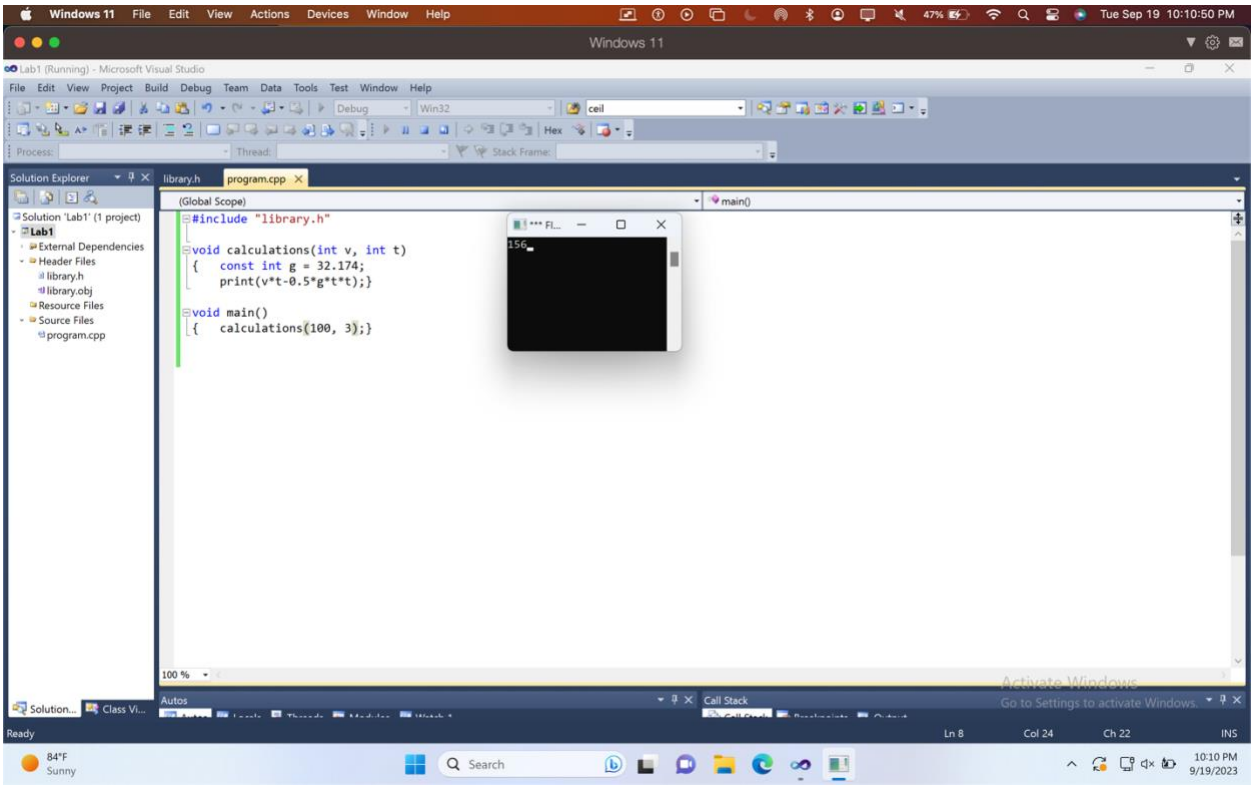
Lab # 4 : A video game: blowing things up with a cannon

ECE 118 – R/RC
Lab on Wednesday at 5:05
Sloan Atkins

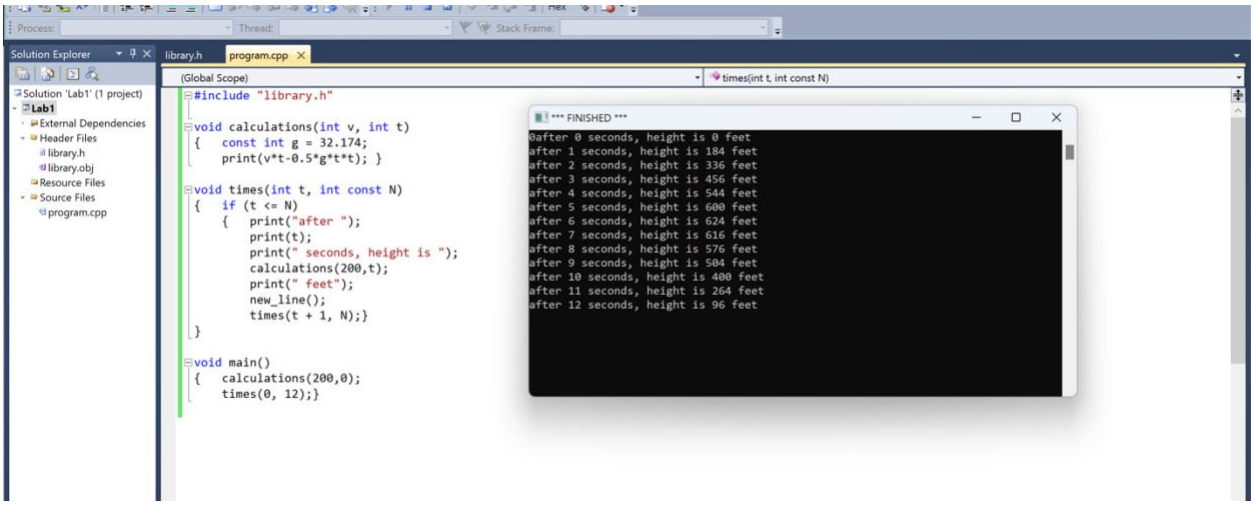
1. The Calculation

```
#include "library.h"

void calculations(int v, int t)
{     const int g = 32.174;
      print(v*t-0.5*g*t*t);}

void main()
{     calculations(100, 3); }
```



