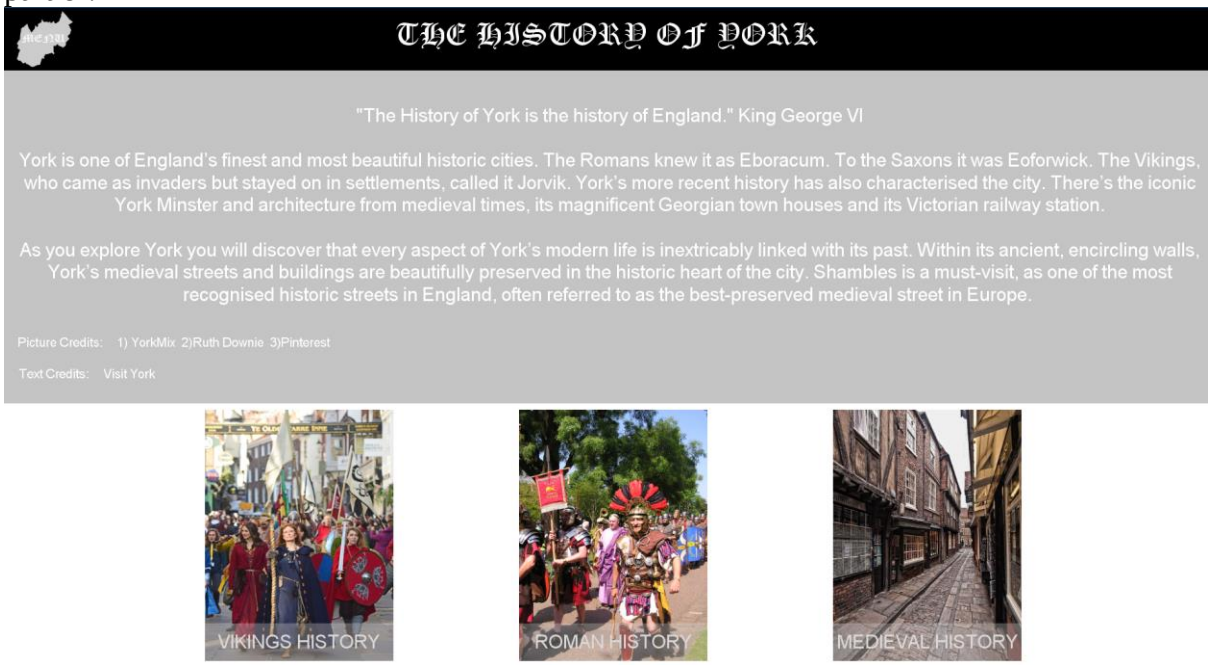
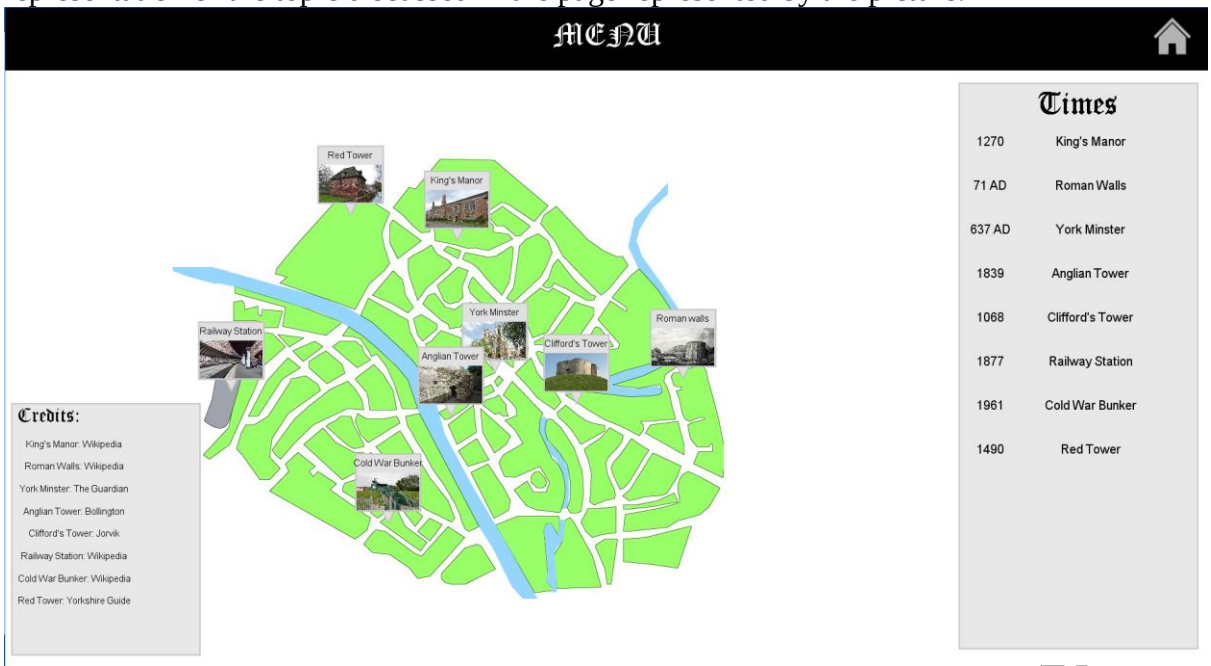
User manual

