SCHOOL OF COMPUTER SCIENCES, UNIVERSITI SAINS MALAYSIA 11800 USM, PENANG, MALAYSIA

CPC354 - COMPUTER GRAPHICS AND VISUALIZATION

Assignment 1 : 3D Ident Program

LECTURER: PROFESOR MADYA DR. AHMAD SUFRIL AZLAN BIN MOHAMED

Prepared By: **GROUP 3**

Name	Matric Number	E-mail Address
Tan Jun Lin	160989	tanjunlin0215@student.usm.my
Peh Jia Jin	161059	jiajin1229@student.usm.my
Jason Leong Sheng Jun	155891	jasonleong@student.usm.my
Teh Wen Jie	159769	wenjie02@student.usm.my

Table of Content

1.0 Team Description

Member	Detailed Works
Tan Jun Lin	**1. Animation of TV ident (JavaScript coding)** **2. Proposed Method and Conclusion** **3. Overall Report Checking**
Peh Jia Jin	**1. Enhanced Dialogue Design (CSS)** **2. Animation of TV ident (JavaScript coding)** **3. Literature Review**
Jason Leong Sheng Jun	**1. Dialogue Design (HTML coding and CSS)** **2. Perform system testing**
Teh Wen Jie	**1. Animation of TV ident (JavaScript coding)** **2. Introduction and Proposed Method** **3. Video creating for Demo**

Table 1: Detailed Work for each of the member

In this project, Jason Leong develops the front-end part of the dialogue interface, while Teh Wen Jie will help him enhance it to make it more user friendly. On the other hand, Tan Jun Lin and Peh Jia Jin utilize the front-end design, which includes buttons, sliders, and so on, for the purpose of implementing all event listeners on the canvas for the back-end part of the TV logo animation. In the end, all of us sit down to discuss the progress of the system and divide up the task according to the sections of the technical report, which include Introduction, Literature Review, Proposed Solution, and Conclusion. Before submitting to the e-learning portal, Jason Leong will take on the responsibility of conducting system testing for the 3D Ident program to make sure all of the functions are working. After testing of the system, Teh Wen Jie will make a video to perform the system demo of our TV ident. On the other hand, Tan Jun Lin will check the overall report to make sure all the parts match the rubric.

2.0 Introduction

In the realm of computer graphics and visualization, a TV ident or TV identification refers to the short visual sequence or animation that appears in the break between TV programs of about 3 to 7 seconds. The main purpose of a TV ident is to brand, promote, and advertise a TV channel. It is similar to the visual signature of the channel, and it is typically presented before or after a TV program, during commercial breaks, or as part of station promotions. These visuals frequently include the channel's logo, colors, and other distinguishing characteristics, creating a distinct and recognisable brand image. A TV ident is essentially an animated logo used for channel branding. It can range from a simple moving trademark to a more elaborate animation.

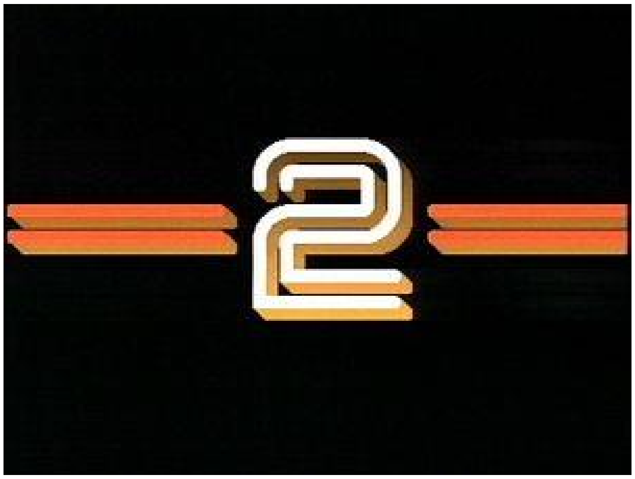
By comparison, TV idents are more effective, allowing audiences to remember and become familiar with it than the regular advertisements. Computer graphics and visualization are key contributors to creating these TV idents. Branding recognition for TV ident to establish an identity, comes part of computer graphics design where graphics design especially animation plays a crucial role in creating 3D logo of a brief video to convey to the audiences. With the right period of animation, it can familiarize the TV branding itself with the audiences to remember at the back of their mind. This is a form of advertising only much simpler and in terms of creating a TV identity the factors surrounding how the animation plays out are very important to allow audiences to capture the moment.