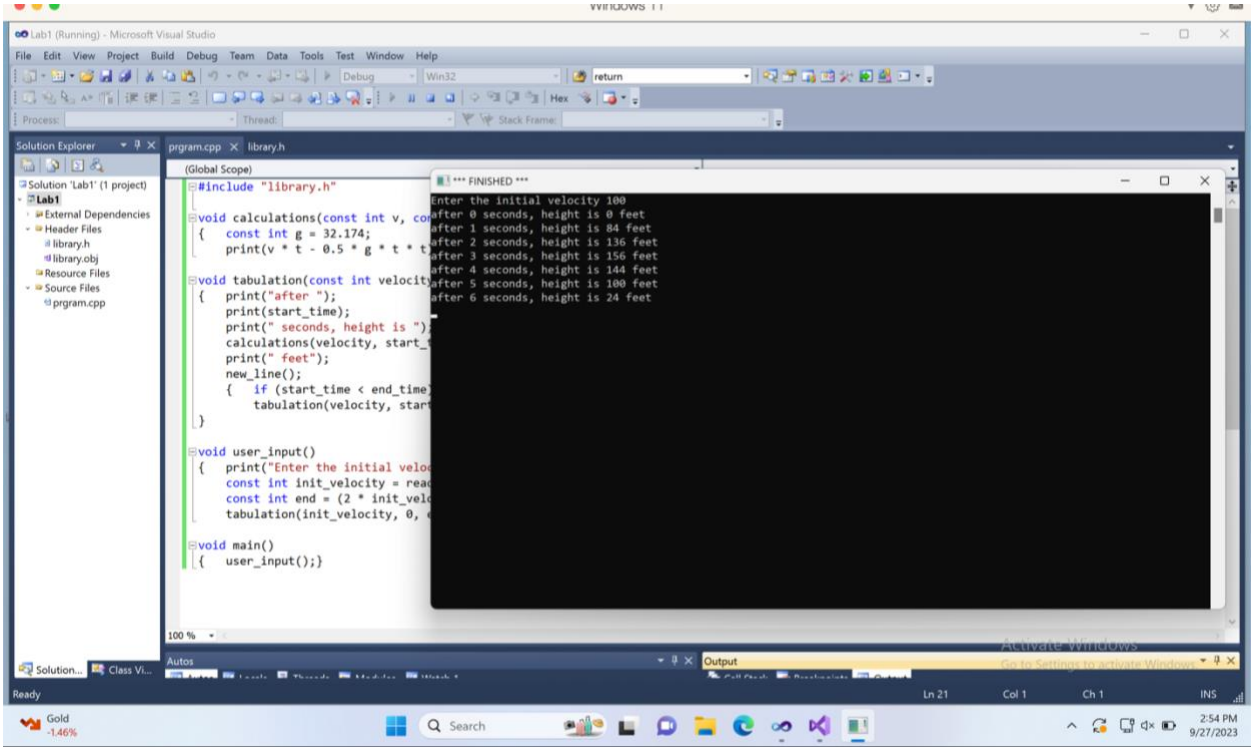
2. Tabulation

```
#include "library.h"

void calculations(const int v, const int t)
{	const int g = 32.174;
	print(v * t - 0.5 * g * t * t); }

void tabulation(const int velocity, const int start_time, const int end_time)
{	print("after ");
	print(start_time);
	print(" seconds, height is ");
	calculations(velocity, start_time);
	print(" feet");
	new_line();
	{	if (start_time < end_time)
		tabulation(velocity, start_time + 1, end_time);}
}

void main()
{	tabulation(200, 0, 13);}
```



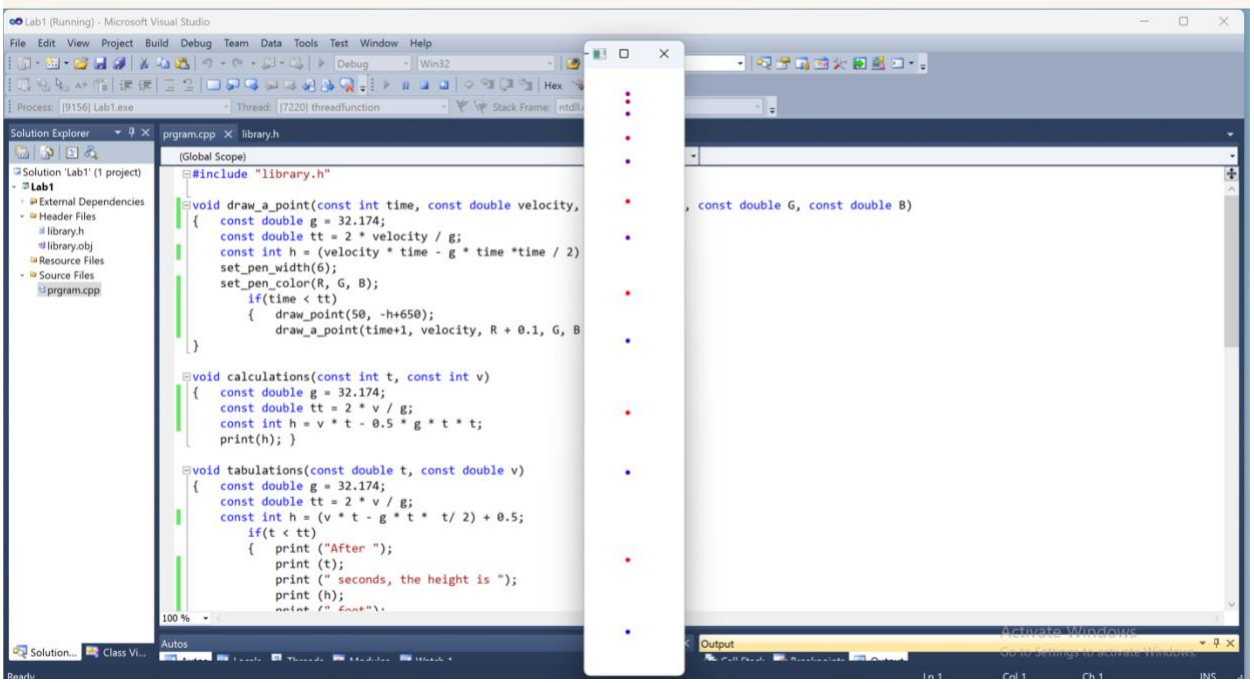
3. Interaction

```
#include "library.h"


void calculations(const int v, const int t)
{	const int g = 32.174;
	print(v * t - 0.5 * g * t * t); }

void tabulation(const int velocity, const int start_time, const int end_time)
{	print("after ");
	print(start_time);
	print(" seconds, height is ");
	calculations(velocity, start_time);
	print(" feet");
	new_line();
	{	if (start_time < end_time)
		tabulation(velocity, start_time + 1, end_time);}
}

void user_input()
{	print("Enter the initial velocity ");
	const int init_velocity = read_int();
	const int end = (2 * init_velocity / 32.174);
	tabulation(init_velocity, 0, end);}

void main()
{	user_input();}
```