I have decided to develop the first option for the assessment, the page-based museum exhibit. The exhibit is about the history of York. The project starts by showing a few pictures of York in the format of a slideshow. By pressing next, the user will be on the main page/ the home page where they can find information about York and some eras of the past that York took part of.



By pressing on either option available at the bottom of the page, the user can find more about that specific topic. In the top-left corner to user can go to the menu, which is gives a visual representation of the topic discussed in the page represented by the picture.



All the pages have the option to make the picture bigger by pressing this symbol . Also there are buttons on each page that give the user the possibility of moving around the exhibit without having to return to the menu. Some pages have also the option to find out more information about the exhibit just by clicking on the image, in case the user does not understand how the page works there is a bubble at the bottom of the page that, when user clicks on, lets the user know that they can press on the image and they will be redirected to a new page. That page is not available in the menu.

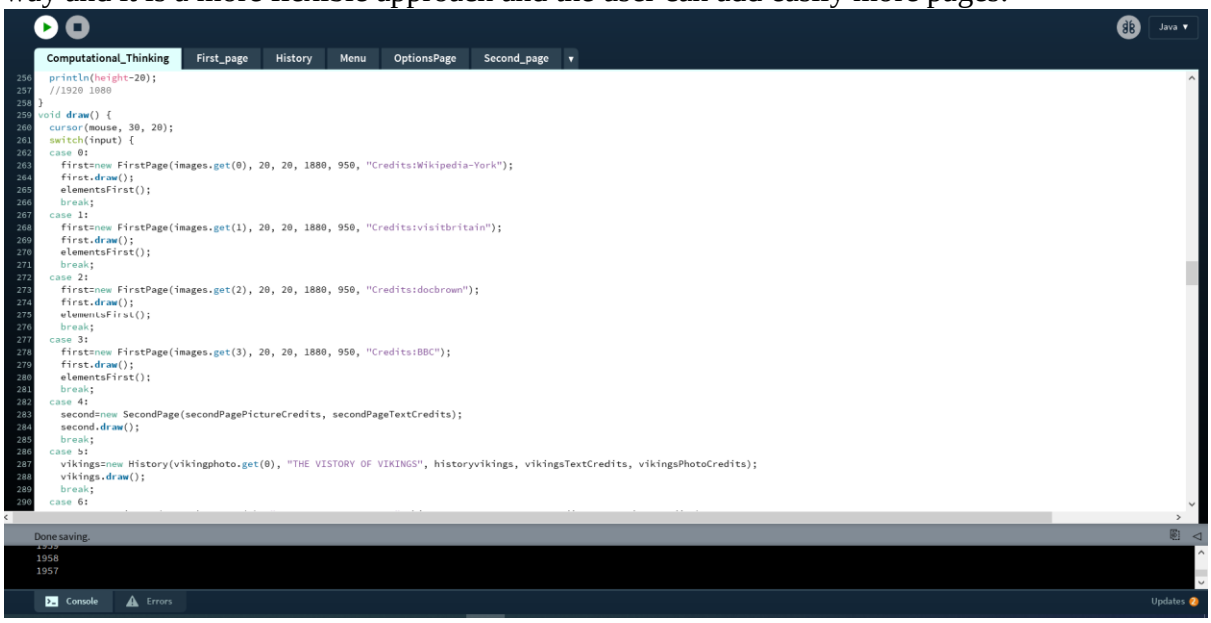
Major features

The major features are: the slideshow at the beginning of the program, the timer, the menu,

