ASSIGNMENT 4
COMP 1010: International College of Manitoba
COURSE TITLE: Introduction to Computer Science 1
TERM: Summer 2023

Assignment 4

DUE DATE: JULY 16TH, 2023

Preparation:

Review the following topics

- variables
- final named constants
- Active Processing: setup() and draw()
- State-driven programming
- Data types – int or float and boolean
- IF statement and nested IF
- Relational Boolean operators
- For loop
- User-defined functions part 2: Functions with input parameters and return values.

Notes:

- Name your sketches using your name, the assignment number, and the question number, exactly as in this example DentStuwA4Q1.
- The purpose of this program is to emulate the game of **Cannonball**. A cannonball is placed at the top of a cliff and there are targets at the bottom. Angle the cannonball in such a way that projectile hits a specific target.
- Assignments must follow the programming standards document.
- After the due date, the moodle submission form will be closed and you will receive zero if you did not submit on time.
- You can submit only one time to the submission form and you cannot delete the older version.
- These assignments are your chance to learn the material for the exams. ***Code your assignments independently*.** We use software to compare all submitted assignments to each other, and pursue academic dishonestly vigorously.

Background

This is a simple game. A cannonball is at the top of a cliff. There is a series of targets at the bottom of the cliff. The player tries to fire the cannonball at an angular velocity suitable to hit the target. The cannonball can only be fired at an angle between 0 and 90 degrees with respect to the cliff. The player can change the speed of the cannonball by pulling the cannonball away from its initial position.

The player wins if the cannonball hits the target and loses if it does not.

Terms:

Cannonball: A simple circle of radius 25 pixels.
Target: A series of rectangles at the bottom of the cliff that the cannonball hits once it lands.
Desired Target: The target that the cannonball must hit in order to win.
Projectile path: The path the cannonball follows once fired at a certain angle and speed. Use the parabolic equation to calculate the projectile path.
Cliff: An object of height 300 pixels on which the cannonball is drawn. Use a quad or a rectangle to draw a cliff (see below).
Angle: The angle at which the cannonball fires.
Velocity: The velocity at which the cannonball fires.

In order to achieve the results, follow these steps.

Step 1: Draw a cliff and a cannonball on the canvas. [2 Marks]

- Draw a cliff using a quad or a rect.
- Create a cannonball using an ellipse and place it on the cliff.
- Declare appropriate variables and constants for the cliff and the cannonball.
- Figure 1 provides an illustration.

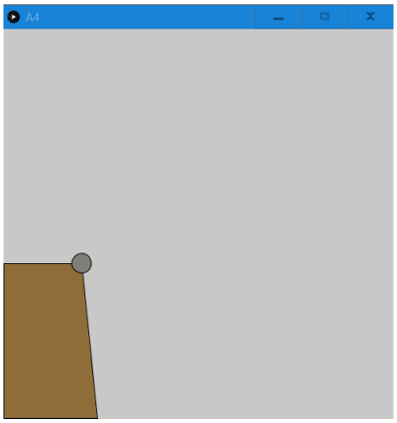


Figure 1

Step 2: Draw targets at the bottom of the cliff. [2 Marks]

- Using a loop (**for** or **while**), draw a series of rectangles, at the bottom of the canvas.
- One of the rectangles is the desired target, which needs to be hit to win the game.
- Color the desired target **green**. Use the **random()** function to randomly assign green to a target every time the program runs.
- Color all the other targets **red**.
- Figure 2 provides an illustration.