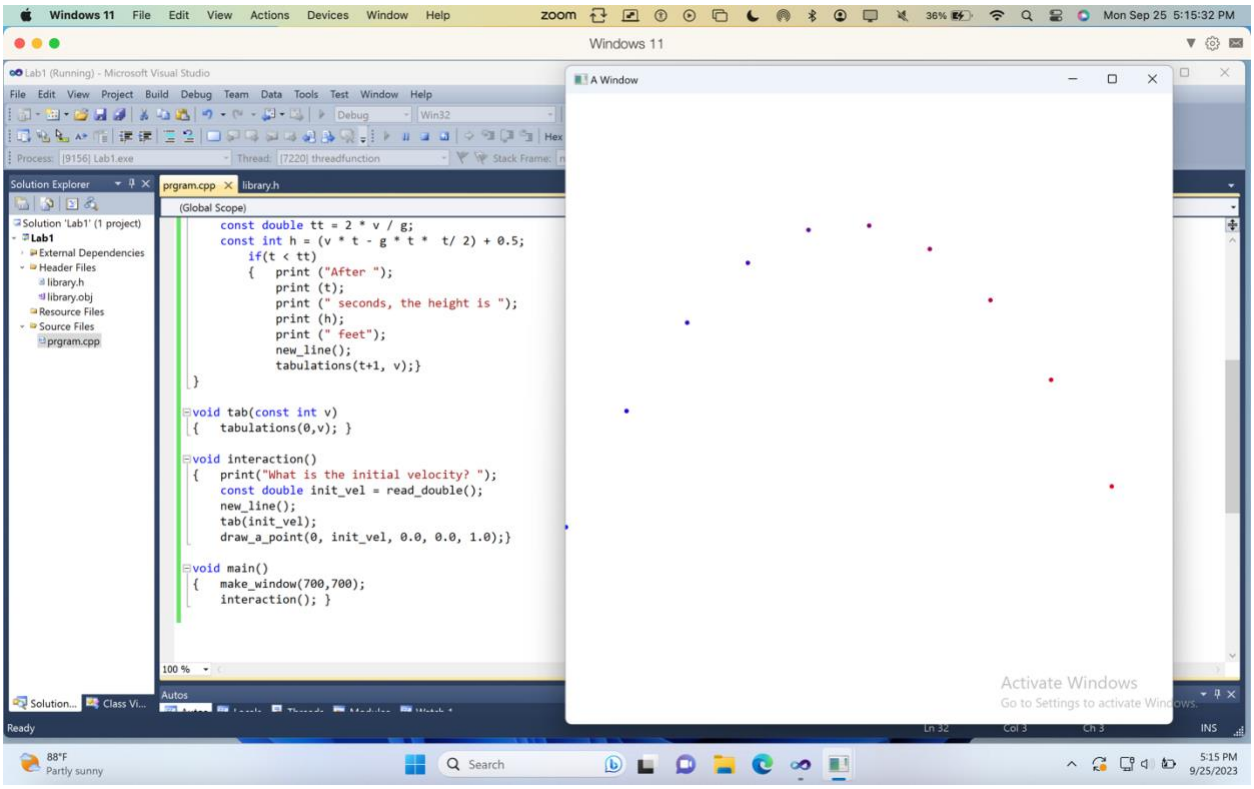
4. *Visual Representation*

```
#include "library.h"

void draw_a_point(const int time, const double velocity, const double R, const
double G, const double B)
{     const double g = 32.174;
      const double tt = 2 * velocity / g;
      const int h = (velocity * time - g * time *time / 2) + 0.5;
      set_pen_width(6);
      set_pen_color(R, G, B);
            if(time < tt)
            {     draw_point(50, -h+650);
                  draw_a_point(time+1, velocity, R + 0.1, G, B - 0.1);}
}

void calculations(const int t, const int v)
{     const double g = 32.174;
      const double tt = 2 * v / g;
      const int h = v * t - 0.5 * g * t * t;
      print(h); }

void tabulations(const double t, const double v)
{     const double g = 32.174;
      const double tt = 2 * v / g;
      const int h = (v * t - g * t *  t/ 2) + 0.5;
            if(t < tt)
            {     print ("After ");
                  print (t);
                  print (" seconds, the height is ");
                  print (h);
                  print (" feet");
                  new_line();
                  tabulations(t+1, v);}
}

void tab(const int v)
{     tabulations(0,v); }

void interaction()
{     print("What is the initial velocity? ");
      const double init_vel = read_double();
      new_line();
      tab(init_vel);
      draw_a_point(0, init_vel, 0.0, 0.0, 1.0); }

void main()
{     make_window(100,700);
      interaction(); }
```


Lab1 (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Team Data Tools Test Window Help
Solution Explorer
Solution 'Lab1' (1 project)
Lab1
External Dependencies
Header Files
library.h
library.obj
Resource Files
Source Files
prgram.cpp
prgram.cpp
library.h
(Global Scope)
#include "library.h"
void draw_a_point(const int time, const double velocity,
{ const double g = 32.174;
const double tt = 2 * velocity / g;
const int h = (velocity * time - g * time *time / 2)
set_pen_width(6);
set_pen_color(R, G, B);
if(time < tt)
{ draw_point(50, -h+650);
draw_a_point(time+1, velocity, R + 0.1, G, B
}
void calculations(const int t, const int v)
{ const double g = 32.174;
const double tt = 2 * v / g;
const int h = v * t - 0.5 * g * t * t;
print(h); }
void tabulations(const double t, const double v)
{ const double g = 32.174;
const double tt = 2 * v / g;
const int h = (v * t - g * t * t/ 2) + 0.5;
if(t < tt)
{ print ("After ");
print (t);
print (" seconds, the height is ");
print (h);
, const double G, const double B)
100 %
Autos
Output
Activate Windows
Go to Settings to activate Windows.
Solution...
Class Vi...
Ready
Ln 1
Col 1
Ch 1
INS

