

Learning Projectile Motion Through Data Visualization and Gaming Interface

Vaibhav Walvaker
The Information School
University of Washington
Seattle, Washington

Srijan Bhushan
The Information School
University of Washington
Seattle, Washington

Anurag Banerjee
The Information School
University of Washington
Seattle, Washington

Abstract—Projectile motion is one of the most fundamental and important concept of physics. It is a very important phenomenon on the basis of which other domains of physics are established. Projectile Motion is basically the motion of a launched object where the only force acting on the object is the gravitational pull. Projectile motion is conventionally taught and learnt through static text and images. This methodology is not very effective or engaging in explaining the concept. To solve this problem we have aimed at using advanced data visualization, simulation and gaming techniques to explain the phenomenon to users. In our project where we have used advanced data visualization tools - JavaScript and its specific data visualization library D3.js for simulating and explaining the concept of projectile motion. We have also added an element of gaming so that the user can learn while enjoying the learning process. We have designed the application such that the user can see dynamic changes and understand the phenomenon with a practical prospective.

Keywords—Projectile; Motion; D3.js; Simulation; Gaming; Design;

I. INTRODUCTION

[1]According to Nelson, Sengupta, Angelo, Clark (2009) Gaming and simulation are very potentially effective ways of teaching and learning many science concepts. “Simulations provide leverage in terms of harnessing a user’s spatial learning and perceptual systems in ways that text and verbal interactions do not”. Simulations have the ability to be dynamically be changed, restarted and stopped whenever the user wants. This feature is very useful in the specific case of projectile motion. A good understanding of projectile motion requires the user to set the effecting values as per desire, restart the simulation as per desire and have a visually appealing interface. In order to accomplish the above we have used the programming language JavaScript and specifically used the D3.js library. D3.js is a powerful library which allows creating dynamic data visualizations. We have effectively utilized the programming language and included required features and created a visually appealing simulation. We have also included an element of gaming in the simulation. Adding a gaming element provides more user engagement and a sense of progress and achievement for the user. All this helps in better understanding of the concept than the conventional text and verbal learning methods.

A. Projectile Motion

Projectile motion is the phenomenon of curved motion of launched object due to various factors including the only

significant acting force – the gravitational force. [2]The motion of object in this phenomenon has two components, the vertical component and the horizontal component. The fundamental of definition establishes that gravity is the only force acting on the launched object. In the absence of gravitational force or the presence of any other force the phenomenon will not be a projectile. A good illustrative example of project motion is shooting a basketball toward the target basket. There are many other examples that we see in everyday life. The projectile motion phenomenon has parameters associated with it which contribute to the phenomenon. Following are the parameters:

- (1) Initial Velocity, V_0
- (2) Angle of Projection, θ
- (3) Horizontal Range, R
- (4) Time of Flight, T
- (5) Maximum Height Reached, H
- (6) Gravitational Force, g

All the above parameters are crucial parts of the projectile motion. We have included all the parameters while simulating and explaining the phenomenon visually.

B. Formulas Used

The visual simulation and game features required us to use the important formulas of projectile motion. These formulas are used to produce the output results which are then inferred by the user. In addition, they are also used to provide inference of effecting factors of the projectile motion. Following are the formulas used:

$$\text{Time of Flight, } T = 2 \times V_0 \times \sin \theta \quad (1)$$

$$\text{Maximum Height Reached, } H = (V_0^2 \times \sin^2\theta) / 2 \times g \quad (2)$$

$$\text{Horizontal Range, } R = (V_0^2 \times \sin 2\theta) / g \quad (3)$$

$$\text{Parabolic Path, } Y = (\tan\theta \times x) - g/2 \times (V_0^2 \times \cos^2\theta) \times x^2 \quad (4)$$

C. Tools Used: JavaScript and D3.js

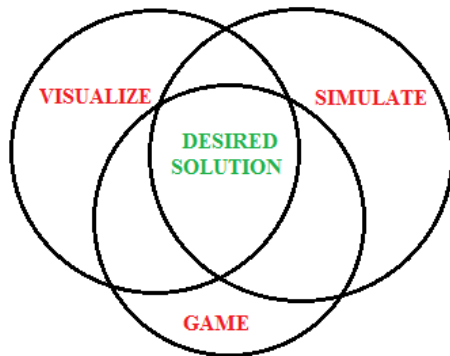
We have used the programming tool JavaScript and its specific powerful visualization library D3.js. JavaScript is the web based programming language. JavaScript is used extensively across the web to create websites with interactive features. D3.js stands for Data Driven Documents, is a JavaScript library used to produce graphs, charts and other visualizations. Both JavaScript and D3.js are widely supported by almost all browsers.

