

1. Description

The program consists of 2 animating systems: conveyor and cannon while the scene is based on a teapot painting factory. At the beginning of the scene, a teapot comes out of the box, and moves along the conveyor. Above the middle of the conveyor, there is a machine that provides a coloured spotlight shining upon the conveyor; this will be called a colour changing machine. When the grey teapot reaches the middle of the conveyor, its colour will change according to the colour of the spotlight of the machine. The colourful teapot continues its move until the end of the conveyor. It will drop down and rotate following the gravity rule, and will fall into a recycle box. At the point where the teapot starts to rotate, the cannon at the top of the box will turn over 45° , and shoot a ball that has the same colour as the spotlight in the colour changing machine. The ball will collide with the side of the machine and drop down to the side board. Then it will roll down and enter the colour changing machine. After the shot, the cannon will rotate from 45° to 90° , and will rotate back to 0° waiting for the next teapot to fall down. After the teapot dropped into the recycle box, 3 synchronised gears located beside the box will rotate and lift the front cover of the box up. There will be a torus with the same colour as the teapot come out of the box. The gear will rotate backward, and the box's cover will drop down after the torus moves away from the box. The torus will roll along the floor toward the bottom of the started point of the teapot. After the torus reaches that point, the colour of the spotlight of the machine will change respectively according to this list of colour: light coral, yellow green, light sea green, corn flower blue, blue violet, bisque, chocolate, golden, and deep pink. This process will keep repeating. The scene is as shown in Figure 1, and 2 important animations: conveyor and cannon are as shown in Figure 2 and Figure 3.

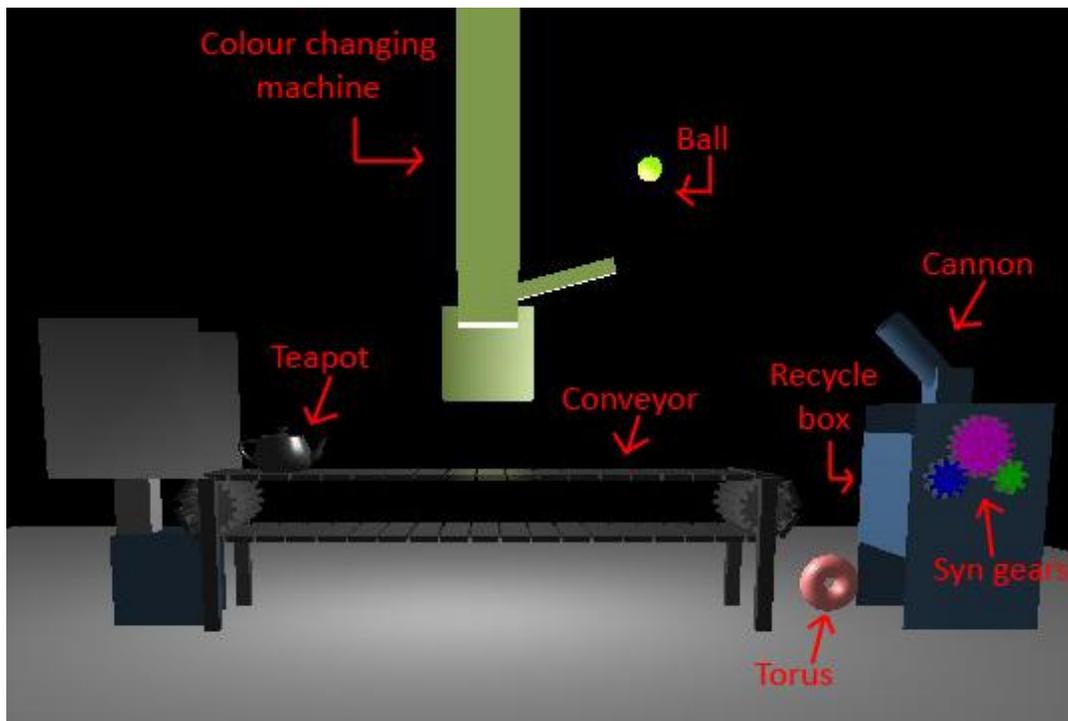


Figure 1: Teapot painting factory with objects labeled.