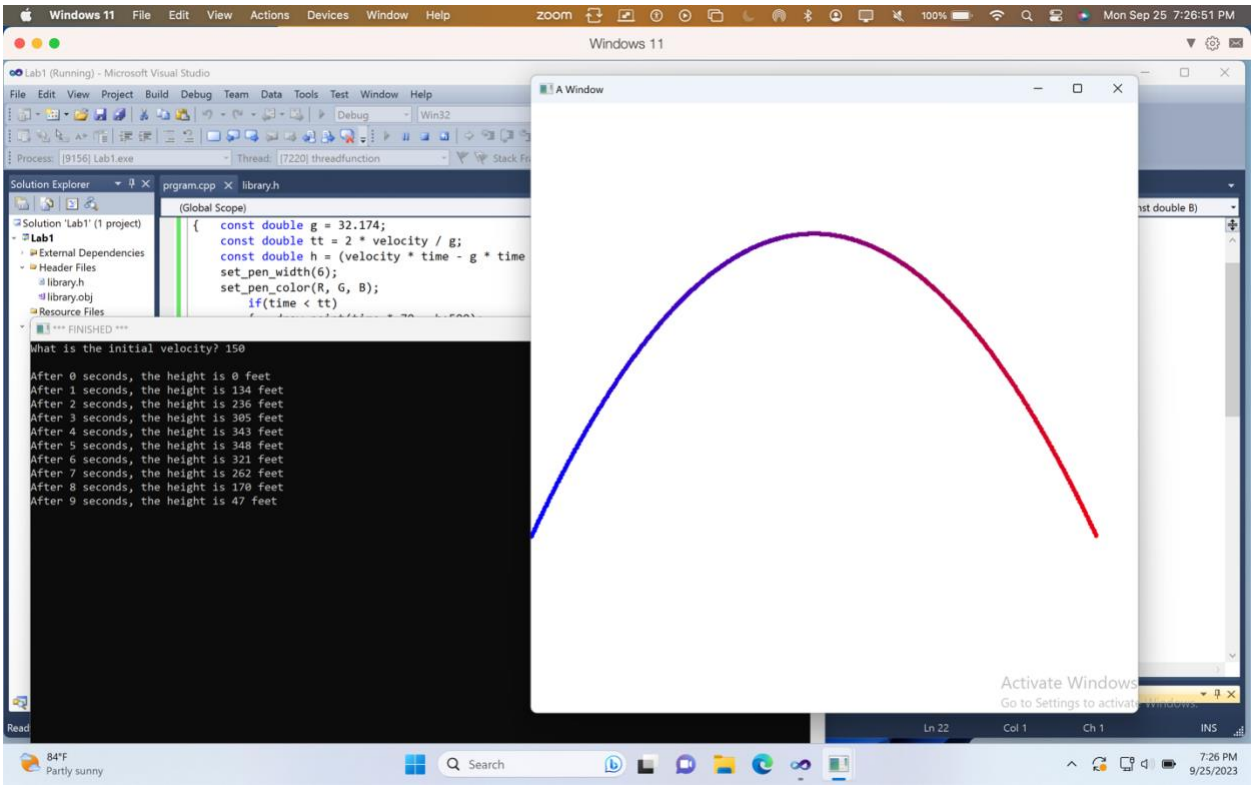
5. *First Flight.*

```
#include "library.h"

void draw_a_point(const int time, const double velocity, const double R, const
double G, const double B)
{     const double g = 32.174;
      const double tt = 2 * velocity / g;
      const int h = (velocity * time - g * time *time / 2) + 0.5;
      set_pen_width(6);
      set_pen_color(R, G, B);
            if(time < tt)
            {     draw_point(time * 70, -h+500);
                  draw_a_point(time+1, velocity, R + 0.1, G, B - 0.1);}
}

void calculations(const int t, const int v)
{     const double g = 32.174;
      const double tt = 2 * v / g;
      const int h = v * t - 0.5 * g * t * t;
      print(h); }

void tabulations(const double t, const double v)
{     const double g = 32.174;
      const double tt = 2 * v / g;
      const int h = (v * t - g * t *  t/ 2) + 0.5;
            if(t < tt)
            {     print ("After ");
                  print (t);
                  print (" seconds, the height is ");
                  print (h);
                  print (" feet");
                  new_line();
                  tabulations(t+1, v);}
}

void tab(const int v)
{     tabulations(0,v); }

void interaction()
{     print("What is the initial velocity? ");
      const double init_vel = read_double();
      new_line();
      tab(init_vel);
      draw_a_point(0, init_vel, 0.0, 0.0, 1.0);}

void main()
{     make_window(700,700);
      interaction(); }
```


Windows 11 File Edit View Actions Devices Window Help
Mon Sep 25 5:15:32 PM
Windows 11
Lab1 (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Team Data Tools Test Window Help
Solution Explorer
Solution 'Lab1' (1 project)
Lab1
External Dependencies
Header Files
library.h
library.obj
Resource Files
Source Files
prgram.cpp
prgram.cpp
library.h
(Global Scope)
const double tt = 2 * v / g;
const int h = (v * t - g * t * t/ 2) + 0.5;
if(t < tt)
{ print ("After ");
print (t);
print (" seconds, the height is ");
print (h);
print (" feet");
new_line();
tabulations(t+1, v);}
}
void tab(const int v)
{ tabulations(0,v); }
void interaction()
{ print("What is the initial velocity? ");
const double init_vel = read_double();
new_line();
tab(init_vel);
draw_a_point(0, init_vel, 0.0, 0.0, 1.0);}
void main()
{ make_window(700,700);
interaction(); }
A Window
Activate Windows
Go to Settings to activate Windows.
Ready
88°F
Partly sunny
Search
5:15 PM
9/25/2023