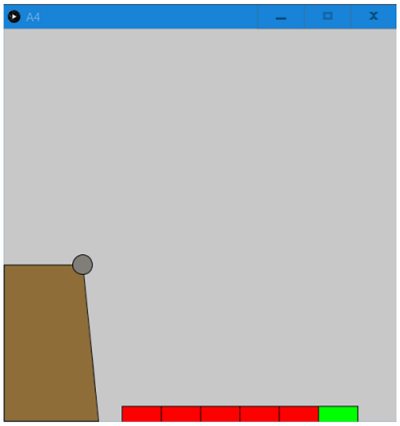


Figure 2

Step 3: Draw angle guides from the cannonball [6 Marks]

- For simplicity, in this question, the cannonball fires at angles between 0^o degrees and 90^o degrees, non-inclusive.
- Draw three angle guides as explained below.
- **Minimum angle guide**: Draw a line at an angle horizontal to the cliff. Specify a fix line size for this guide.
- **Maximum angle guide**: Draw a line at an angle vertical to the cliff. This guide will be of the same length as minimum angle guide.
- **Active angle guide**: This guide is displayed during the **click-pull-aim-release** process explained in Step 4. When the player clicks on the cannonball and while keeping the mouse button pressed, pulls the cannonball down, the active angle guide is drawn.
- **Active angle (angle)**: This is the angle that the mouse position makes with the initial cannonball position when the player clicks and pulls the cannonball downwards, away from the initial position.
- The active angle guide is a line drawn from the current mouse position at an angle of current angle through the initial cannonball position. The length of this guide is the sum of the distance between the current mouse position and initial cannonball position and the length of minimum angle guide. Draw this guide with more stroke weight than the other two guides.
- Declare necessary variables
- Create a function for this step.
- Figure 3 & 4 provides an illustration.

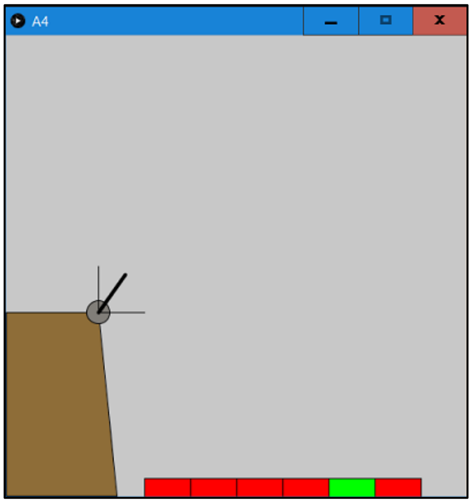
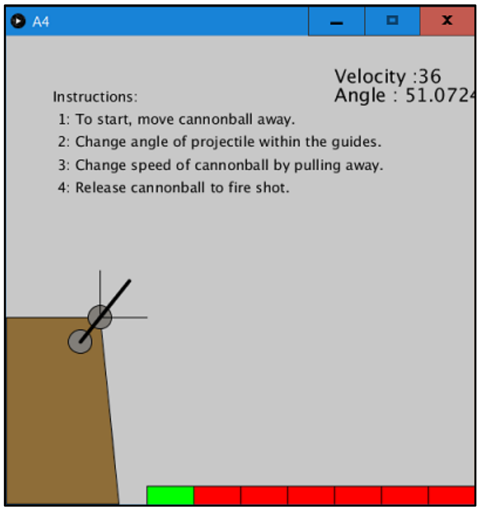


Figure 3

Step 4: Fire the cannonball [8 Marks]

- In order to fire the cannonball, the player should **click-pull-aim-release**.
- The player should **click** on the cannonball and while keeping the mouse button pressed, **pull** the cannonball backwards.
- Pulling the cannonball changes the velocity at which the cannonball fires. The player can increase the velocity of the cannonball by pulling the cannonball further away from its initial position.
- The velocity of the cannonball is determined by the distance of the mouse from the initial position of the cannonball. However, velocity should be within a maximum and minimum value. Declare two constants for the minimum and maximum velocity. A maximum velocity will prevent the cannonball from overshooting the canvas. Similarly, a minimum velocity prevents the cannonball from falling back on to the cliff.
- In order to **aim** the projectile path at the desired target, the player needs to change the angle of the projectile path. This is achieved by moving the cannonball such that the current angle guide stays within the minimum and maximum angle guide.



Velocity :36
Angle : 51.0724

Instructions:
1: To start, move cannonball away.
2: Change angle of projectile within the guides.
3: Change speed of cannonball by pulling away.
4: Release cannonball to fire shot.

