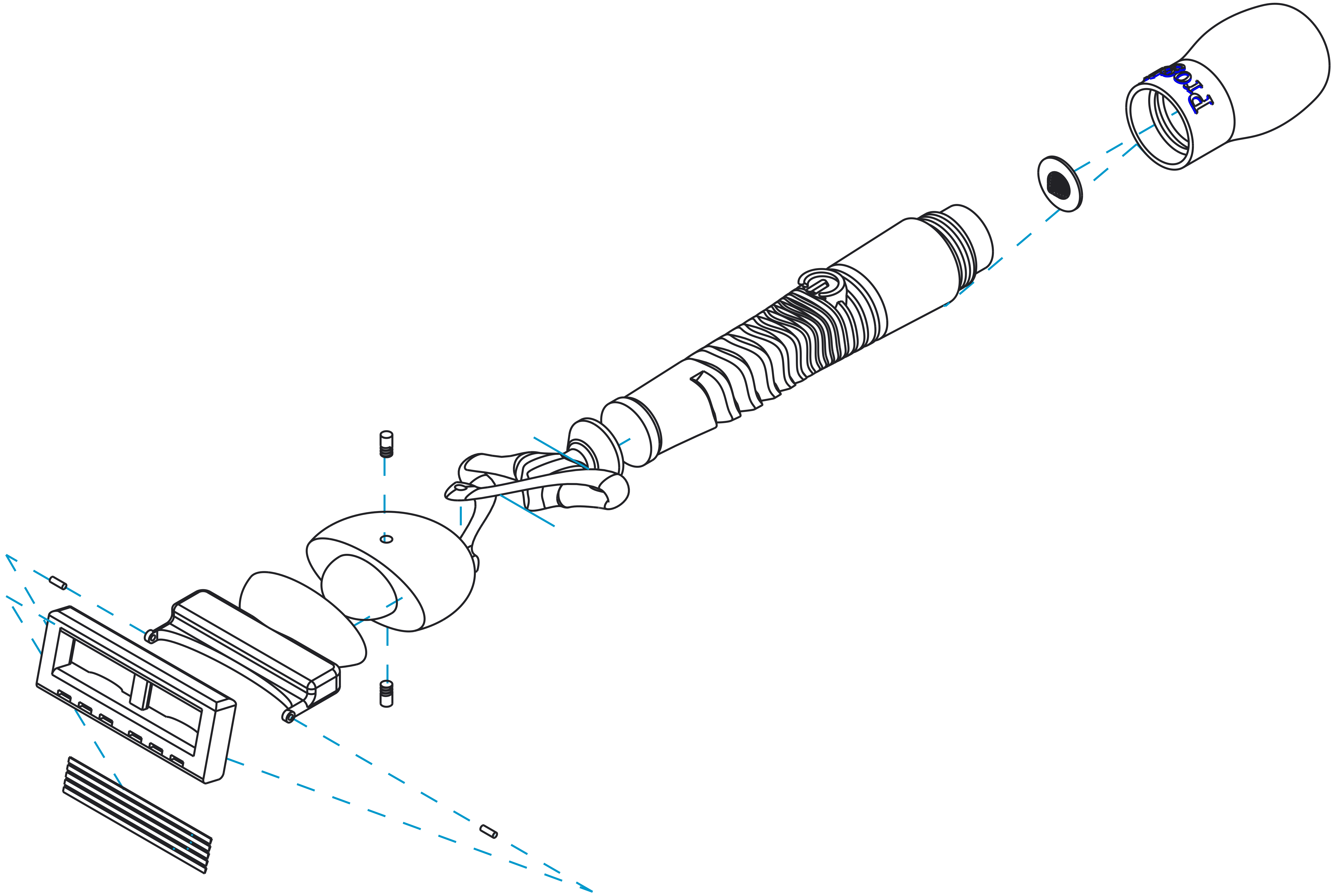




Two moving parts. The frame is connected to the Hinge connector via a pin constraint. The hinge pivots 90 degrees. The S-Hold is connected to the Swivel Mechanism via a cylinder constraint. The S-Hold is set to pivot 180 degrees.