the slideshow that is visible when the user clicks on the picture inside the page representing King's Manor and the music presented inside the York Minster page, the personalized mouse icon, the hover over the buttons, the functionality of the buttons and the method I have approached for text by initializing it in external files and the fonts I have used.

Description of the code

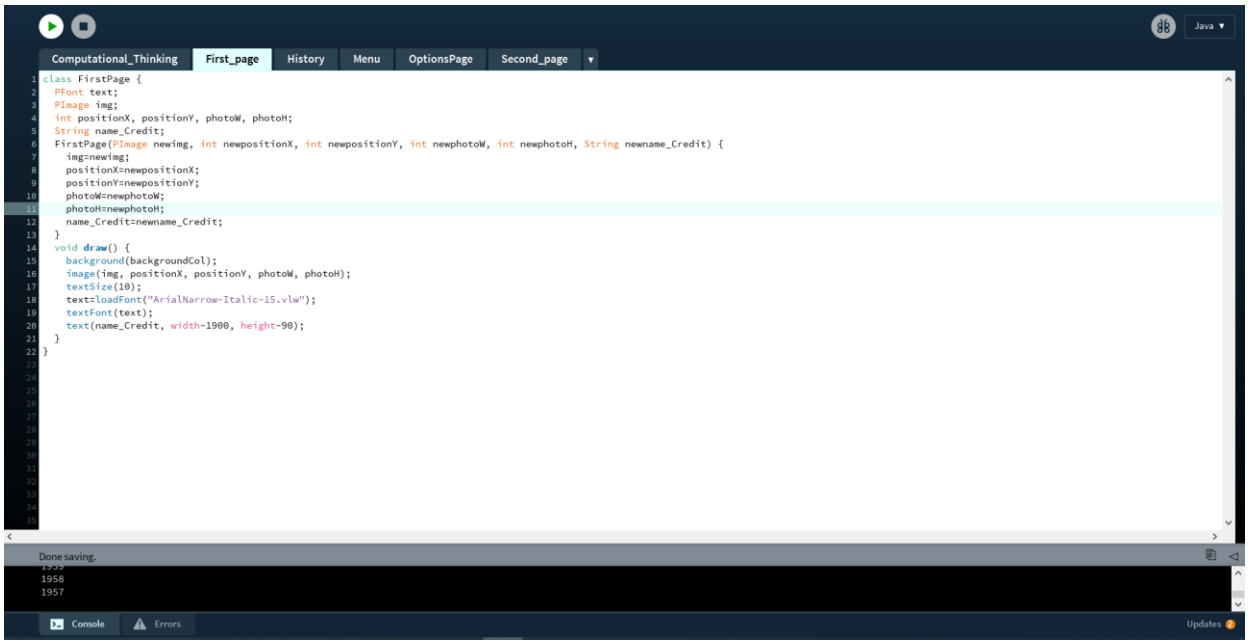
I have used switch statements for moving from a page to another because it is more elegant this way and it is a more flexible approach and the user can add easily more pages.



The code has 5 classes: FirstPage, History, Menu, OptionsPage and SecondPage. The class FirstPage has one object that is called "first", the purpose of this class is to showcase the slideshow of the pictures of York. Inside the class FirstPage, the program runs the correct image that is inside the ArrayList "images" and the correct credit. Although, the variables positionX, positionY, photoW, photoH are constants, I have decided to make the page more flexible for future changes and to include these variables inside the constructor. As seen in the previous picture each slide is inside a case. The case is changed depending on the frameCount as shown below and as long as the user hasn't pressed the button next.

```
println(timedout);
if (frameCount %30==0 && count<images.size() && !pressed) {
  input=count;
  count++;
}
```

Also the reason why the button next that is inside elementsFirst() function is inside the draw function in the main page of the code is because I wanted it to be more intuitive and to increase its strokeWeight when the mouse is over it, and the only way this could be possible is if it was inside the draw function.