3.0 Literature Review

In this report, we will examine three TV ident case studies designed by Red Bee Media: BBC 2 'Computer Generated 2', BBC THREE, and BBC 2 CHRISTMAS.Driven by the advancement of computer technology, the development of CGI technology has a great impact on the visual aesthetics of TV programs. This evolution gives creators greater creative expression and control. By analyzing these three TV commercials created at different points in time, we can see the transformative journey of CGI and its impact on visual storytelling.

3.1 BBC 2 ‘Computer Generated 2’ ident

In the year 1979, the ‘Computer Generated 2’ ident was created and used by BBC 2
(BBC 2, 2022).



In 1979, the world's first computer-generated television ident was revolutionary at the time and was a groundbreaking achievement. During this period, computers operated on a command line and the process consisted of pixel-by-pixel programming of images or logos.Skilled programmers needed to draw on the expertise of computer graphics professionals to craft elaborate television ident. The design was processed through a large supercomputer on a render farm, a computer system specifically designed to render CGI.

3.2 BBC THREE

In the year 2008, Red Bee Media recreated some TV idents for BBC THREE (BBC THREE, 2008).

BBC
three
Subtitles

Red Bee Media chose to create computer-generated TV idents to improve quality compared to previous versions. They aimed for better surface texture details, opting for a glossy finish in their TV ident. The software they used allowed them to render different surface finishes by combining sampled and coded texture maps onto animated elements. This contributed to a more refined and detailed visual presentation.

3.3 BBC 2 CHRISTMAS

In 2011, Red Bee Media conceived the idea to redesign a Christmas Special TV ident for BBC 2, known as BBC 2 Christmas.

Subtitles
BBC
TWO

Unlike BBC THREE, which relies on CGI entirely for its TV ident. BBC 2 stands out for its emphasis on physical elements. BBC 2 is known for its distinctive features such as metal objects covered in fabric or fluffy toys and often seen with an aperture through a tent (Macdonald, 2015). The BBC 2 Christmas ident maintains a tangible quality and exhibiting traces of human touch. Instead of creating models in CGI, BBC 2 opts to photograph physical miniature models adding a human warmth to the TV ident. The animated parts of BBC 2 CHRISTMAS were initially captured using live-action puppetry and later refined through digital editing. This emulation of real puppetry enhances the realistic appearance of CGI elements while retaining the flexibility of digital production.

Model of elements of BBC 2 CHRISTMAS (Macdonald, 2015)

4.0 Proposed Solution

In this assignment, we are required to produced a TV ident that met the following requirements:

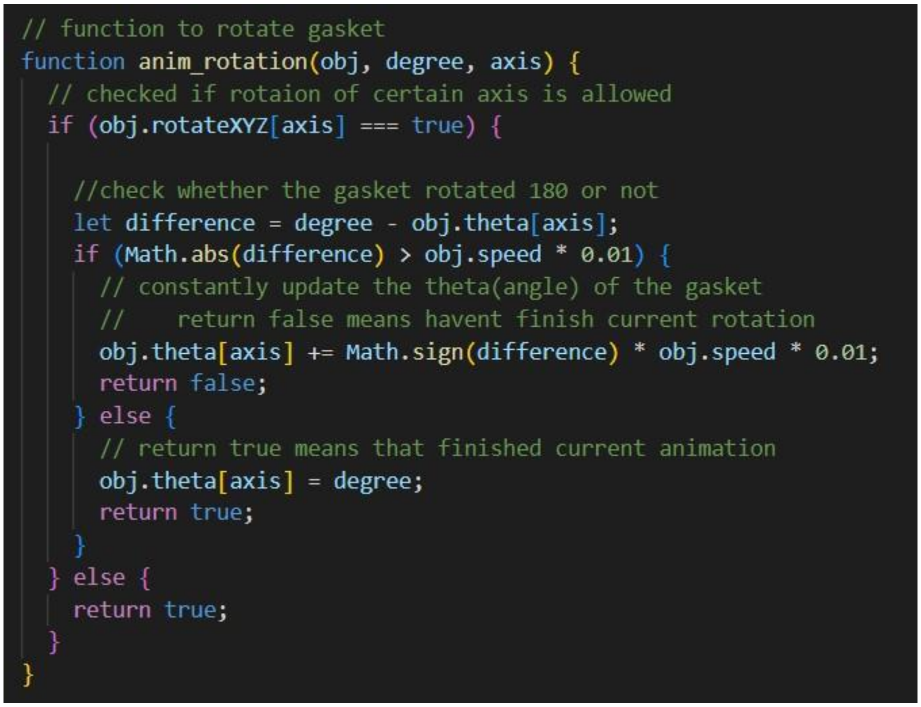
1. Begin by positioning the 3D gasket in the center of the screen.
2. Rotate it to the right of 180 degrees.
3. Rotate back to the left of 360 degrees.
4. Enlarge to an appropriate size based on the scaleFactor and return to its own size.
5. Move around the screen (loop) until a button is pressed.

