Outer Billiard Report

Evan Huynh

September 16, 2017

Abstract

This report gives a briefly look about our code and progress in the "Outer Billiard" project.

Contents

List of Figures

1 Introduction

This report gives a briefly look about our code and progress in the "Outer Billiard" project in Mathematica.[1]

---

[1] This program was originally written in Mathematica 11.

2 Method

As stated in the problem, the ball outside of the triangle travels a distance d to one corner, the continue with d before deflect to another corner. This problem can be solved using vector. Let assume there exists two points K (x_K, y_K) and L (x_L, y_L). The new point M which is a reflection of K over L will have the coordinate $x_M = x_L - x_K$, $y_M = y_L - y_K$. In order word, by translating point L with $\overrightarrow{LK}$, we obtained the coordinate of point M. This concept can be demonstrate with a function **reflectPoint** in the appendix.

One such issue exist in this problem is to ascertain which corner of the $\triangle ABC$ to draw the next symmetrical point. From the hypothesis, the billiard travel across the first corner, then the second, then the third, and repeat the process. Since the ball travels and marks the location before changing its direction, the number of marks determine which direction it goes when changing the direction.

To repeat the process systematically, we first let the ball travels to A then B and C, consecutively. This can be solved using modularity. Let n be the marked location, including the initial location K (x_K, y_K). if $n \equiv 1 \pmod{3}$ the ball goes toward A, or if $n \equiv 2 \pmod{3}$, it goes over B. else it runs to C. This strategy works as drawing n segments require at least $n + 1$ points, and since the drawing process repeats over three corner of the equilateral triangle, the modularity of 3 of marked points determines the next directions the ball moves.

3 Result

4 Discussion

As shown in the figure, the point will returns to the initial position after a number of finite turns. We can prevent this from happens by randomly choosing a next corner for the next direction of the ball. Instead, we decided to keep our original solution, even though the result might not be like "Outer Billiard".
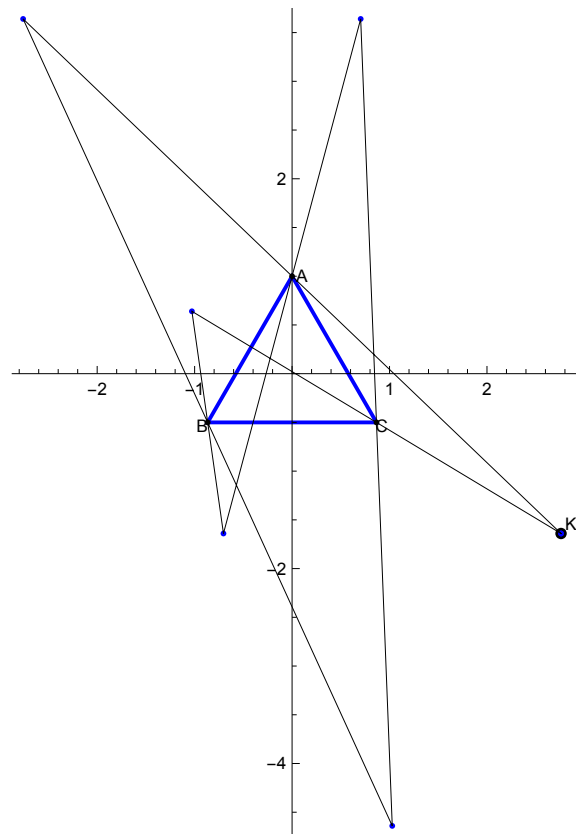
Figure 1: This figure demonstrate moving direction of the particle from outside of the triangle, generated by Mathematica.

5 Appendix

5.1 Mathematica

5.1.1 Function

```
1 reflectPoint[outPoint_?ListQ,middlePoint_?ListQ]:=(
2 xMove = -(outPoint[[1]]-middlePoint[[1]]);
3 yMove = -(outPoint[[2]]-middlePoint[[2]]);
4 {middlePoint[[1]]+xMove,middlePoint[[2]]+yMove}
5 )
```

5.1.2 Main program

```
1 Manipulate[
2 (*----------------*)
3 (*Create the first equilateral triangle*)
4 triangle1=Polygon[CirclePoints[3]];
5 xA=0;yA=1; pointA={xA,yA}; textA= Text["A",{0.1,1}];
6 xB=-(Sqrt[3]/2);yB=-(1/2); pointB={xB,yB}; textB = Text ["B"
     ↪ ,{-0.92,-0.54}];
7 xC=Sqrt[3]/2; yC=-(1/2); pointC={xC,yC}; textC = Text["C"
     ↪ ,{0.92,-0.54}];
8 xK;yK; pointK={xK,yK}; textK=Text["K",{xK+0.1,yK+0.1}];
9 pointList={pointK};
10
11 plot2={EdgeForm[Directive[Thick,Blue]],Directive[White],
     ↪ triangle1,Directive[Black],Point[pointA],Point[pointB
     ↪ ],Point[pointC],textA,textB,textC,PointSize[0.02],
     ↪ Point[pointK],textK};
12
13 (*Add point to list*)
14 doCtimes; doCtimes=Floor[doCtimes];
15 Do[
16 If[Mod[Length[pointList],3]==1,
17 pointList=AppendTo[pointList,reflectPoint[Last[pointList],
     ↪ pointA]];,
18 If [Mod[Length[pointList],3]==2,
19 pointList=AppendTo[pointList,reflectPoint[Last[pointList],
     ↪ pointB]];,
20 pointList=AppendTo[pointList,reflectPoint[Last[pointList],
     ↪ pointC]];
21 ]
22 ]
23 ,doCtimes];
```

```
24 plot3={Blue,Point[pointList],Black,Line[pointList]};
25
26 (*Export the result*)
27 plot4={plot2,plot3};
28 Show[Graphics[plot4],Axes-> True,AxesStyle->Black]
29 (*----------------*)
30
31 ,{{xK,2,"x-coordinate"},-5,5},{{yK,2,"y-coordinate"
     ↪ },-5,5},{{doCtimes,3,"Number␣of␣movements"},0,10}]
```