The Cap was made using a simple revolve feature The Embossed text is an extrude offset on a tangential plane then wrapped around the cylindrical surface. Within the cap is a minor spring, created via a cylindrical extrude and a spiral sweep. See on exploded isometric drawing. The spring has a pitch of 0.40mm and slant height of 2.83mm (3.s.f)



DRAWN BY	TITLE Gillette Fusion Proglide Razor	
DATE	DEPARTMENT	
FILENAME	FILE LOCATION	
TUTOR	QUANTITY	SCALE
GROUP No.		SHEET

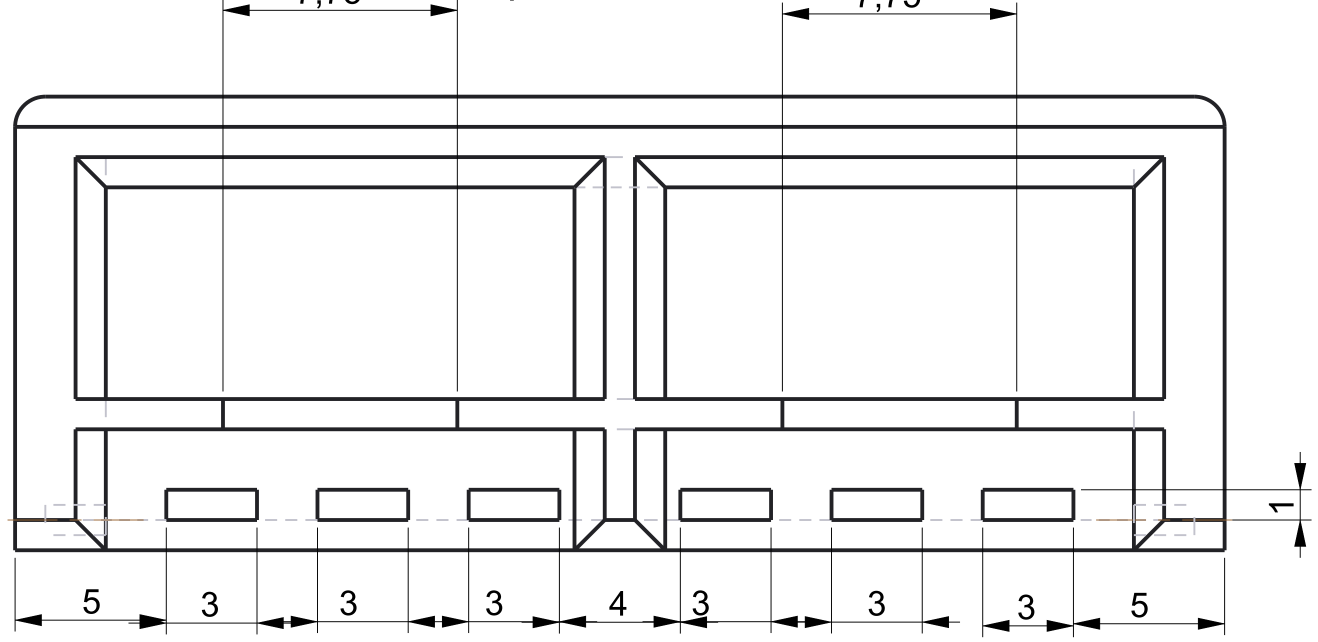
THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