II. THE PROBLEM WITH UNDERSTANDING PROJECTILE MOTION

Projectile Motion might look like an easy concept to understand at first sight. However, understanding it completely is not as easy for everyone. Many students might face problem understanding the factors that are affecting the object, its motion and its path. The effect of each is hard to understand through conventional method of text. Specifically, it is hard to understand the combined effects of all the effecting factors. The methodology of learning through text or static images is that consumes a lot of time and does not give a practical approach. It does not give the ability to student to understand the how the phenomenon actually changes with changed effects. The method is very complex and overwhelming to understand quickly. A better approach is to lean through simulation and visualization. We have worked on a simulation in which student will learn through trial and error. This methodology of trial and error makes the student understand how and why the changes are happening. This gives a much more practical approach to learn the concept. It helps understand the effects better as the student is able to see the changes dynamically at one place in a short time span.

III. PROPOSED SOLUTION

As discussed earlier, we propose a solution which is built on simulation, visualization and gaming components. We aim at combining all three aspects into one application that can be used by student users to understand the concept. Firstly, we want to enable user to select all effecting inputs such as velocity. Our solution enables user to change effecting parameter inputs and see changes accordingly. Secondly, we want simulate and visualize the phenomenon effectively. Specifically, visualize the path formed by the projectile motion and also display the components while it forms the trajectory. Thirdly, we propose adding an element of game in the application. We propose adding an indication of accomplishment when the user successfully implements the phenomenon. In addition, we also propose at adding dynamic prompts guiding the user to understand the implementation better. All this combined together will build a very effective application to help users understand the concept better.



IV. RELATED WORK

Most of the work we found while searching for explanations of projectile motion were videos or long textual explanations of the principles which governs the motion. We also found a flash based game [3] which shows the path generated by a projectile

as well as other parameters related to projectile. However, it lacked control of important parameters such as gravitational force (g) and inference of velocity component at each stage of the trajectory. It also lacks guiding the user and helping the user to achieve the target implementation.

The physics classroom [4] has elaborate explanations about all of the formulas used as well as some basic images and gifs. It explains how and why an object launched at a particular velocity and angle follows a parabolic path. A user would have to read through the entire explanation to come up with an idea for a projectile. This consumes a lot of time and effort.

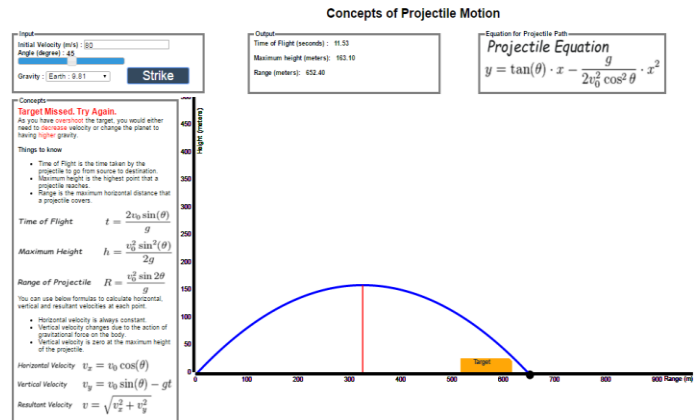
We aimed at solving all of the drawbacks we found during our research. In the following section we discuss our designed solution.

V. DESIGNED AND IMPLEMENTED SOLUTION

As discussed earlier, our solution was designed in JavaScript and D3.js. It is a web application. It requires a browser to run and execute. Most of the popular browsers like Chrome and Firefox run it. The application runs on a single page and does not require any navigation to other web pages.

A. Application User Interface and Experience

The designed solution has simple and lean user interface. We have designed the interface so as to not overwhelm the user with many things. We have included only the most important things to display. The major part of the screen is dedicated to the launching and tracing of the projectile's path. The trajectory is drawn in the 2-D plane. The launch of the object is from the left of the screen and it travels in the right direction. As part of the game, we have created a target in the game. This is a target image on the screen. It is placed along the horizontal axis at a random distance from the launch point. Following figure shows the application running.