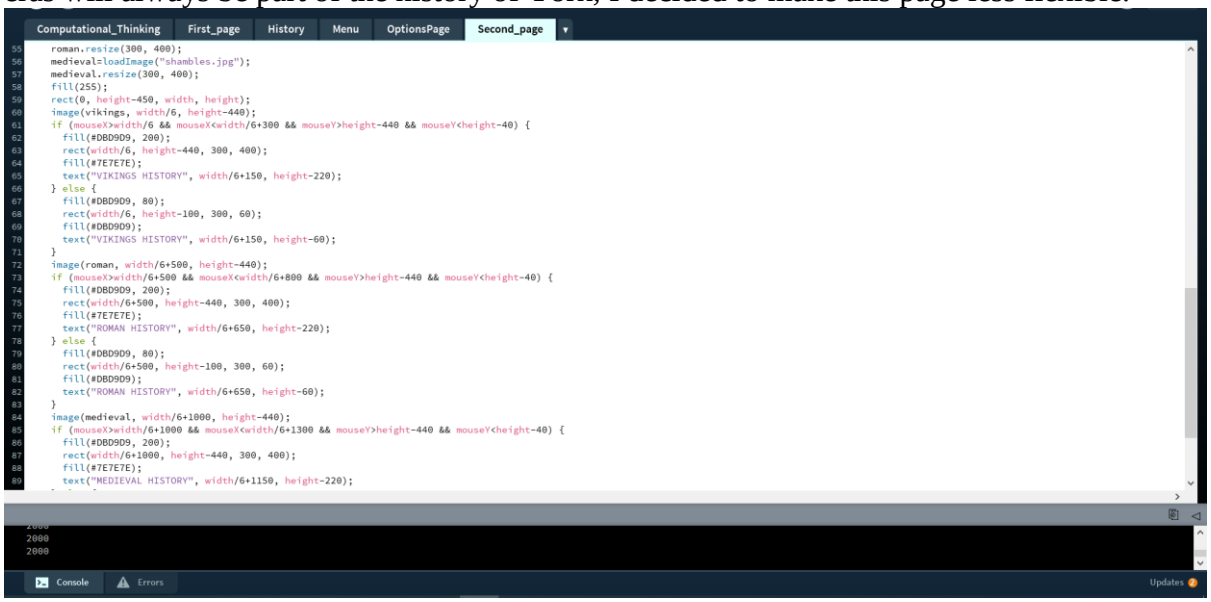


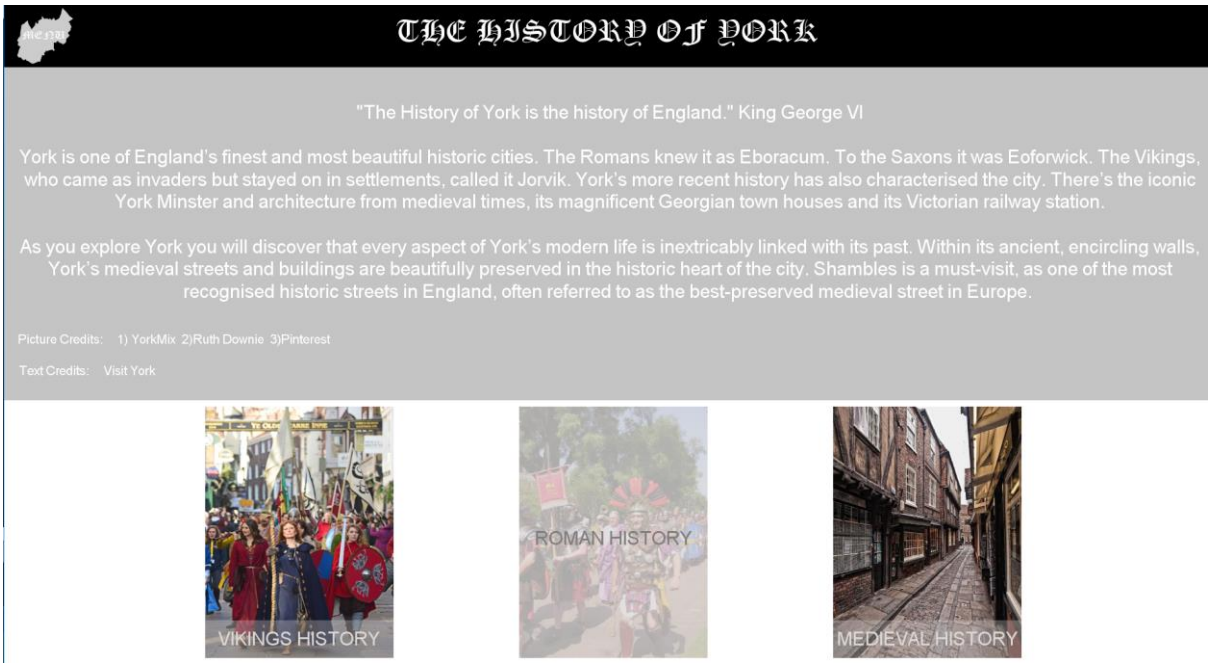
Another feature I have implemented is the mouse that has an icon which symbolizes York.

```
cursor(mouse, 30, 20);
```