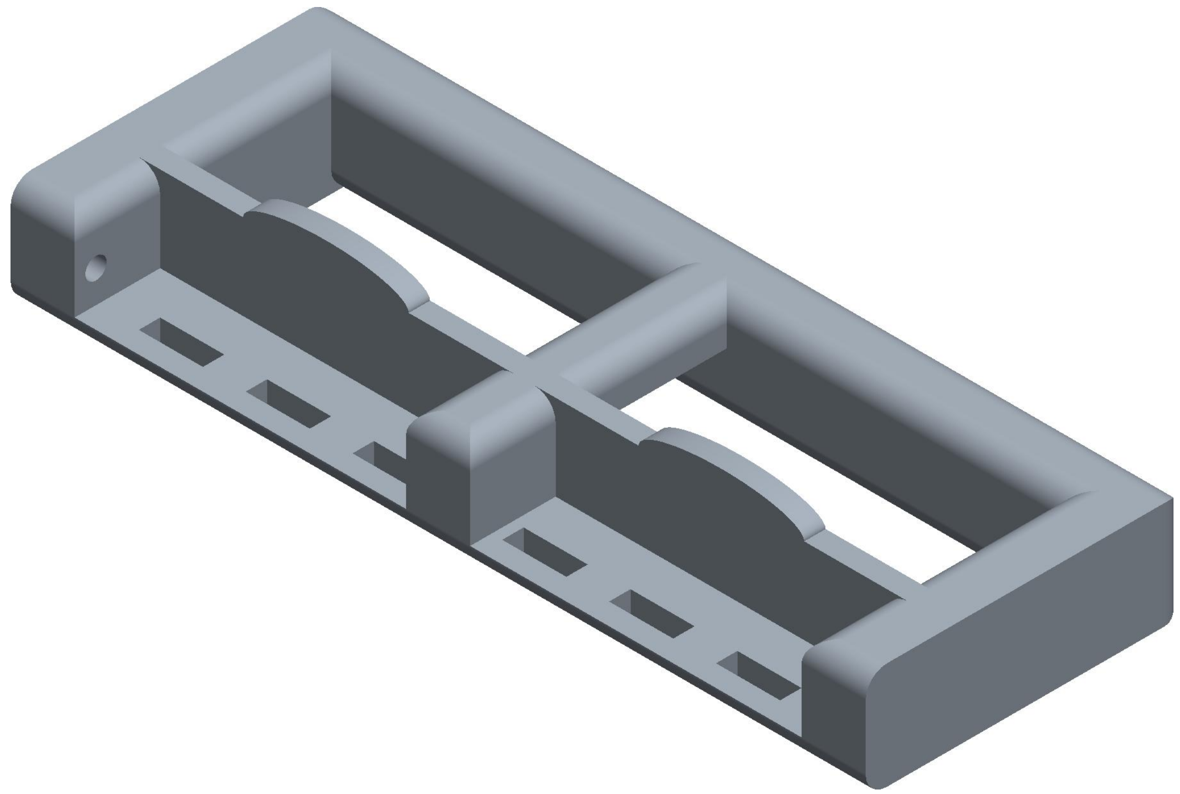
ALL TOLERANCES ARE ±0.1mm UNLESS OTHERWISE SPECIFIED