The feature inputs to be selected are placed on the left top side of the screen. The outputs of the simulation are placed on the mid top of the screen. We have also included an image of formulas on the right top side of the screen to act as a reference for the user. The guiding text prompts have been placed on the top mid position. The axes of the plot contain actual values for vertical and horizontal distances. It is in meters. To give a better gaming element we have displayed a ball being launched

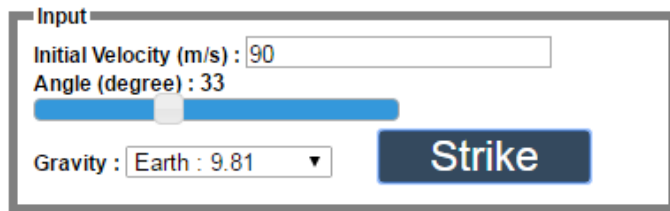
by the user. The text as displayed in simple font and color scheme.

We have used text box and sliders for input selection. The following features acts as inputs:

- (1) Velocity
- (2) Angle of Projection
- (3) The planet – gravitational force

These features have been selected as inputs because they are affecting the projectile trajectory motion. We have included the planet-gravitational pull as a design solution. The design solution to include the planet feature is based on the idea that this feature will be very interesting for users and engaging. It will be engaging because of the general outer space curiosity. This engaging design feature will lead to better understanding of the effect of gravitational pull force on the projectile motion. All these three input parameters are dynamically changed by user which in turn produces a new projectile path and motion. The user can at any time change the feature inputs or restart the application at any point of time.

The following figure shows the feature inputs selection interface and design.

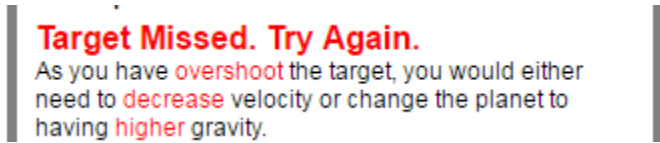


The input interface for the velocity is a simple text box in which the user will enter the value. For the angle of projection, we decided to implement a slider interface. The angle of projection generally has a fixed range which is more effectively selected through the decided interface. The planet feature has a drop down interface. This lets the user select one of the planets from the drop down menu. The menu options contain the planet name and the value of the gravitational pull.

The text prompts placed on the middle left of the screen just below the selection inputs. They act as guiding components for the user in the application. We have included three textual prompts which are displayed depending upon what the user executes. Following are the three prompts:

- (1) If the user enters excessive speed: “Target Missed. Try Again. As you have overshoot the target, you would either need to decrease velocity or change the planet to having higher gravity.”
- (2) If the user enters insufficient speed: “Target Missed. Try Again. As you have undershot the target, you would either need to increase velocity or change the planet to having lower gravity.”
- (3) If the user hits the target: “Target Hit. Well Done.”

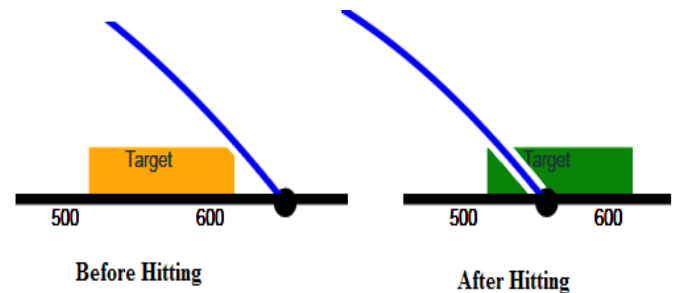
The figure below showcases the text prompts that appear in the application. This particular figure shows an example where user is exceeding the velocity and over shooting the target.



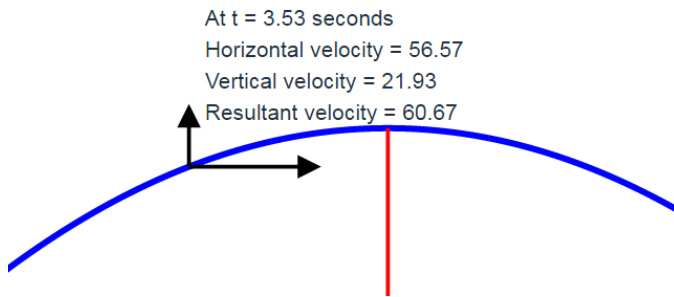
The above prompts help the user change the input accordingly. The user thus understands how changing values are affecting the projectile’s path. The path visually changes and is redrawn. This provides effective understanding of the effects on input variables like velocity.