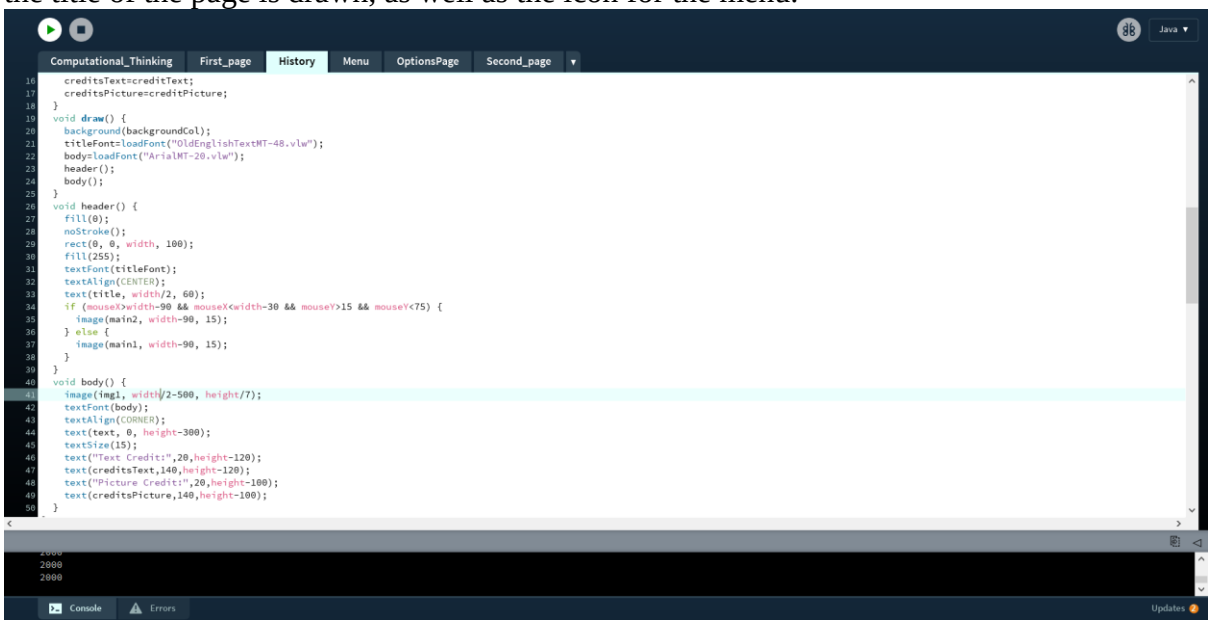
This part of the code is inside the draw() function in order to be always visible and updated with the new positions of mouseX and mouseY.

Inside the class SecondPage I decided to make the design more welcoming by ading a hover over the image-buttons when the mouse is over one of the pictures . Given the fact that those eras will always be part of the history of York, I decided to make this page less flexible.





The class “History” works like a template, it has the same design for every era. In the header the title of the page is drawn, as well as the icon for the menu.