ALL DIMENSIONS IN MILLIMETRES

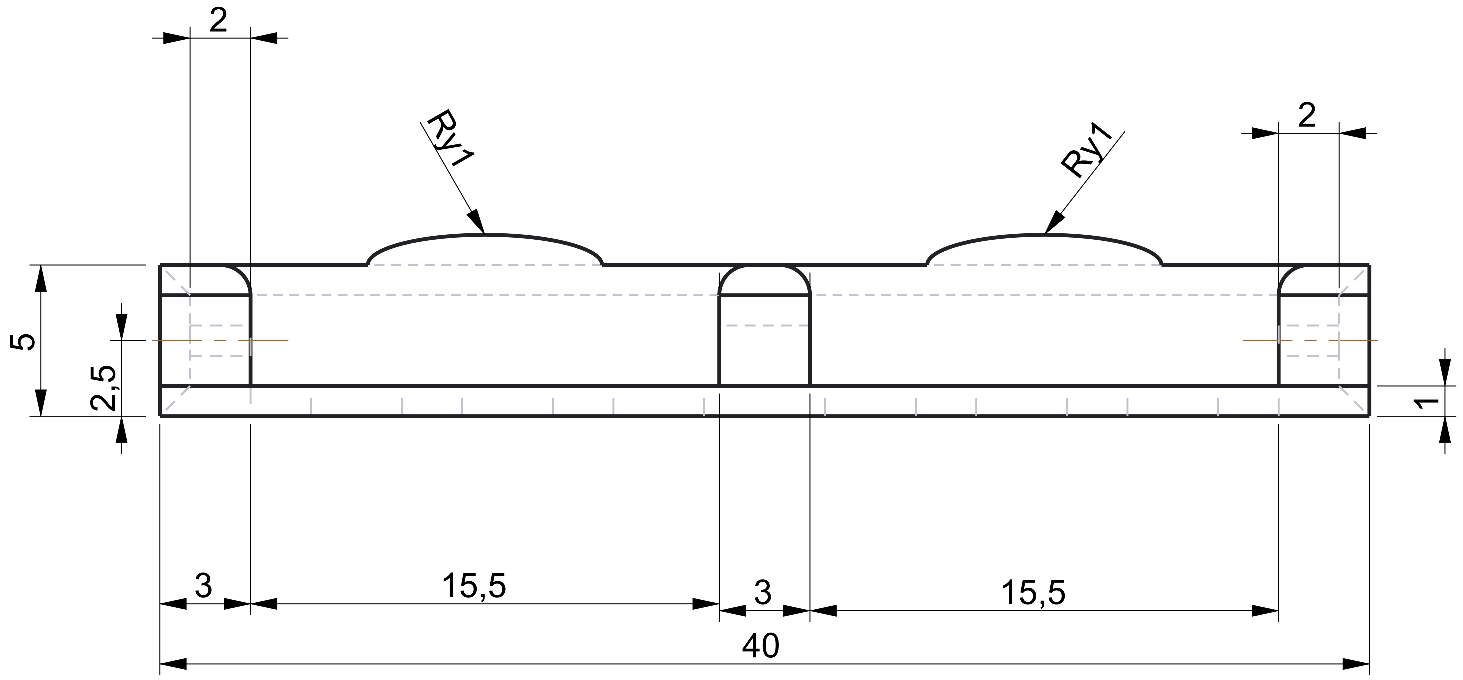
A3

Top View

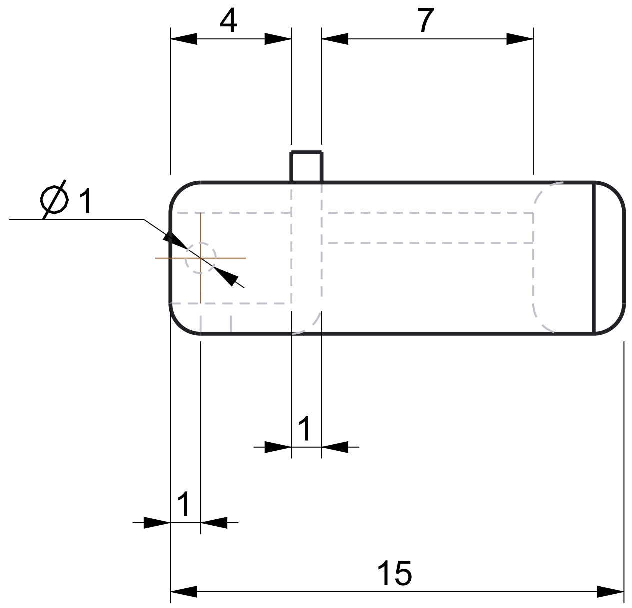




All Round dimensions are R1
unless otherwise stated.