We have various functions for rotation, expansion and movement in order to fulfill the requirements mentioned above. In addition, a dialogue with a user-friendly interface has been created as well for the users to manipulate the input. In the dialogue, there are 2 different menu controls, which are Gasket Properties and Animation Properties. The Gasket Properties allows users to manipulate the 3D gasket. Users can manipulate input such as the speed of rotation, color of the 3D gasket, number of subdivisions, scale, and scale factor.

Users are able to change the speed of animation from 100% to 900%. The four circles of the color picker are the side of the gasket. Each circle represents each side of the gasket which are Left, Bottom, Right and Back. When the users click any of the circles, a RGBA custom color picker will be shown and the color selected will be previewed in the circle and also the canvas. Besides that, users are allowed to choose up to 5 subdivisions and the least subdivision is 0. For the scale of the 3D gasket, we provided a range of 1 to 3 for the users to manipulate. Lastly, the scale factor is able to change from the range of 0 to 3.

In the Animation Properties, users are able to alter the rotation angle from 0 to 360 degrees. Besides, there are two options provided as well which are rotation about x-axis and rotation about y-axis. Users can choose either one of them, none of them or both of them. Lastly, the 3D gasket transition style can be decided by the users too. We provided 5 kinds of transition styles which are Random, Rotating, Dancing, Flipping, and Paralysing. A Play and Restart button are shown in the dialogue as well, users are able to start whenever they want to and stop the animation by clicking the Stop button which will be displayed after the Play button is clicked.

Rotation

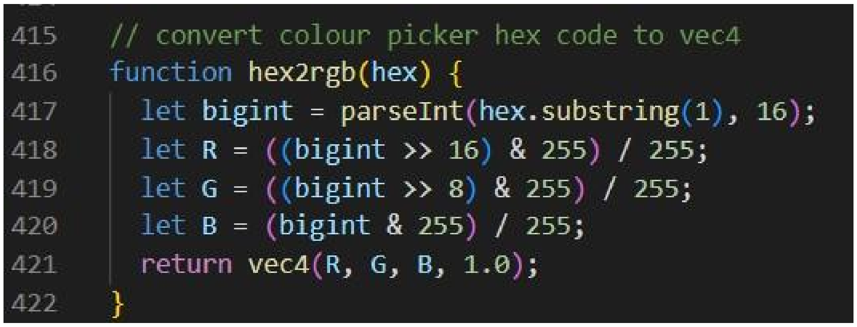


```
// function to rotate gasket
function anim_rotation(obj, degree, axis) {
  // checked if rotaion of certain axis is allowed
  if (obj.rotateXYZ[axis] === true) {

    //check whether the gasket rotated 180 or not
    let difference = degree - obj.theta[axis];
    if (Math.abs(difference) > obj.speed * 0.01) {
      // constantly update the theta(angle) of the gasket
      //    return false means havent finish current rotation
      obj.theta[axis] += Math.sign(difference) * obj.speed * 0.01;
      return false;
    } else {
      // return true means that finished current animation
      obj.theta[axis] = degree;
      return true;
    }
  } else {
    return true;
  }
}
```

Source code above refers to a function for rotation. It will examine and adjust the theta of different axis to allow the gasket to rotate. The 360 degrees animation will be divided into two 180 degrees animation. If the gasket has not yet completed the animation, it will constantly update the theta of the gasket and re-render the gasket.

