Bricks Breaker Project Specification

Created by Nguyen Duy Anh Tuan, last modified on 09 Mar 2015

I. Gameplay – General description

The player moves the bottom bar displayed on the VGA screen by using the buttons on the Zedboard to deflect a bouncing ball to smash the bricks on the upper part of screen. There are several rules in the movements of the ball, how the score is calculated and game specifications that will be justified in the following sections.

You can refer to the YouTube video here to have a basic idea of the game. Even though the project requirement is simpler, you are encouraged to make the game more interesting. Please note that you are required to follow all the requirements specified in this document to implement the basic features of the game. For the extra features, you are free to do what you want.

II. Graphical User Interface

Figure 1 shows the layout of the main screen where the player interacts with to play the game.

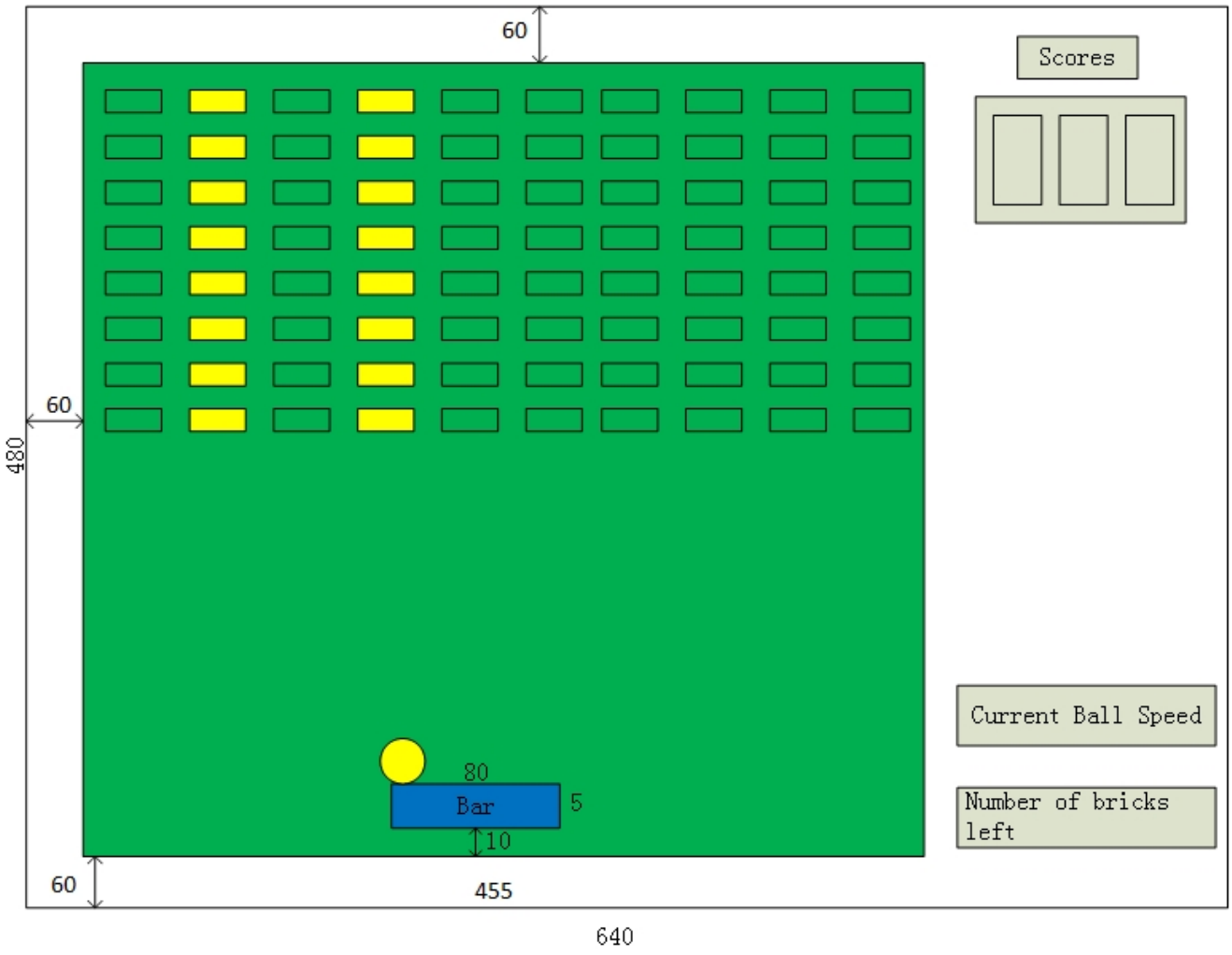

Figure 1 - Main Screen Layout

There are two main zones in this screen which are Brick Zone and Score Zone.