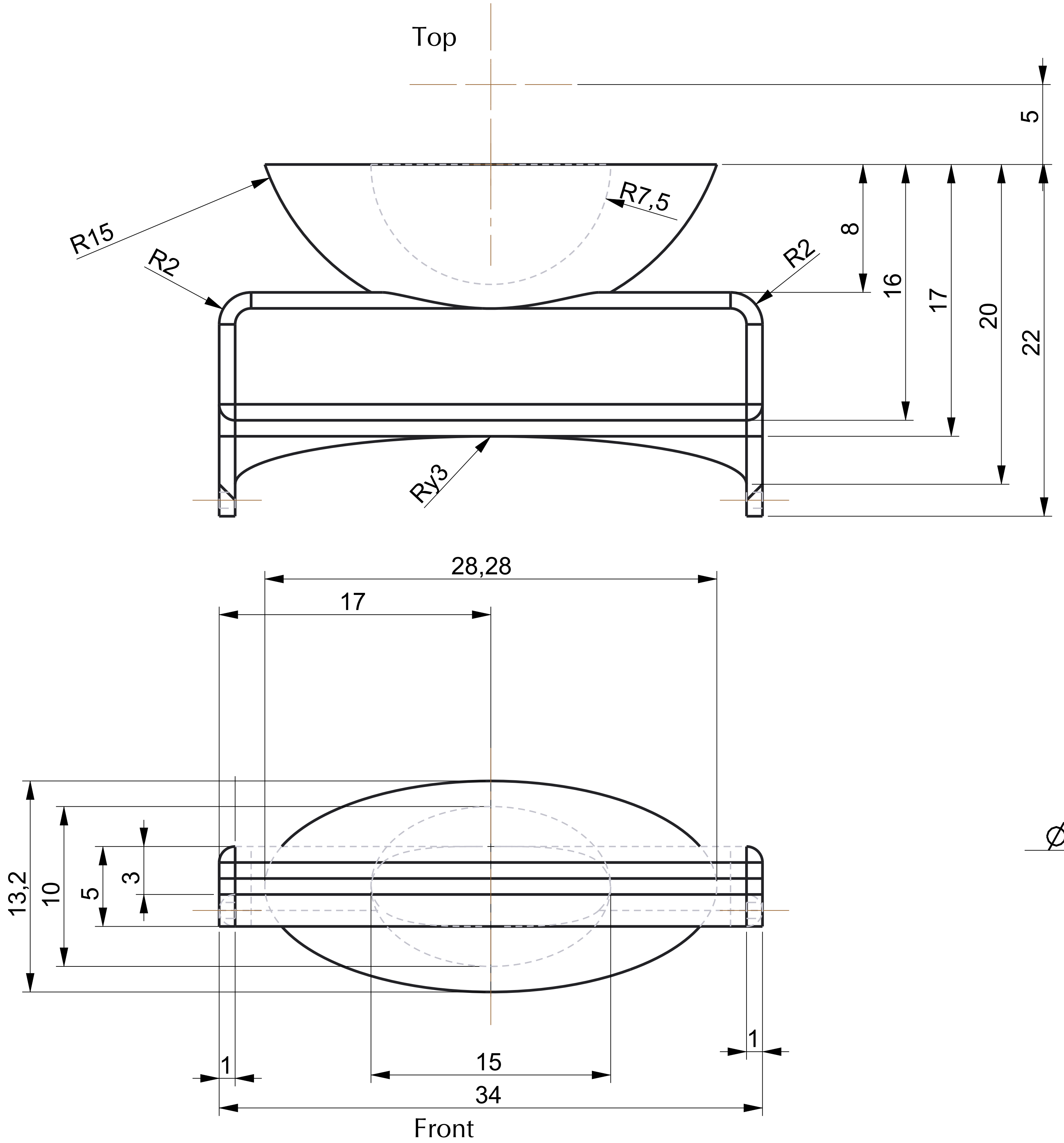
Front View

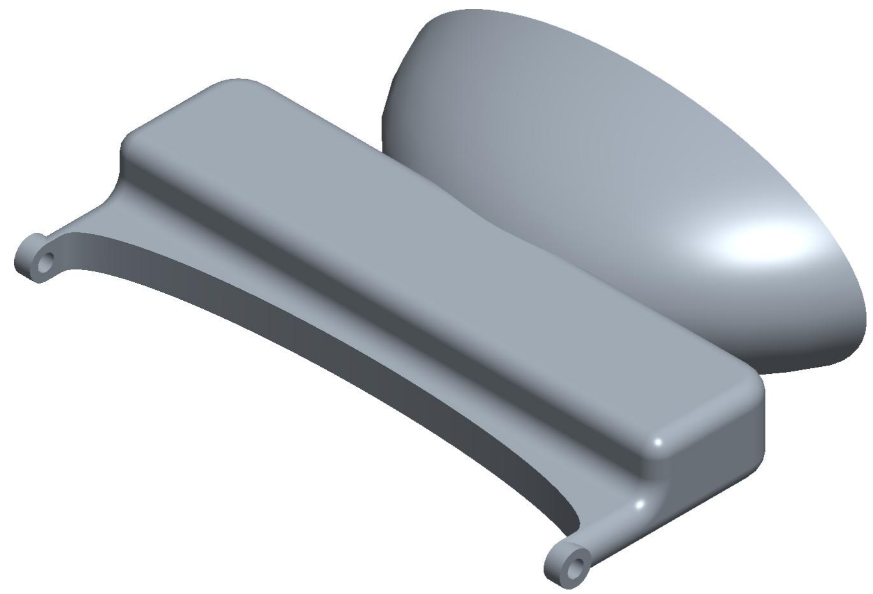


Right Side

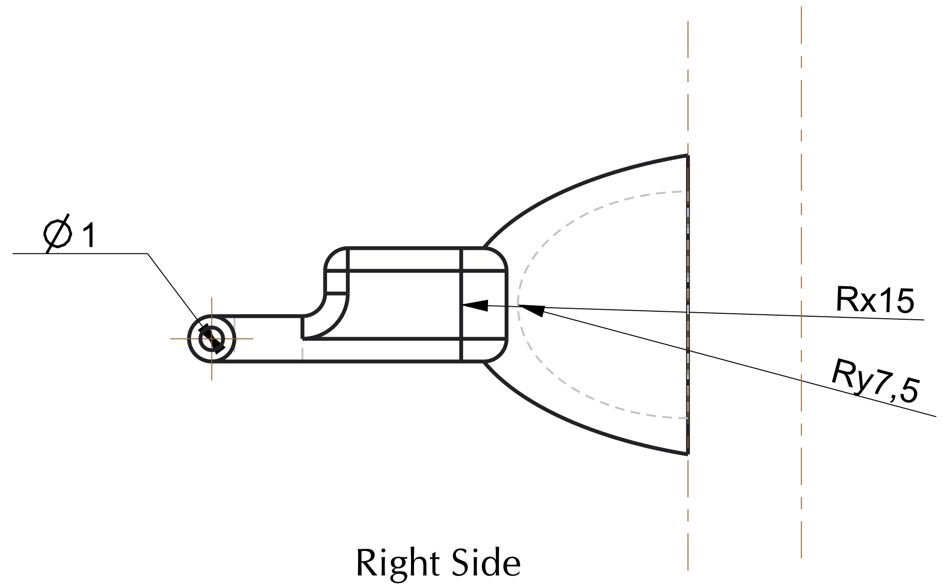
THE UNIVERSITY OF AUCKLAND NEW ZEALAND	DRAWN BY Connor McDowall	TITLE Frame			
	DATE 21/04/16	DEPARTMENT Engineering General			
	FILENAME	FILE LOCATION			
	TUTOR Rohann DSouza	MATERIAL PVC	QUANTITY 1	SCALE 4:1	SHEET 1 of 1	A3
	GROUP No.				