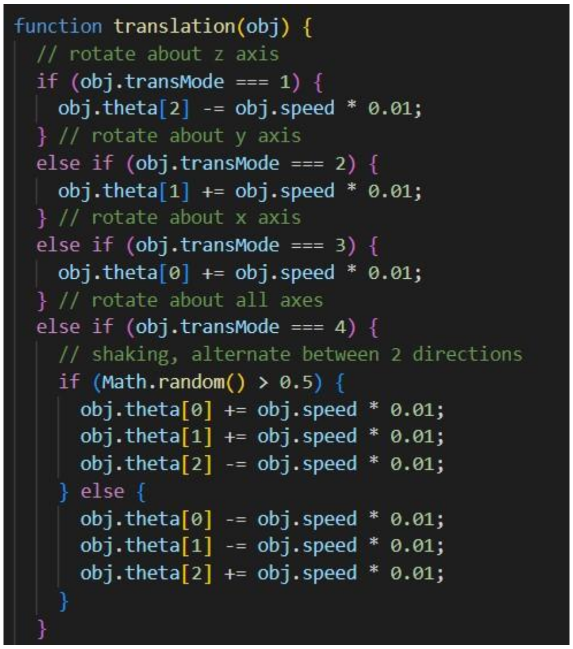
Hex to vec4 (Color)



```
// convert colour picker hex code to vec4
function hex2rgb(hex) {
  let bigint = parseInt(hex.substring(1), 16);
  let R = ((bigint >> 16) & 255) / 255;
  let G = ((bigint >> 8) & 255) / 255;
  let B = (bigint & 255) / 255;
  return vec4(R, G, B, 1.0);
}
```

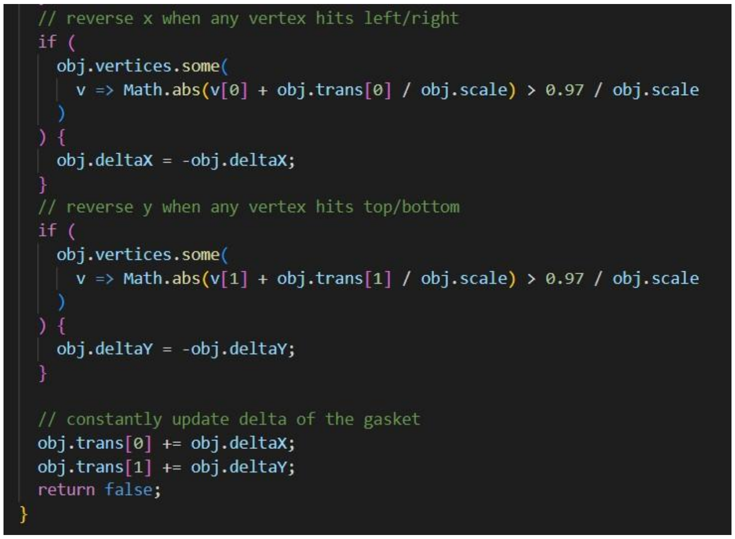
Source code above shows the function that converts the color picker hex code to vec4 floating decimal numbers. In the source code, we used two operators to get the corresponding converted floating decimal numbers which are the bitwise right shift operator (>>) and bitwise AND operator (&).

Moving



```
function translation(obj) {
  // rotate about z axis
  if (obj.transMode === 1) {
    obj.theta[2] -= obj.speed * 0.01;
  } // rotate about y axis
  else if (obj.transMode === 2) {
    obj.theta[1] += obj.speed * 0.01;
  } // rotate about x axis
  else if (obj.transMode === 3) {
    obj.theta[0] += obj.speed * 0.01;
  } // rotate about all axes
  else if (obj.transMode === 4) {
    // shaking, alternate between 2 directions
    if (Math.random() > 0.5) {
      obj.theta[0] += obj.speed * 0.01;
      obj.theta[1] += obj.speed * 0.01;
      obj.theta[2] -= obj.speed * 0.01;
    } else {
      obj.theta[0] -= obj.speed * 0.01;
      obj.theta[1] -= obj.speed * 0.01;
      obj.theta[2] += obj.speed * 0.01;
    }
  }
```

This function is divided into two sections. The first section is when the gasket is moving while the second part is when the gasket hits the canvas wall and bounces off, continuing to move. Source code above deals with the initial aspect of animation, specifically focusing on the rotation of the gasket along the x-axis, y-axis, and z-axis. Additionally, shaking behavior is added as well. Each behavior results in changes of the gasket based on the specified axis.