The size of the Brick Zone is 455x480 pixels, with margins from the left, up, and down edges of the monitor to the edges of the game GUI to be 60, 60, and 60 pixels respectively as shown in Figure 1. The brick zone is located at the left part of the screen.

a) The bricks

There are total of 8 rows of brick and each row has 10 bricks as shown in Figure 1.

Figure 2 illustrates the size of the brick and the distances between bricks. You are free to design the pattern and the color of the bricks. During the game, some of the bricks may be able to change to a special color to double the score when they are hit by the ball. The rule will be provided in Section III.

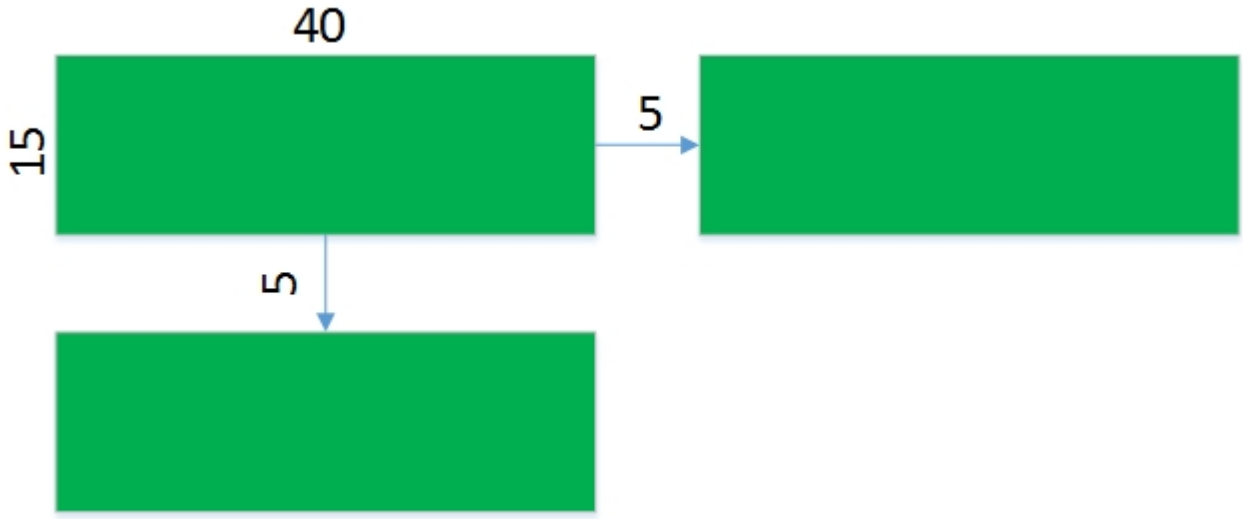

Figure 2 - The size and distances of one brick to neighboring bricks

b) The bar

The bar is located **10 pixels** away from the bottom of the Brick Zone as depicted in Figure 1. The player can only move it horizontally by pressing the buttons on the Zedboard.

There are different types of region on the bar which cause different effects to the ball (angle and speed) depending upon which region the ball hits. The rule will be provided in Section III. The size of the bar and the locations of the regions are described in Figure 3.

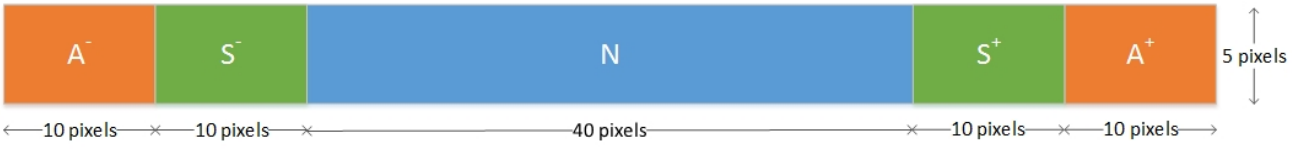

Figure 3 - The size of the bar and the different regions (A, S and N)

c) The ball

The ball is simply a circle with the radius of 7 pixels as presented in Figure 1 and you are free to design the pattern as well as the color of it.