Further discussing the above, the third prompt which displays the text has a certain design behind it. We wanted the user to realize a sense of accomplishment after hitting the target successfully. After many trial and errors, the user would understand the logic behind increasing/decreasing the angle/speed and how it is affecting the trajectory. Eventually when he hits the target, this displayed text would make it an enjoyable experience and not just a reading.

In addition to the displaying text when the target hits, we have also decided to change the color of the target image placed on the bottom of the screen. The color changes from yellow to green. This gives a good experience visually and perceptually to the user. The figure below shows the target before hitting successfully and after.

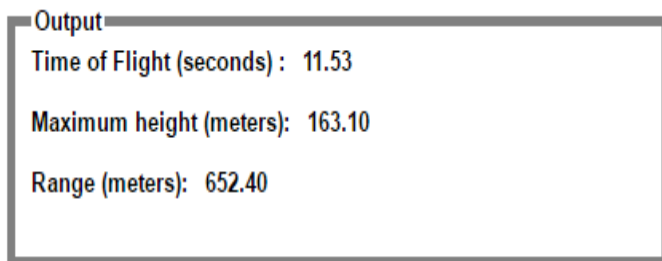


Another very important aspect we have targeted in the solution is displaying the velocity component and time elapsed at each point of the parabolic trajectory. The velocity with which the object is launched has two components – $V \sin \theta$ and $V \cos \theta$ in addition to the resultant velocity. We wanted to give the user as much knowledge about the phenomenon as possible. For displaying the velocity components in addition to the resultant velocity, we decided to have a unique user interface. In our designed interface, there are two perpendicular arrow vectors which are displayed when the user hovers the mouse at a particular instance of the parabolic path. The time elapsed is also displayed along with it. The figure below shows the design and the interface.



This interface helps the user understand the direction of components of velocity and also understand how they change. In addition to it, it also shows the time elapsed to provide the user with a practical and real time sense of the moving object. This showcase of actual time and velocity component at any instance desired by the user lacks in the conventional teaching and learning methods of projectile motion.

Another crucial aspect of understand projectile motion is inferring the output parameters of a particular projectile motion attempt. These output parameters include time of flight, maximum height reached and range. We have dedicated a portion of the application to display these values for each user attempt. The figure below shows the output parameters display. This display is very important is it required for better understanding the phenomenon. After each attempt user will see the updated values of these output parameters. These updated values help user understand how the input variables are effecting them.



To allow user to relate what is happening on screen with background mathematical and physics equations and terms, we have dedicated a small portion of the application screen to a 'Things to Know' list. This provides the user some background knowledge that the user can refer to. The figure below showcases the reference list.

Things to know

- Time of Flight is the time taken by the projectile to go from source to destination.
- Maximum height is the highest point that a projectile reaches.
- Range is the maximum horizontal distance that a projectile covers.

Time of Flight $t = \frac{2v_0 \sin(\theta)}{g}$

Maximum Height $h = \frac{v_0^2 \sin^2(\theta)}{2g}$

Range of Projectile $R = \frac{v_0^2 \sin 2\theta}{g}$

B. Backend Implementation

The backend implementation involves a lot of mathematical calculations and dynamic updating of data. We have used the basic formulas of projectile motion a mentioned earlier in section I-B. In specific, the formula equation (4) is used to plot the parabolic path of the launched object.

The formula equations (1), (2) and (3) are used for the displaying the parametric outputs. With respect to the visualization part, the SVG elements in the middle of the screen, the parabolic path plot, is updated every time there is a change in the input variables. Each time the input variables are changed the new output parameters are calculated at the backend and accordingly the new parabolic path is drawn.

For plotting specifically the parabolic path, an array is formed for x-axis values. Then an array for y-values is formed whose values are calculated through equation (4). Both of these arrays are then plotted against each other to get the parabolic locus on screen.

VI. CONCLUSION

By effective design and implementation of the application we have successfully built a interac

The equations are an exception to the prescribed specifications of this template. You will need to determine whether or not your equation should be typed using either the Times New Roman or the Symbol font (please no other font). To create multileveled equations, it may be necessary to treat the equation as a graphic and insert it into the text after your paper is styled.