ALL TOLERANCES ARE ±0.1mm UNLESS OTHERWISE SPECIFIED | ALL DIMENSIONS IN MILLIMETRES

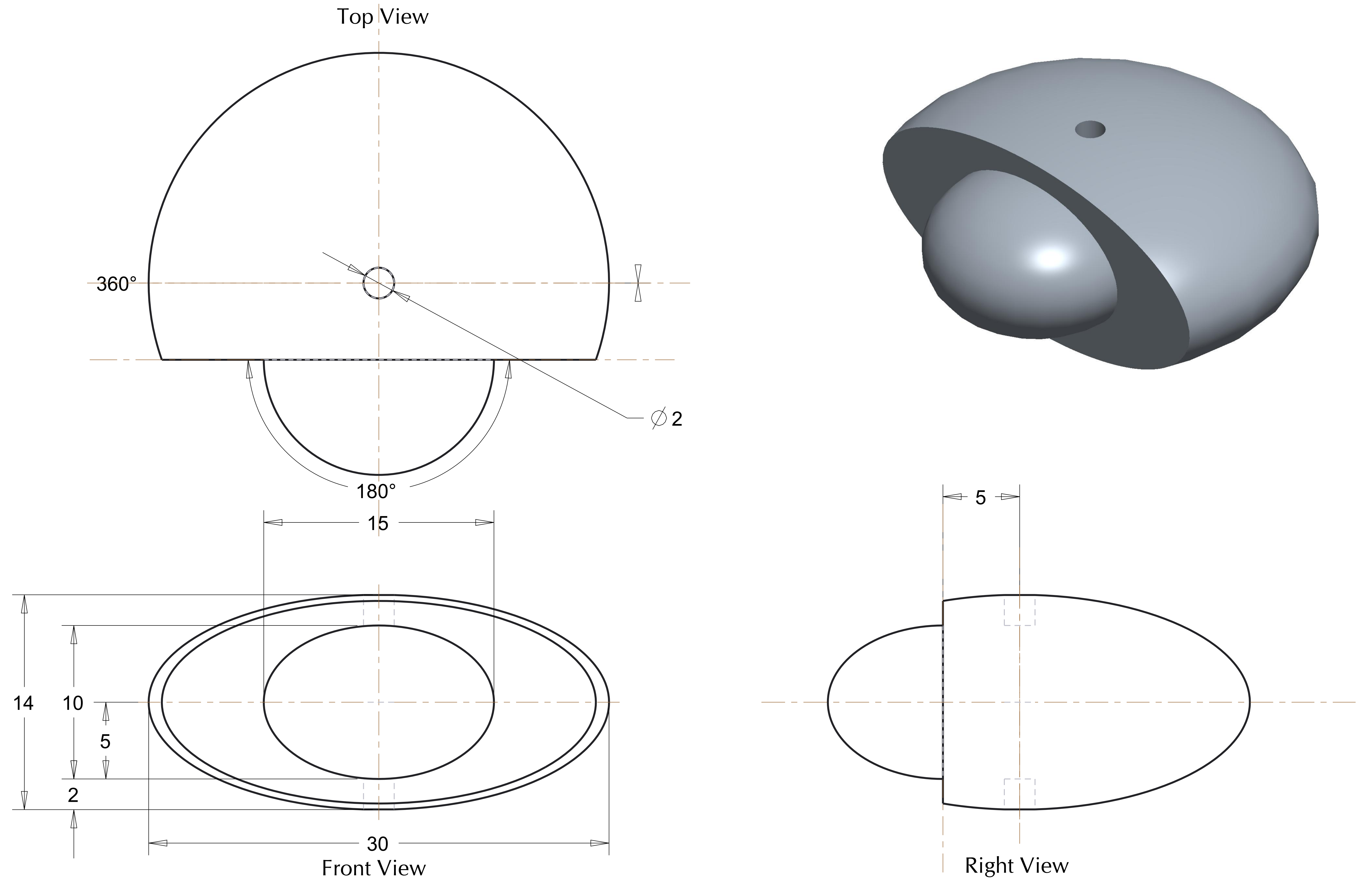




All Rounds dimensions are
R1 unless otherwise stated



	THE UNIVERSITY OF AUCKLAND NEW ZEALAND	DRAWN BY Connor McDowall	TITLE Hinge Connection				
		DATE 21/04/16	DEPARTMENT Engineering General				
		FILENAME	FILE LOCATION				
ALL TOLERANCES ARE ±0.1mm UNLESS OTHERWISE SPECIFIED	ALL DIMENSIONS IN MILLIMETRES	TUTOR Rohann DSouza	MATERIAL PVC	QUANTITY	SCALE 3:1	SHEET 1 of 1	A3
		GROUP No.					