The ball moves along a straight line going through its center. The angle of that line and the direction of the movement are determined when the ball hits the (left, right, upper) sides of the Brick Zone, the bricks and the bar. The details of ball movement will be discussed in Section III.

Apart from the Brick Zone, the remaining space is used for the Score Zone. You are free to design the Score zone but it must have the basic information as listed below:

- The current score (start at 0).
- The elapsed time in second (start counting when the ball first leaves the bar in the beginning of the game).
- The current ball speed in pixel per second.
- The current bricks left.

III. Gameplay – Details

This section provides the detailed rules of the game.

1. Frame rate

The desired slowest frame rate is **25 frame per second** (FPS), i.e. the screen must be updated every 40 milliseconds.

2. The bricks

Each column of brick is managed by one thread which results in 10 separate threads (call brick-threads) are used just to handle the bricks. When all the bricks managed by one thread are broken, the corresponding thread must be stopped. The thread that is utilized to handle the movement of the ball (called ball-thread) sends the instantaneous location of the ball to these threads. Each of these brick-threads will check whether the ball hits any of its brick to inform the ball-thread to re-calculate the movement of the ball.

The brick-threads are also responsible to update the current score. Under normal condition, for every brick that the ball hits, the player gets **1 point**. You should beware of the situation in which the ball can hit more than one brick.

At any point in time, there must be two columns of brick which can be golden bricks. The player will get **2 points** if the ball hits the golden brick. The brick-threads compete to be golden by requesting the imaginary golden-brick shared resource that can only serve two threads at a time. Therefore, the counting semaphore should be used here for the imaginary golden-brick shared resource. The brick-threads release the golden resources **every 10 points** gained by the player. In other words, the brick-threads release the golden resources at the moment when the current score is 10 or 20 or 30, etc.

The bricks that are hit (including the golden bricks) **must be disappeared in the next frame**. If you have any animation to show that the ball is disappearing, the animation must start from the next frame.

3. The movement of the bar

There are 5 usable buttons (BTNL, BTNU, BTNR, BTND and BTNC) on the Zedboard that you can use to control the bar. You can use any button that is suitable.

The bar can only be moved left or right. The bar must be completely inside the Brick Zone, i.e., the left most edge of the bar must be at least 1 pixel away from the left edge of the Brick Zone, similarly for distance between the right most edge of the bar and the right edge of the Brick Zone.

When the button is **pressed and then released immediately** (being pressed **in less than 250 milliseconds**), the bar will move left (right) **25 pixels**.

If the button is **held for more than 250 milliseconds**, the bar will **move continuously** at the speed of **200 pixels per second** until the button is released.

The ball moves along a straight line going through its center (let's call it movement line). The angle of the straight line and the direction of the movement of the ball are determined whenever the ball hits the edges of the Brick Zone (left, right, upper), the bricks and the bar.

a) The reflection of the ball

In case of hitting the (left, right, upper) sides of the Brick Zone, the ball reflects in a way as shown in Figure 4. **The angle of incidence equals the angle of reflection**. After bouncing back from a surface, the speed of the ball may decrease, but in the current project, **the speed is unchanged**. If the ball hits the **N** (**n**ormal) region on the bar, the ball will reflect similarly to the case of hitting the sides of the Brick Zone.

The **initial angle** of the movement line with respect to the horizontal surface of the bar is **90^{o}**. It means the ball will move upward vertically from the bar when the game starts.