6. A Nice Arc.

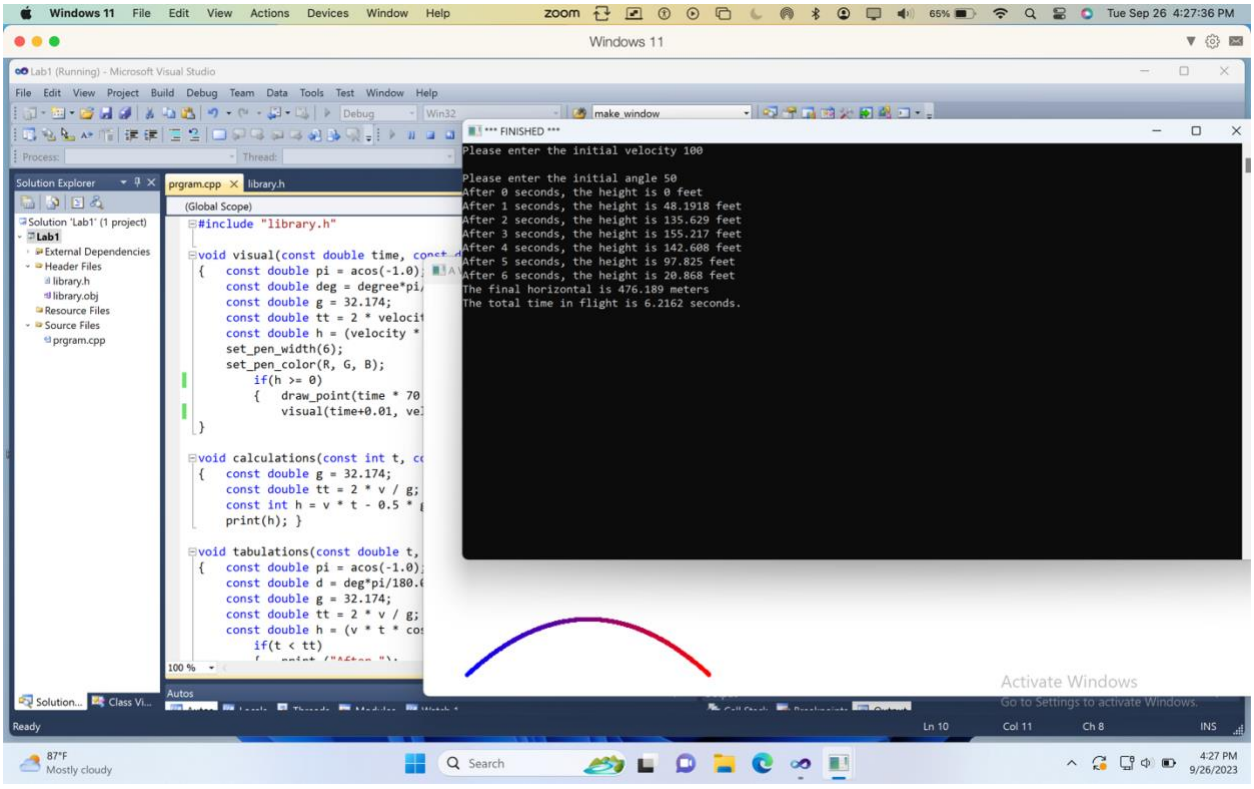
```
#include "library.h"

void draw_a_point(const double time, const double velocity, const double R, const
double G, const double B)
{     const double g = 32.174;
      const double tt = 2 * velocity / g;
      const double h = (velocity * time - g * time *time / 2);
      set_pen_width(6);
      set_pen_color(R, G, B);
            if(time < tt)
            {     draw_point(time * 70, -h+500);
                  draw_a_point(time+0.01, velocity, R + 0.001, G, B - 0.001);}
}

void calculations(const int t, const int v)
{     const double g = 32.174;
      const double tt = 2 * v / g;
      const int h = v * t - 0.5 * g * t * t;
      print(h); }

void tabulations(const double t, const double v)
{     const double g = 32.174;
      const double tt = 2 * v / g;
      const int h = (v * t - g * t *  t/ 2) + 0.5;
            if(t < tt)
            {     print ("After ");
                  print (t);
                  print (" seconds, the height is ");
                  print (h);
                  print (" feet");
                  new_line();
                  tabulations(t+1, v);}
}

void tab(const int v)
{     tabulations(0,v); }

void interaction()
{     print("What is the initial velocity? ");
      const double init_vel = read_double();
      new_line();
      tab(init_vel);
      draw_a_point(0, init_vel, 0.0, 0.0, 1.0);}

void main()
{     make_window(700,700);
      interaction(); }
```


Windows 11
Lab1 (Running) - Microsoft Visual Studio
A Window
Solution Explorer
prgram.cpp
library.h
(Global Scope)
const double g = 32.174;
const double tt = 2 * velocity / g;
const double h = (velocity * time - g * time
set_pen_width(6);
set_pen_color(R, G, B);
if(time < tt)
*** FINISHED ***
What is the initial velocity? 150
After 0 seconds, the height is 0 feet
After 1 seconds, the height is 134 feet
After 2 seconds, the height is 236 feet
After 3 seconds, the height is 305 feet
After 4 seconds, the height is 343 feet
After 5 seconds, the height is 348 feet
After 6 seconds, the height is 321 feet
After 7 seconds, the height is 262 feet
After 8 seconds, the height is 170 feet
After 9 seconds, the height is 47 feet
Activate Windows
Search
84°F
Partly sunny
7:26 PM
9/25/2023