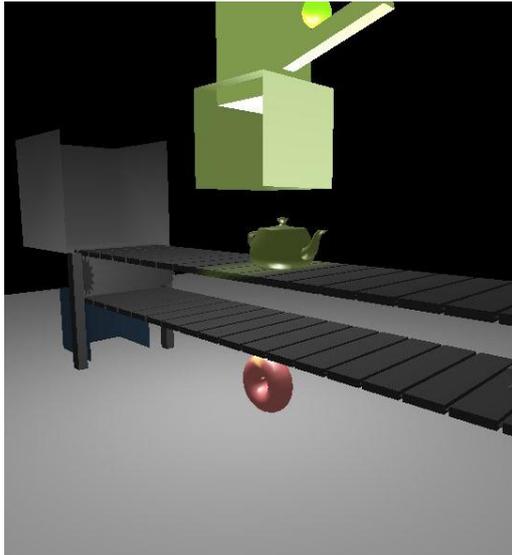


Figure2: Conveyor

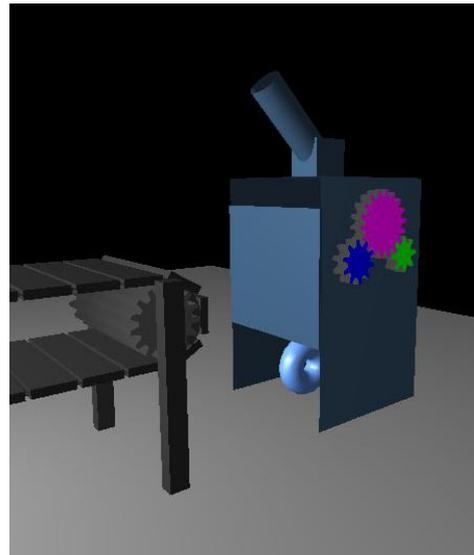


Figure3: Cannon on top of the recycle box

2. Challenges

2.1: Conveyor

Creating a moving conveyor is the hardest part of the project. The conveyor consists of 43 blocks that are 10 units away from each other. These blocks have to move along the conveyor at the same speed, rotate themselves when they are at the end of the conveyor, and continue their move along the conveyor's path. This process has to be repeating, and it is hard for 43 blocks that have different coordinates to move at the same paths because their coordinates change all the time. This problem was solved by storing the x and y coordinates of each block in the arrays. These coordinates will be used to analyse the location of the block by `move_block()` function. This function will check the location of the block, and if the block is at the beginning or the end of the conveyor, rotate it in a clockwise direction by 180° ; otherwise continues moving.

2.2: Spotlight

Normally the spotlight has to be enabled in the initialize function which will run only at the start of the program. However, the spotlight in this program has to change colour all the time, so the new function was created for activating this spotlight, and was used in the display function.

2.3 Projectile motion

It is hard to make a projectile motion looks realistic because the objects in the program do not have specific mass or launching speed. Therefore, the mass was assumed to be 1 and the speed was assumed to be 1.41units per second for the ease of calculation.

3. Mathematical, Algorithmic and Implementation

3.1: Gravity

The teapot moves at a constant speed, and falls down at the end of the conveyor. As it falling down, the speed is still the same according to the law of momentum. The speed of the teapot and the gravity together causes the teapot to drop down in a projectile motion and fall into a recycle box. Because the shape of the teapot is not symmetric, it has to rotate when it is dropping down. An algorithmic of the teapot is as follows:

On the conveyor	-->	At the end of conveyor	-->	After teapot rotate > -5°
Teapot_x++		Teapot_x++ //remain the same speed		Teapot_x+=2 // speed up
		Teapot_rot--		Teapot_y-- Teapot_rot--

3.2: Projectile Motion

The projectile motion occurs when the object moves in the air, and the gravity causes its path to be in parabolic shape. In this motion, the x component of the object is unchanged; however, the y component decreases due to the effect of the gravity.

The projectile motion is applied in the cannon firing. The cannon shoot the ball when it is at 45° at a speed of 1.41 units per second. The x component of the teapot moves 2 units to the left as the time change (ball_x-=2). The y component starts with the increment of 2 units as well but as the ball moving along the path, the increment of y decreases and becomes negative. This method of calculating the y component causes the ball to move as in projectile motion. Figure4 shows the initial x and y component of the ball.

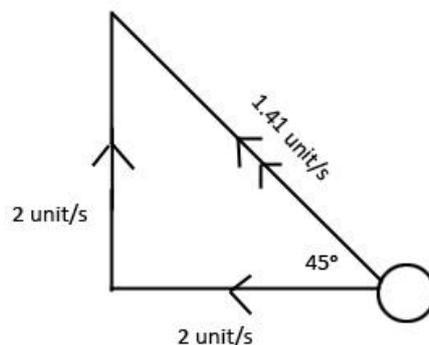


Figure 4: x and y components of the ball

3.3: Collision

The collision of the ball with the side of the colour changing machine is detected by setting a condition to check when the x coordinate of the ball reaches the side of the machine. When this happens, the ball bounces back and drops on the side board by increasing the x coordinate by 0.36 (causes the ball to move to the right) and the increment of y always decreases by 0.03 (cause the ball to drop).