Figure 4

- To fire the cannonball, the player **releases** the mouse button, which begins the movement of the cannonball along a projectile path calculated by Step 5.
- To assist the player, current angle at which the active guide is drawn and current velocity is displayed at the top right of the canvas.

Step 5: Move the cannonball along a projectile path [8 Marks]

- The cannonball is fired at an angle with a certain velocity and moves on a path that follows a parabolic path.
- If an object is thrown from a cliff of height **cliffH** and width **cliffW**, at an initial angle **angle** and an initial velocity **V**, then the object follows a parabolic path. The X and Y coordinates of the object, **posX** and **posY,** are represented by the parametric equations:

  .

  $$posX = cliffW + V_x * \mathbf{t}$$
  $$posY = 16 * t^2 + V_y * t + cliffH,$$

  where *t* represents time,
  V_x is the X-component of the initial velocity **V**,
  V_y is the Y-component of the velocity **V.**

  ***V**ₓ* is X-component of the velocity and is the product of velocity and cos of the angle that cannonball makes with the X-axis.
  $\boldsymbol{V_y}$ is the Y-component of the velocity and is the product of velocity and sin of the angle that cannonball makes with the Y-axis.
  t in the above equation time. It determines the location of the cannonball on the projectile path after t seconds. If the value of t =1, it will provide posX and posY of the cannonball on the projectile path at time 1 seconds. Refer to video **A4_time_illustration.mp4** for reference. It shows the position of the ball with respect to time. The video is captured at lower frame rate for explanation. You are not required to modify the frame rate.

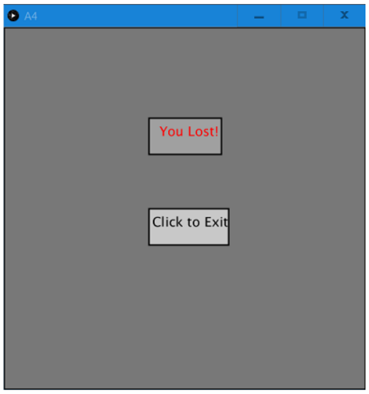


Figure 5

- The value of t will have to be incremented by a small amount (say, 0.1) over multiple frames to move the cannonball on its parabolic path.
- Modify the program such that once the cannonball is fired, it does not leave the canvas and stays at the bottom of the canvas.
- Watch the video for illustration.
- Create a function for this step.

Step 6: Display Instructions. [1 Mark]

- Display the four instructions as shown in the Figure 4.

Step 7: Determine when the game is over and display exit options [3 Marks]

- If the cannonball hits the desired target, the player wins; otherwise, the player loses. At that point, the program should stop and display the result (namely, You Win! or You Lost!) and an exit button.
- The exit button should close the canvas.

- Declare a function to achieve this functionality.
- Figure 5 and 6 on the right illustrates the two exit options for reference.
- Watch **A4_gameWin.mp4** and **A4_gameLost.mp4** for illustration.

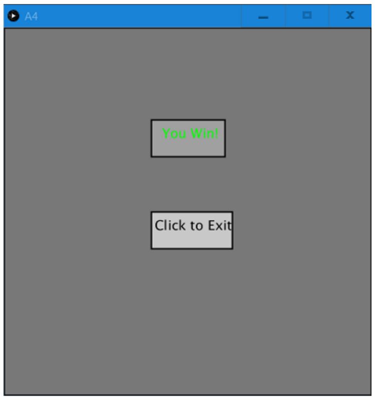


Figure 6

Other requirements:

a) Use a canvas size of 500 by 500
b) Use constants and float or int primitive data types where necessary.
c) Use variables where necessary.
d) Use Strings as and when required.
e) Use appropriate colors.
f) **IMPORTANT:** At least three of the functions declared in the program should have input parameters and should return a value.

NOTE: Use only the built-in functions and features taught in class or in the notes. If unsure, please consult your instructor.

Total marks: 30

End of assignment 4