7. Take the Battle to the Enemy.

```
#include "library.h"

void visual(const double time, const double velocity, const int degree, const
double R, const double G, const double B)
{	const double pi = acos(-1.0);
	const double deg = degree*pi/180.00;
	const double g = 32.174;
	const double tt = 2 * velocity / g;
	const double h = (velocity * time * cos(deg) - g * time * time / 2);
	set_pen_width(6);
	set_pen_color(R, G, B);
		if(h >= 0)
		{	draw_point(time * 70 + 50, -h+450);
			visual(time+0.01, velocity, degree, R + 0.0025, G, B -
0.0025);}
}

void calculations(const int t, const int v)
{	const double g = 32.174;
	const double tt = 2 * v / g;
	const int h = v * t - 0.5 * g * t * t;
	print(h); }

void tabulations(const double t, const double v, const double deg)
{	const double pi = acos(-1.0);
	const double d = deg*pi/180.00;
	const double g = 32.174;
	const double tt = 2 * v / g;
	const double h = (v * t * cos(deg) - g * t * t / 2);
		if(t < tt)
		{	print ("After ");
			print (t);
			print (" seconds, the height is ");
			print (h);
			print (" feet");
			new_line();
			tabulations(t+1, v, d);}
}

void final(const double t, const double v, const double degree)
{	const double pi = acos(-1.0);
	const double deg = degree*pi/180.00;
	const double g = 32.174;
	const double tt = 2 * v / g;
	const double h = (v * t * cos(deg) - g * t * t / 2);
	const double d = v * t * sin(deg);
	print("The final horizontal is ");
	print(d);
	print(" meters");
	new_line();
```

```
      print("The total time in flight is ");
      print(tt);
      print(" seconds."); }

void battle()
{     print("Please enter the initial velocity ");
      const double init_vel = read_double();
      new_line();
      print("Please enter the initial angle ");
      const double degree = read_double();
      const double g = 32.174;
      const double tt = 2 * init_vel / g;
      tabulations(0, init_vel, degree);
      final(tt, init_vel, degree);
      visual(0, init_vel, degree, 0.0, 0.0, 1.0);}

void main()
{     make_window(1000,500);
      battle();}
```

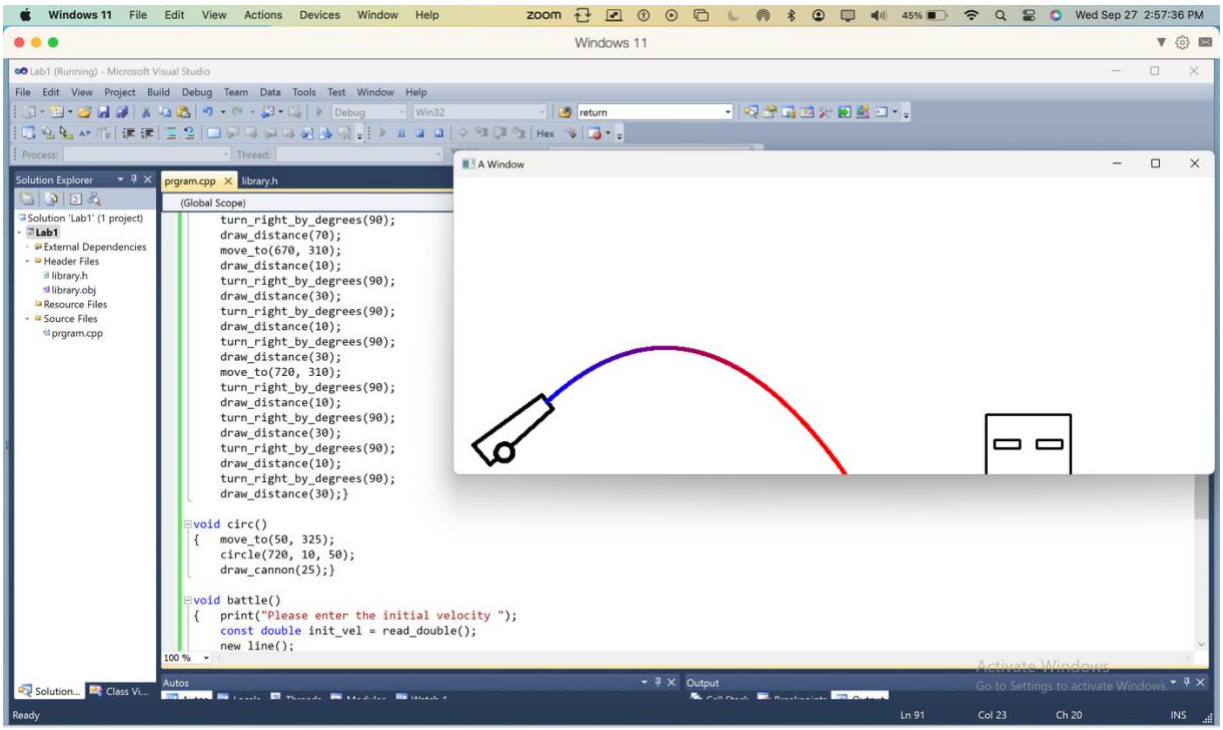


8. The Game

```
#include "library.h"

void visual(const double time, const double velocity, const int degree, const
double R, const double G, const double B)
{	const double pi = acos(-1.0);
	const double deg = degree*pi/180.00;
	const double g = 32.174;
	const double tt = 2 * velocity / g;
	const double h = (velocity * time * cos(deg) - g * time * time / 2);
	set_pen_width(6);
	set_pen_color(R, G, B);
		if(time < tt)
		{	draw_point(time * 70 + 110, -h+265);
			visual(time+0.01, velocity, degree, R + 0.0025, G, B -
0.0025);}
}

void calculations(const int t, const int v)
{	const double g = 32.174;
	const double tt = 2 * v / g;
	const int h = v * t - 0.5 * g * t * t;
	print(h); }

void tabulations(const double t, const double v, const double deg)
{	const double pi = acos(-1.0);
	const double d = deg*pi/180.00;
	const double g = 32.174;
	const double tt = 2 * v / g;
	const double h = (v * t * cos(deg) - g * t * t / 2);
		if(t < tt)
		{	print ("After ");
			print (t);
			print (" seconds, the height is ");
			print (h);
			print (" feet");
			new_line();
			tabulations(t+1, v, d);}
}

void final(const double t, const double v, const double degree)
{	const double pi = acos(-1.0);
	const double deg = degree*pi/180.00;
	const double g = 32.174;
	const double tt = 2 * v / g;
	const double h = (v * t * cos(deg) - g * t * t / 2);
	const double d = v * t * sin(deg);
	print("The final horizontal is ");
	print(d);
	print(" meters");
	new_line();
```

```
	print("The total time in flight is ");
	print(tt);
	print(" seconds."); }

void circle(const int N, int radius, int degrees)
{	if (N > 90 - degrees)
	{	const double pi = acos(-1.0);
		double x = (2*pi*radius)/360;
		set_pen_width(6);
		draw_distance(x);
		turn_right_by_degrees(1);
		circle(N-1, radius, degrees);}
}

void draw_cannon(const double radius)
{	set_pen_color(color::black);
	set_pen_width(5);
	start_shape();
	turn_left_by_degrees(90);
	draw_distance(0.5*radius);
	turn_right_by_degrees(90);
	draw_distance(1.25*radius);
	turn_right_by_degrees(94);
	draw_distance(4*radius);
	turn_right_by_degrees(87);
	draw_distance(0.85*radius);
	turn_right_by_degrees(87);
	draw_distance(2.68*radius);
}

void headquarters()
{	set_pen_width(3);
	turn_right_by_degrees(132);
	move_to(630, 350);
	draw_distance(70);
	turn_right_by_degrees(90);
	draw_distance(100);
	turn_right_by_degrees(90);
	draw_distance(70);
	move_to(670, 310);
	draw_distance(10);
	turn_right_by_degrees(90);
	draw_distance(30);
	turn_right_by_degrees(90);
	draw_distance(10);
	turn_right_by_degrees(90);
	draw_distance(30);
	move_to(720, 310);
	turn_right_by_degrees(90);
	draw_distance(10);
	turn_right_by_degrees(90);
	draw_distance(30);
```

```
      turn_right_by_degrees(90);
      draw_distance(10);
      turn_right_by_degrees(90);
      draw_distance(30);}

void circ()
{     move_to(50, 325);
      circle(720, 10, 50);
      draw_cannon(25);}

void battle()
{     print("Please enter the initial velocity ");
      const double init_vel = read_double();
      new_line();
      print("Please enter the initial angle ");
      const double degree = read_double();
      const double g = 32.174;
      const double tt = 2 * init_vel / g;
      tabulations(0, init_vel, degree);
      final(tt, init_vel, degree);
      visual(0, init_vel, degree, 0.0, 0.0, 1.0);}

void main()
{     make_window(900,350);
      circ();
      headquarters();
      battle();}
```