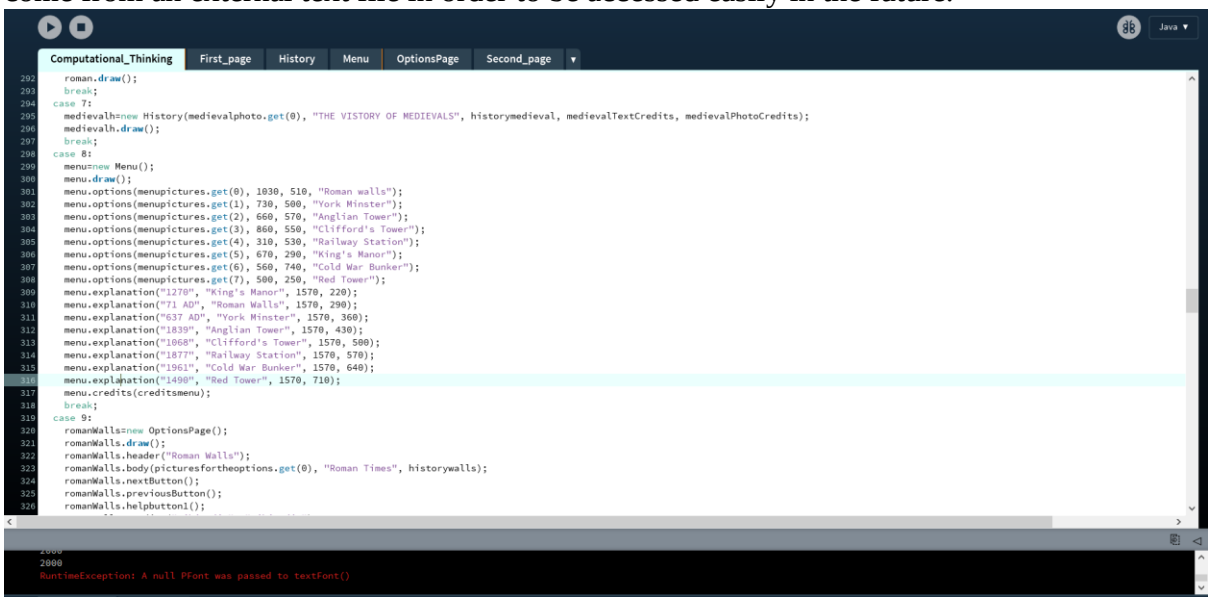
This is what is in the draw function, every picture is inside a different ArrayList in order to make the code more flexible for future changes, such as a slideshow, the text comes from external files.

```
case 5:
  vikings=new History(vikingphoto.get(0), "THE VISTORY OF VIKINGS", historyvikings, vikingsTextCredits, vikingsPhotoCredits);
  vikings.draw();
  break;
case 6:
  roman=new History(romanphoto.get(0), "THE VISTORY OF ROMANS", historyromans, romanTextCredits, romanPhotoCredits);
  roman.draw();
  break;
case 7:
  medievalh=new History(medievalphoto.get(0), "THE VISTORY OF MEDIEVALS", historymedieval, medievalTextCredits, medievalPhotoCredits);
  medievalh.draw();
  break;
```

This is how I initialized the text files, in order to make the text fit in the box, I had the pass the words that were out of the box to a new line, this is the reason why join(name of the String,”\n”) is there.

```
String[] history=loadStrings("historyofvikings.txt");
historyvikings=join(history, "\n");
String[] historyroman=loadStrings("romanhistory.txt");
historyromans=join(historyroman, "\n");
String[] medievaltext=loadStrings("medievalhistory.txt");
historymedieval=join(medievaltext, "\n");
```

For the menu, I have implemented a more ambitious idea. The menu is structured in the shape of a map that is a close copy of the map of York as showed in the picture above in the section user manual. The class “Menu” has only one object and the functions “options” and “explanation” are called 8 times for each option available in the menu in order to make efficient use of the class “Menu” while using only one object in the same case. As shown below, the pictures are inside the ArrayList “menupictures”, while the credits for the pictures come from an external text file in order to be accessed easily in the future.