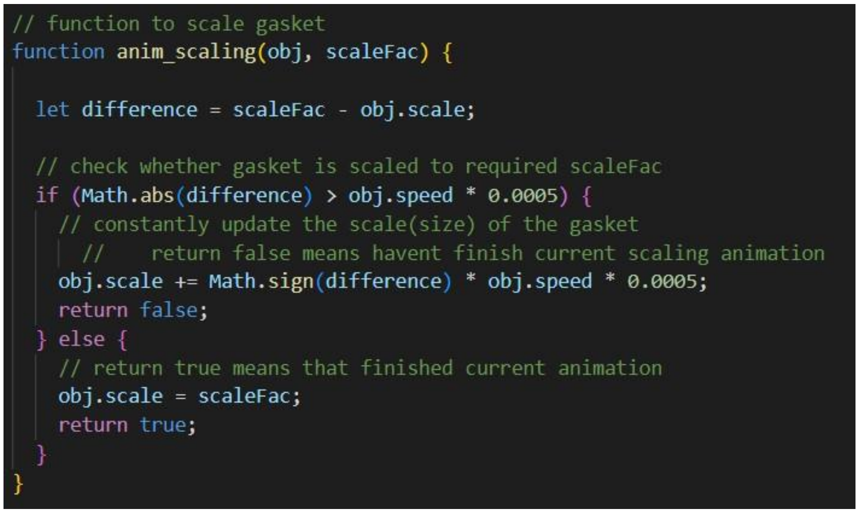


```
  // reverse x when any vertex hits left/right
  if (
    obj.vertices.some(
      v => Math.abs(v[0] + obj.trans[0] / obj.scale) > 0.97 / obj.scale
    )
  ) {
    obj.deltaX = -obj.deltaX;
  }
  // reverse y when any vertex hits top/bottom
  if (
    obj.vertices.some(
      v => Math.abs(v[1] + obj.trans[1] / obj.scale) > 0.97 / obj.scale
    )
  ) {
    obj.deltaY = -obj.deltaY;
  }

  // constantly update delta of the gasket
  obj.trans[0] += obj.deltaX;
  obj.trans[1] += obj.deltaY;
  return false;
}
```

The source code refers to the second section. This section shows how the gasket reacts upon hitting the wall; it exhibits a bouncing-off behavior. This function will consistently return false as the animation will not end until the user initiates the restart by clicking the Restart button.

Scaling



```
// function to scale gasket
function anim_scaling(obj, scaleFac) {

  let difference = scaleFac - obj.scale;

  // check whether gasket is scaled to required scaleFac
  if (Math.abs(difference) > obj.speed * 0.0005) {
    // constantly update the scale(size) of the gasket
      //    return false means havent finish current scaling animation
    obj.scale += Math.sign(difference) * obj.speed * 0.0005;
    return false;
  } else {
    // return true means that finished current animation
    obj.scale = scaleFac;
    return true;
  }
}
```

In the source code presented above, there is a function responsible for scaling the gasket, altering its size to either bigger or smaller. If the gasket hasn't reached the expected scale, the function increases its size and re-renders it on the canvas. The function returns false as long as the animation is ongoing but switches to true upon completion.

5.0 Conclusion

In this project, our program has successfully completed all the criteria, including designing a user-friendly dialogue and animation for the TV-ident. We included all the features required by the marking criteria, such as animation speed, color picker, and number of subdivisions. Beside that, we have added extra features not mentioned in the specification, including scaling, zoom factor, rotation angle in degree, x-axis and y-axis rotation, and transition styles. To make it more user-friendly, we provided buttons with textbox display, slider with bubble display, and a transparency circular exteriors for each of the sections to ensure that their appearance looks modern and beautiful. To make this program more useful, we also included Responsive Web Design which will push the properties box to the bottom of canvas if a certain resolution is reached such as below 1600px.

For the animation of TV ident, we also successfully follow all the criteria that animation of the TV ident that has been mentioned before:

1. *Begin by positioning in the center of the screen.*
2. *Rotate it to the right of 180 degrees.*
3. *Rotate back to the left of 360 degrees.*
4. *Enlarge to an appropriate size based on the scaleFactor and return to its own size.*
5. *Move around the screen (loop) until a button is pressed.*

The functions that we used in order to create these TV idents as follows:

- hex2rgb that converts the hex code of a color picked to vec4 floating decimal numbers.
- Rotation Function which checks and changes the theta of different axis for the gasket to rotate.
- Scaling Function which scales the gasket.
- Moving Function which ensures the gasket is moving and bounce off when it hits the wall.

With the active cooperation of each of us, we have successfully produced a TV-ident program that meets all the criteria with extra features within the specified time.

6.0 References

1. (PDF) television branding in Ghana: A study of TV idents - researchgate. (n.d.). https://www.researchgate.net/publication/312939174_TELEVISION_BRANDING_IN_GHANA_A_STUDY_OF_TV_IDENTS
2. Wikimedia Foundation. (2023a, July 26). *BBC Two "Computer generated 2" Ident*. Wikipedia. https://en.wikipedia.org/wiki/BBC_Two_%27Computer_Generated_2%27_ident#References
3. Boden, L. (2018, February 7). *BBC3 ident*. Vimeo. https://vimeo.com/14636808