The path of the ball comprises of 4 states: launching, colliding, rolling down, and resetting.

3.4 Door and Synchronised Gears

The door of the recycle box and the synchronised gears beside the box are activated at the same time by the door timer function. The conditions are set to make the door lifts up when the teapot is in the recycle box and moves down when the torus leaves the box. When the door is opening, the blue gear rotates in an anticlockwise direction which synchronise with other 2 gears. These 3 will rotate in an opposite direction when the door is closing. The synchronise gears was modeled by calculating the second angle between 2 gears. The formula is as follows:

$\text{angle_gear2} = -(\text{no. of gear1} * \text{angle_gear1}) / (\text{no. of gear2}).$

3.5: Colour Changing of the Object

There are 4 things in this program that have to vary their colours: spotlight, ball, teapot, and torus. A spotlight and a ball always have the same colour. The colour of the teapot always starts at (0.2, 0.2, 0.2) which is grey, but as the teapot moves along the conveyor, its colour will change into the colour of the spotlight. The torus's colour depends on the colour of that teapot. According to this, 3 arrays were created to store the colour of spotlight, teapot, and torus.

Colour of the spotlight change by copying the colour array into the spotlight_colour array. This spotlight_colour array is used for both spotlight and ball. By using the same method, the colour of the teapot changes by copying the spotlight_colour array into the teapot_colour array. The timer was used to make the colour of the teapot gradually changes instead of suddenly change. After this the teapot_colour will be copied onto the torus_colour array.

The colour code in the colour array is obtained from the decimal code in the colour code website. The decimal numbers are converted by dividing by 255 and times 1.

3.6: Reset Procedure

The objects used in this program such as teapot, torus, and ball will be reused as it reaches its final state. This was done in order to minimise the memory used of the program. The teapot resets itself back to its initial position, and changes its colour back to grey when the the teapot falls into the recycle box and after the door moves up. The ball from the cannon resets back to be inside the cannon after it is fired and hit the middle of the colour changing machine. The torus resets when it reaches the end of its path.

3.7: Camera

The camera is set to locate at $ex = 0$, $ey = 70$, and $ez = 400$ and look at $lx = ex - r \cdot \sin(\text{angle_camera})$, $ly = 10$, $lz = ez - r \cdot \cos(\text{angle_camera})$ at the initial state. The angle of the camera starts at 0° so the initial $lx = 0$, $lz = 10$, and $lz = ez - r$. lz should be 0 at the start so the radius r is set to equal to 400. Figure 4 shows the setting of the camera.

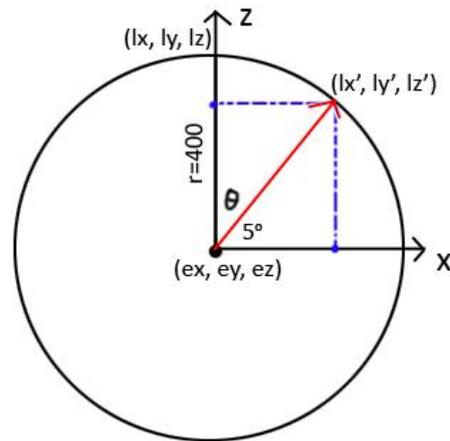


Figure 4: Diagram of Camera

When the camera zoom in, $ex \pm 4\sin\theta$ and $ez \pm 4\cos\theta$; this will shift the camera 4 units towards its path. In the opposite way, the camera zooming out will have $ex \mp 4\sin\theta$ and $ez \mp 4\cos\theta$.

When the camera rotates in an anticlockwise direction, 5° will be added to the angle of the camera, and 5° will be deducted if the camera rotates in a clockwise direction. The angle of the camera will be used to calculate lx and lz . The `gluLookAt` function is used as follows:

```
gluLookAt (ex, ey, ez, ex-r*sin(angle_camera), ly, ez-r*cos(angle_camera), 0.0, 1.0, 0.0);
```

4. Control Function

There is only 1 control function for this program which is a camera control as in the specification.

- Up arrow key: move the camera forward in the current direction
- Down arrow key: move the camera backward in the current direction, and keep looking at the same direction
- Left arrow key: rotate the camera 5° to the left.
- Right arrow key: rotate the camera 5° to the right.

Reference

Hyper Physics (n.d.). Retrieved from <http://hyperphysics.phy-astr.gsu.edu/hbase/traj.html#tra4>

Rapid Table (2013). Retrieved from http://www.rapidtables.com/web/color/RGB_Color.htm

The Physics Class room (2013) Retrieved from
<http://www.physicsclassroom.com/Class/momentum/u4l1a.cfm>