Slide 1:

Title: Projectile Motion Simulation in C++

Subtitle: Unveiling the Dynamics of Motion through Code

Slide 2:

Title: Visualization and Understanding

- Harnessing the Power of Visual Representation

- Visualization serves as an invaluable tool for comprehending intricate concepts.

- By graphically representing data and information, we unlock insights, discern patterns, and effectively communicate ideas.

- Our code's simulation of projectile motion and advanced graphing capabilities provide a captivating visual representation that brings the principles of motion to life.

- As we immerse ourselves in the animated graph and analyze the trajectory, we embark on a captivating journey of exploration and discovery.

- Through this visual portrayal, we not only grasp the concept but also develop a profound appreciation for the intrinsic elegance of physical laws.

Slide 3:

Title: Summary

- Name: Summary

- A Comprehensive C++ Program Unveiling Projectile Motion Simulation

- The code is a robust C++ program that simulates projectile motion and generates dynamic motion graphs using the graphics.h library.

- It prompts users for initial velocities and angles of projection for two projectiles, calculates essential parameters such as time of flight, maximum height, and horizontal range, and plots the motion on a graph.

- Key functions utilized include `drawLine`, `initgraph`, `setcolor`, `line`, `outtextxy`, and `putpixel`.

- The code offers educational value, enhancing conceptual understanding and programming skills, and finds applications in physics education, research, game development, engineering, and more.

- Limitations include dependence on a non-standard library, lack of error handling, limited graphical capabilities, and absence of input validation for grid size.

- With further development, these limitations can be addressed, making it a valuable tool for analyzing projectile motion.

Slide 4:

Title: Introduction

- Name: Introduction

- Unleashing the Power of Simulation: Projectile Motion in Focus

- Our code presents an exceptional simulation of projectile motion, providing users with a captivating visual representation empowered by the graphics.h library.

- Users can input initial velocities and angles of projection for two projectiles, unveiling the mesmerizing behavior of projectiles in flight.

- The code's underlying algorithms seamlessly calculate essential parameters such as time of flight, maximum height, and horizontal range, allowing users to gain comprehensive insights into projectile motion dynamics.

Slide 5:

Title: Utilized Header Files and Functions

- Name: Header Files and Functions

- The code incorporates several header files serving distinct purposes:

1. `<iostream>`: Enables input/output operations for console interaction and printing information.

2. `<graphics.h>`: Compiler-specific library facilitating graphics functions like line drawing and pixel manipulation.

3. `<cmath>`: Provides mathematical functions, including trigonometric calculations required for angle conversions.

4. `<chrono>` and `<thread>`: Support time-related operations and introduce delays in the animation.

Slide 6:

Title: drawLine Function

- Name: drawLine Function

- The code utilizes the `drawLine` function to create lines between specified points on the screen.

- This function accepts coordinates (`x1`, `y1`, `x2`, `y2`) and a color code as arguments.

- It sets the drawing color using `setcolor` and draws lines using `line`.

- The `drawLine` function contributes to the graph's visual structure, facilitating the creation of the grid, x-axis, and y-axis.

Slide 7:

Title: Main Function and Initialization

- Name: Main Function and Initialization

- The `main` function serves as the program's entry point.

- It initializes variables, sets up the graphics system, handles user input, performs calculations, draws the projectile motion graph, prints information, waits for a key press, and closes the graphics system.

Slide 8:

Title: User Input Validation

- Name: User Input Validation

- The code ensures valid user input within specified ranges.

- It employs while loops and conditional statements to prompt the user until valid input is provided.

- Invalid input is handled using `cin.clear()` and `cin.ignore()` to prevent input stream errors.

Slide 9:

Title: Calculation of Projectile Motion Parameters

- Name: Calculation of Projectile Motion Parameters

- The code calculates parameters such as time of flight, maximum height, and horizontal range using relevant formulas and user input.

- It utilizes trigonometric functions from the `<cmath>` library to perform angle conversions and other calculations.

Slide 10:

Title: Graphics Setup and Drawing

- Name: Graphics Setup and Drawing

- The code sets the background color to black using `setbkcolor` and clears the screen using `cleardevice`.

- It draws a green grid using `setcolor` and `line` functions, with 20-pixel spacing for vertical and horizontal lines.

- The `drawLine` function is employed to draw the white x-axis and y-axis.

- The `outtextxy` function displays the "VELOCITY" and "TIME" labels on the graph.

Slide 11:

Title: Animation and Graph Plotting

- Name: Animation and Graph Plotting

- The code animates projectile motion by plotting the trajectory of each projectile using the `putpixel` function.

- It introduces delays between each plot using `std::this\_thread::sleep\_for` to create an animated effect.

- Animation is implemented using for loops and projectile motion equations to calculate x and y coordinates at each time step.

Slide 12:

Title: Printing Calculated Information

- Name: Printing Calculated Information

- The code prints calculated information about projectile motion for both projectiles.

- Output is displayed using `std::cout` statements, with different colors used to differentiate between the two projectiles.

Slide 13:

Title: Graphical Output and User Interaction

- Name: Graphical Output and User Interaction

- The `getch` function waits for a key press before closing the graphics system.

- The `closegraph` function is called to terminate the graphics system after user interaction.

- The `return 0` statement indicates successful program execution.

Slide 14:

Title: Conclusion

- Name: Conclusion

- Our code exemplifies the simulation of projectile motion and graph plotting using the graphics.h library.

- It incorporates essential header files and functions for input/output, graphics, mathematics, time-related operations, and multi-threading.

- The code enables users to visualize and analyze projectile behavior, compute key parameters, and interact with the simulation.

Slide 15:

Title: Appreciation and Thank You

- Name: Appreciation and Thank You

- We express our sincerest appreciation to our teacher for their invaluable guidance and unwavering support throughout this project.

- Their expertise and mentorship have played a pivotal role in deepening our understanding of projectile motion and its practical implementation through code.

- We are profoundly grateful for our teacher's patient explanations, insightful discussions, and dedication to our growth.

Slide 16:

Title: Teacher's Contributions

- Name: Teacher's Contributions

- Our teacher's unwavering commitment to our education and exceptional dedication to our development as students have been truly remarkable.

- Their guidance has not only facilitated our comprehension of physics but also sparked our curiosity and inspired us to explore further in related fields.

- We express our heartfelt gratitude to our teacher for going above and beyond to ensure our success and fostering a love for learning within us.

Slide 17:

Title: Impact on Our Academic Journey

- Name: Impact on Our Academic Journey

- We consider ourselves fortunate to have had such an exceptional mentor, whose influence has left an indel

ible mark on our academic journey.

- The knowledge and skills acquired from working with this code will continue to shape our future endeavors and accomplishments.

- We deeply appreciate our teacher's immense contribution to our academic growth and personal development.

Slide 18:

Title: Thank You, Dear Teacher

- Name: Thank You, Dear Teacher

- Once again, we extend our sincere appreciation and gratitude to our teacher for their unwavering support and exceptional teaching.

- Thank you, dear teacher, for your guidance, inspiration, and unwavering dedication to our education.

Slide 19:

Title: Q&A and Discussion

- Name: Q&A and Discussion

- We now open the floor for any questions, further discussion, or additional insights.

- Let's take this opportunity to explore more about projectile motion, the code, or any related topics.

- Thank you all for your attention and participation.