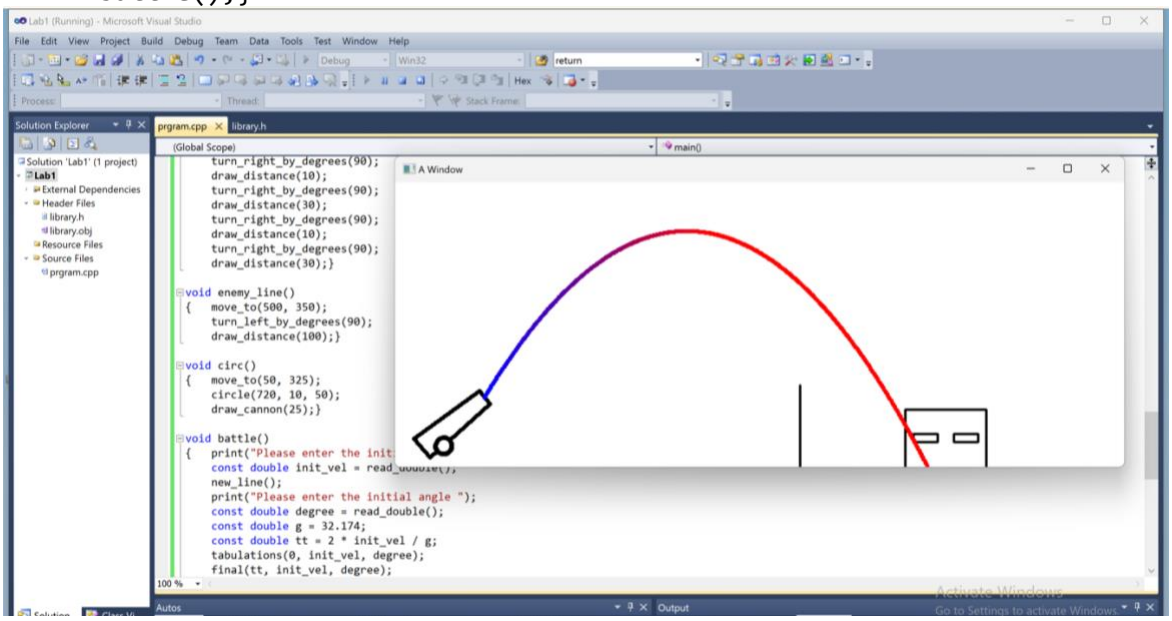


9. Enemy Defenses

```
#include "library.h"

void visual(const double time, const double velocity, const int degree, const
double R, const double G, const double B)
{	const double pi = acos(-1.0);
	const double deg = degree*pi/180.00;
	const double g = 32.174;
	const double tt = 2 * velocity / g;
	const double h = (velocity * time * cos(deg) - g * time * time / 2);
	set_pen_width(6);
	set_pen_color(R, G, B);
		if(time < tt)
		{	draw_point(time * 70 + 110, -h+265);
			visual(time+0.01, velocity, degree, R + 0.0025, G, B -
0.0025);}
}

void calculations(const int t, const int v)
{	const double g = 32.174;
	const double tt = 2 * v / g;
	const int h = v * t - 0.5 * g * t * t;
	print(h); }

void tabulations(const double t, const double v, const double deg)
{	const double pi = acos(-1.0);
	const double d = deg*pi/180.00;
	const double g = 32.174;
	const double tt = 2 * v / g;
	const double h = (v * t * cos(deg) - g * t * t / 2);
		if(t < tt)
		{	print ("After ");
			print (t);
			print (" seconds, the height is ");
			print (h);
			print (" feet");
			new_line();
			tabulations(t+1, v, d);}
}

void final(const double t, const double v, const double degree)
{	const double pi = acos(-1.0);
	const double deg = degree*pi/180.00;
	const double g = 32.174;
	const double tt = 2 * v / g;
	const double h = (v * t * cos(deg) - g * t * t / 2);
	const double d = v * t * sin(deg);
	print("The final horizontal is ");
	print(d);
	print(" meters");
	new_line();
```

```
	print("The total time in flight is ");
	print(tt);
	print(" seconds."); }

void circle(const int N, int radius, int degrees)
{	if (N > 90 - degrees)
	{	const double pi = acos(-1.0);
		double x = (2*pi*radius)/360;
		set_pen_width(6);
		draw_distance(x);
		turn_right_by_degrees(1);
		circle(N-1, radius, degrees);}
}

void draw_cannon(const double radius)
{	set_pen_color(color::black);
	set_pen_width(5);
	start_shape();
	turn_left_by_degrees(90);
	draw_distance(0.5*radius);
	turn_right_by_degrees(90);
	draw_distance(1.25*radius);
	turn_right_by_degrees(94);
	draw_distance(4*radius);
	turn_right_by_degrees(87);
	draw_distance(0.85*radius);
	turn_right_by_degrees(87);
	draw_distance(2.68*radius);
}

void headquarters()
{	set_pen_width(3);
	turn_right_by_degrees(132);
	move_to(630, 350);
	draw_distance(70);
	turn_right_by_degrees(90);
	draw_distance(100);
	turn_right_by_degrees(90);
	draw_distance(70);
	move_to(670, 310);
	draw_distance(10);
	turn_right_by_degrees(90);
	draw_distance(30);
	turn_right_by_degrees(90);
	draw_distance(10);
	turn_right_by_degrees(90);
	draw_distance(30);
	move_to(720, 310);
	turn_right_by_degrees(90);
	draw_distance(10);
	turn_right_by_degrees(90);
	draw_distance(30);
```

```
      turn_right_by_degrees(90);
      draw_distance(10);
      turn_right_by_degrees(90);
      draw_distance(30);}

void enemy_line()
{     move_to(500, 350);
      turn_left_by_degrees(90);
      draw_distance(100);}

void circ()
{     move_to(50, 325);
      circle(720, 10, 50);
      draw_cannon(25);}

void battle()
{     print("Please enter the initial velocity ");
      const double init_vel = read_double();
      new_line();
      print("Please enter the initial angle ");
      const double degree = read_double();
      const double g = 32.174;
      const double tt = 2 * init_vel / g;
      tabulations(0, init_vel, degree);
      final(tt, init_vel, degree);
      visual(0, init_vel, degree, 0.0, 0.0, 1.0);}

void main()
{     make_window(900,350);
      circ();
      headquarters();
      enemy_line();
      battle();}
```



10. Make it Interesting.

```
#include "library.h"