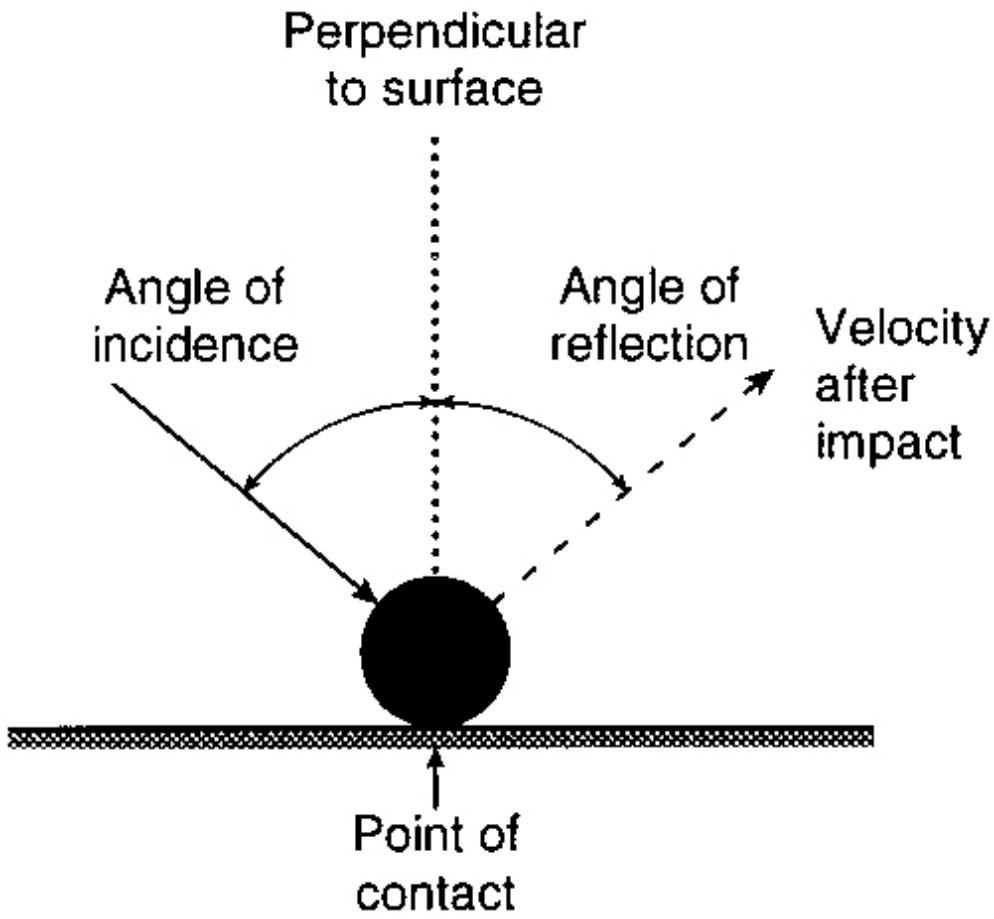

Figure 4 - The bouncing of the ball under normal condition. (Image source: http://ffden-2.phys.uaf.edu/)

In case of hitting the *edges of the bricks*, the reflection surface is the corresponding edge. If the ball collides with *the corner of the brick*, the reflection surface is the imaginary line that makes a 45^{o} angle to the edges of the brick as illustrated in Figure 5. The angle of incidence **equals** the angle of reflection.

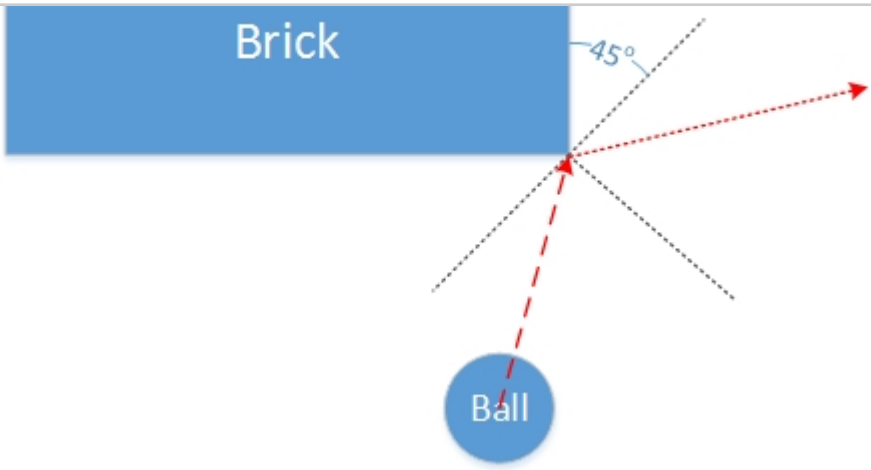

Figure 5 – The ball hits the corner of the brick

If the ball hits the **A$^-$** or **A$^+$** (**A**ngle) region on the bar**, the angle of reflection** will be **decreased** or **increased** by **15^o**respectively as shown in Figure 6. The **minimum** and **maximum angles** of the movement line with respect to the bar are **15^o** and **165^o** respectively.