DRAWN BY Connor McDowall	TITLE Swivel Mechanism					
DATE 21/04/16	DEPARTMENT Engineering General					
FILENAME	FILE LOCATION					
TUTOR Rohann DSouza	MATERIAL PVC	QUANTITY 1	SCALE 4:1	SHEET 1 of 1	A3	
GROUP No.						

THE UNIVERSITY OF AUCKLAND NEW ZEALAND

ALL TOLERANCES ARE ±0.1mm UNLESS OTHERWISE SPECIFIED | ALL DIMENSIONS IN MILLIMETRES

Issue: 1 Date: 19/04/2016	Product Design Specification For Gillette Fusion Proglide Razor	Reference: A0001

Relating Specifications:

Issuing Authority:

Contents:

Foreword: Men have needed to Shave for thousands of years. The Gillette Fusion Proglide Razor will be the best to date.

Introduction: Design a Premium Razor that delivers an excellent shave and has longevity.

Scope: All Adolescents and Adults from the ages 12 to 80 must be able to use the razor.

Definitions

Functional Requirements:

- Provide a clean shave to the user
- Adjust to the changing gradients and curves of the human neck, face and chin.
- The razor must have rigidity and return to a default position after use.
- The razor must adapt to different face shapes.
- Use Gillette's Vibration aided shaving functionality.
- The razor must have blades within a detachable frame. The razor must use replacement frames, following Gillette business plan and razor blade model business strategy.
- The blade must be of high quality product to meet the quality Proctor and Gamble demand.
- The product must operate in temperatures between -20˚C and 60 ˚C.
- The product must operate in water temperature between 0˚C and 100 ˚C. Higher temperatures may cause plastic materials to soften/melt.
- Humidity exposed to will range between 0% to 100%.
- Must be capable of full submersion.
- Must resist and be unaffected by shaving creams, soaps, detergents or any other personal hygiene product.
- Target Population is for adolescents to adults aged 12 to 80. The razor must be able to function for the 95% confidence interval of this population. Varying hand, face, chin and neck sizes will need to be accounted for.
- Function for both men and women of different cultures, worldwide. Will be marketed to men.
- Razors appearance must be aesthetically pleasing, in line with Gillette's strategy of a premium razor.
- Total assembly length must be in the range 100mm – 180mm.
- Must have a minimum dimension (part) no less than 1mm and an overall assembly dimension no less than 8mm
- Overall weight must be less than 200g
- Frame must be able to move in a way that adjusts to changing gradients
- The razor will be fully assembled before distribution
- Replacement of Frame must take less than 10 seconds for someone with average co-ordination
- The Swivel Mechanism, Bolt, Hinge connector and Frame must be made out of a polymer
- The Cap, Shaft, S-Hold, pins and Spring must be made out of a metallic alloy
- Blades must be of Gillette's patented design
- Any Plastic and Metallic alloy as long as other specifications met

Other Additional Requirements:

- No Sharp edges apart from the blades to prevent accidental lacerations
- Must have non-toxic materials
- Any paints, solvents, dyes used must not bleach
- Must have a satisfactory grip so the razor does not slip.
- Must be comfortable to hold
- Electric compartment facilitating vibration must be watertight or waterproof
- Materials must be prone to erosion/corrosion
- Must have a life time of 910 hours with 90% efficiency
- Must be drop resistant, withstanding drops from a maximum of 2m
- Must have sufficient room to facilitate shaving residue processing and removal.