void visual(const double time, const double velocity, const int degree, const
double R, const double G, const double B)
{	const double pi = acos(-1.0);
	const double deg = degree*pi/180.00;
	const double g = 32.174;
	const double tt = 2 * velocity / g;
	const double h = (velocity * time * cos(deg) - g * time * time / 2);
	set_pen_width(6);
	set_pen_color(R, G, B);
		if(time < tt)
		{	draw_point(time * 70 + 110, -h+265);
			visual(time+0.01, velocity, degree, R + 0.0025, G, B -
0.0025);}
}

void calculations(const int t, const int v)
{	const double g = 32.174;
	const double tt = 2 * v / g;
	const int h = v * t - 0.5 * g * t * t;
	print(h); }

void tabulations(const double t, const double v, const double deg)
{	const double pi = acos(-1.0);
	const double d = deg*pi/180.00;
	const double g = 32.174;
	const double tt = 2 * v / g;
	const double h = (v * t * cos(deg) - g * t * t / 2);
		if(t < tt)
		{	print ("After ");
			print (t);
			print (" seconds, the height is ");
			print (h);
			print (" feet");
			new_line();
			tabulations(t+1, v, d);}
}

void final(const double t, const double v, const double degree)
{	const double pi = acos(-1.0);
	const double deg = degree*pi/180.00;
	const double g = 32.174;
	const double tt = 2 * v / g;
	const double h = (v * t * cos(deg) - g * t * t / 2);
	const double d = v * t * sin(deg);
	print("The final horizontal is ");
	print(d);
	print(" meters");
	new_line();
```

```
	print("The total time in flight is ");
	print(tt);
	print(" seconds."); }

void circle(const int N, int radius, int degrees)
{	if (N > 90 - degrees)
	{	const double pi = acos(-1.0);
		double x = (2*pi*radius)/360;
		set_pen_width(6);
		draw_distance(x);
		turn_right_by_degrees(1);
		circle(N-1, radius, degrees);}
}

void draw_cannon(const double radius)
{	set_pen_color(color::black);
	set_pen_width(5);
	start_shape();
	turn_left_by_degrees(90);
	draw_distance(0.5*radius);
	turn_right_by_degrees(90);
	draw_distance(1.25*radius);
	turn_right_by_degrees(94);
	draw_distance(4*radius);
	turn_right_by_degrees(87);
	draw_distance(0.85*radius);
	turn_right_by_degrees(87);
	draw_distance(2.68*radius);
}

void headquarters()
{	int x = random_in_range(0, 130);
	set_pen_width(3);
	turn_right_by_degrees(132);
	move_to(x + 630, 350);
	draw_distance(70);
	turn_right_by_degrees(90);
	draw_distance(100);
	turn_right_by_degrees(90);
	draw_distance(70);
	move_to(x + 670, 310);
	draw_distance(10);
	turn_right_by_degrees(90);
	draw_distance(30);
	turn_right_by_degrees(90);
	draw_distance(10);
	turn_right_by_degrees(90);
	draw_distance(30);
	move_to(x + 720, 310);
	turn_right_by_degrees(90);
	draw_distance(10);
	turn_right_by_degrees(90);
```

```
      draw_distance(30);
      turn_right_by_degrees(90);
      draw_distance(10);
      turn_right_by_degrees(90);
      draw_distance(30);}

void enemy_line()
{     move_to(500, 350);
      turn_left_by_degrees(90);
      draw_distance(100);}

void circ()
{     move_to(50, 325);
      circle(720, 10, 50);
      draw_cannon(25);}

void battle()
{     print("Please enter the initial velocity ");
      const double init_vel = read_double();
      new_line();
      print("Please enter the initial angle ");
      const double degree = read_double();
      const double g = 32.174;
      const double tt = 2 * init_vel / g;
      tabulations(0, init_vel, degree);
      final(tt, init_vel, degree);
      visual(0, init_vel, degree, 0.0, 0.0, 1.0);}

void main()
{     make_window(900,350);
      circ();
      headquarters();
      enemy_line();
      battle();
      new_line();
      new_line();
      battle();
      new_line();
      new_line();
      battle();}
```