Number equations consecutively. Equation numbers, within parentheses, are to position flush right, as in (1), using a right tab stop. To make your equations more compact, you may use the solidus (/), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in

$$a + b = \gamma \quad (1)$$

Note that the equation is centered using a center tab stop. Be sure that the symbols in your equation have been defined before or immediately following the equation. Use “(1)”, not “Eq. (1)” or “equation (1)”, except at the beginning of a sentence: “Equation (1) is . . .”

A. Some Common Mistakes

- The word “data” is plural, not singular.
- The subscript for the permeability of vacuum μ_0 , and other common scientific constants, is zero with subscript formatting, not a lowercase letter “o”.
- In American English, commas, semi-/colons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
- A graph within a graph is an “inset”, not an “insert”. The word *alternately* is preferred to the word “alternately” (unless you really mean something that alternates).
- Do not use the word “essentially” to mean “approximately” or “effectively”.
- In your paper title, if the words “that uses” can accurately replace the word “using”, capitalize the “u”; if not, keep using lower-cased.
- Be aware of the different meanings of the homophones “affect” and “effect”, “complement” and “compliment”, “discreet” and “discrete”, “principal” and “principle”.
- Do not confuse “imply” and “infer”.
- The prefix “non” is not a word; it should be joined to the word it modifies, usually without a hyphen.
- There is no period after the “et” in the Latin abbreviation “et al.”.
- The abbreviation “i.e.” means “that is”, and the abbreviation “e.g.” means “for example”.

An excellent style manual for science writers is [7].

VII. USING THE TEMPLATE

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use

the scroll down window on the left of the MS Word Formatting toolbar.

A. Authors and Affiliations

The template is designed so that author affiliations are not repeated each time for multiple authors of the same affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization). This template was designed for two affiliations.

1) *For author/s of only one affiliation (Heading 3):* To change the default, adjust the template as follows.

a) *Selection (Heading 4):* Highlight all author and affiliation lines.

b) *Change number of columns:* Select the Columns icon from the MS Word Standard toolbar and then select “1 Column” from the selection palette.

c) *Deletion:* Delete the author and affiliation lines for the second affiliation.

2) *For author/s of more than two affiliations:* To change the default, adjust the template as follows.

a) *Selection:* Highlight all author and affiliation lines.

b) *Change number of columns:* Select the “Columns” icon from the MS Word Standard toolbar and then select “1 Column” from the selection palette.

c) *Highlight author and affiliation lines of affiliation 1 and copy this selection.*

d) *Formatting:* Insert one hard return immediately after the last character of the last affiliation line. Then paste down the copy of affiliation 1. Repeat as necessary for each additional affiliation.

e) *Reassign number of columns:* Place your cursor to the right of the last character of the last affiliation line of an even numbered affiliation (e.g., if there are five affiliations, place your cursor at end of fourth affiliation). Drag the cursor up to highlight all of the above author and affiliation lines. Go to Column icon and select “2 Columns”. If you have an odd number of affiliations, the final affiliation will be centered on the page; all previous will be in two columns.

B. Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is “Heading 5”. Use “figure caption” for your Figure captions, and “table head” for your table title. Run-in heads, such as “Abstract”, will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this

one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named “Heading 1”, “Heading 2”, “Heading 3”, and “Heading 4” are prescribed.

C. Figures and Tables

a) *Positioning Figures and Tables:* Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation “Fig. 1”, even at the beginning of a sentence.

TABLE I. TABLE TYPE STYLES

Table Head	Table Column Head		
	Table column subhead	Subhead	Subhead
copy	More table copy ^a		

^a Sample of a Table footnote. (*Table footnote*)

Fig. 1. Example of a figure caption. (*figure caption*)

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity “Magnetization”, or “Magnetization, M”, not just “M”. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write “Magnetization (A/m)” or “Magnetization {A[m(1)]}”, not just “A/m”. Do not label axes with a ratio of quantities and units. For example, write “Temperature (K)”, not “Temperature/K”.

ACKNOWLEDGMENT (*Heading 5*)

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted

expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

REFERENCES

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

- [1] http://sites.nationalacademies.org/cs/groups/dbassesite/documents/webpage/dbasse_080068.pdf. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
- [2] http://www.bsu.edu/eft/sandlot/p/class_linear.html. Elissa, “Title of paper if known,” unpublished.
- [3] https://phet.colorado.edu/sims/projectile-motion/projectile-motion_en.html. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetism Japan, p. 301, 1982].
- [4] <http://www.physicsclassroom.com/Class/vectors/u3l2a.cfm>.