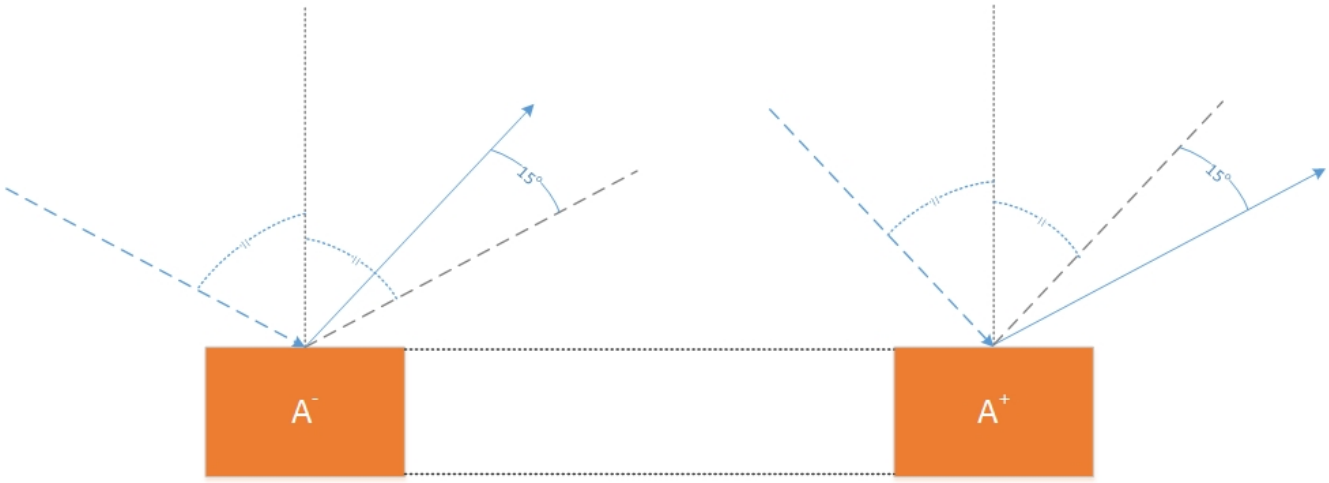

Figure 6 - The changes of the angle of the movement line when the ball hits the A- or A+ region on the bar.

b) The speed of the ball

The speed of the ball is calculated in pixels per second and is determined along the movement line. It is your job to calculate the respective vertical and horizontal speeds of the ball based on the angle of the movement line and the direction of the movement.

The **initial** speed is **250 pixels per second**. Considering the required frame rate of 25 FPS, the ball moves 10 pixels per frame.

In case of *hitting the **S$^-$** (**S**peed) region on the bar*, the speed of the ball will be **decreased** by **100 pixels per second**. If the ball *hits the **S$^+$** region*, the speed will be **increased** by **100 pixels per second**.

Besides, the speed of the ball *automatically increases* by **25 pixels per second** whenever the player gains 10 more points.

The **minimum** and **maximum speeds** of the ball are **50 and 1000 pixels per second**.

5. Game over

- The ball falls off the lower edge of the Brick Zone.
- All bricks are destroyed.

IV. Extra features

You are free and encouraged to implement extra features to the game. Some suggestions:

- The game can be paused/resumed/reset.
- The player has the option to replay the game when it is over.
- The bar will be extended for some duration (maybe 10 seconds) when the ball hits the gem brick.
- The bar regions are color coded.
- Multiplayer.
- Higher FPS.
- Etc.

V. Mid-project requirements

You are supposed to demonstrate the first milestone 2 or 3 weeks after lab 3 (please check the schedule) to show the progress of your project. For this milestone, the following features must be finished:

- The main screen of the game with all the required components in the Brick Zone and Score Zone.
- The ball and the bar can move smoothly, i.e. you should be able to update the screen at 25 FPS.
    - The ball can move up and down continuously at 250 pixels per second.
    - The bar can be controlled by the player while the ball is moving. The requirements specified in Section III.3 for the movement of the bar should be all done.
    - The brick columns should be able to change their color while the ball and the bar are moving.

No labels