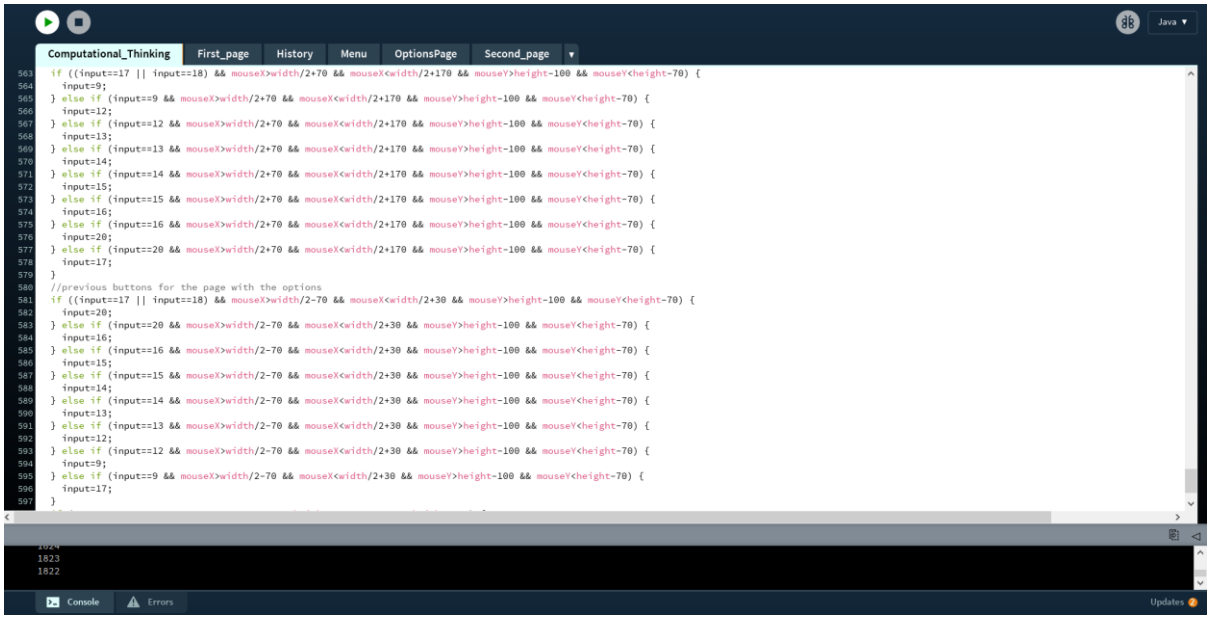
The class “OptionsPage” works as a template for the pages where the options will be showcased. All the options have this functions: header(), body(), nextButton(), previousButton(), credits(). However some of the pages, more exactly, romanWalls and kingsManor have extensions of the pages, secret pages that open when the user clicks on the picture, this pages don’t have the buttons previous and next because they are not part of the main menu. The object yorkminsterpage has a special function that is for sound and also for the credit of the sound. Therefor, the reason why the class “OptionsPage” does not have anything inside the constructor is because not all the pages have the same format, due to the same reason the functions: body(), nextButton(), previousButton(), credits() are not inside the draw() function that is inside the class, however the header() function could be inside the draw() function, but I decided to put it outside in case I wanted to modify some things. While the extensions of the pages romanWalls and kingsManor have some elements that are particular to those two pages, the back button.

```
void getBack() {
  strokeWeight(2);
  fill(colorbutton);
  stroke(#BFBDBD);
  rect(20, height-100, 100, 50);
  fill(255);
  textFont(backfont);
  textSize(25);
  text("BACK", 70, height-65);
}
```

In regards to how the buttons are able to function inside the code without leading to pages that do not exist is due to the code that is inside mousePressed. The code is restrictive in the sense that if the user clicks on a predetermined spot on the screen which showcases a button, the page that will be opened is in direct connection to the page the user is on, due to the fact that the input of that page is also taken into consideration. The selection of the pages is possible due to the switch statements that are changed in consideration to the elements mentioned earlier, the position of mouseX and mouseY and the input representing the page the user is on.

```
  //transfer from the menu to the option page (York Minster)
  if (input==8 && mouseX>width/2-230 && mouseX<width/2-230+100 && mouseY>height/2-40 && mouseY<height/2-40+60) {
    input=12;
  }
```

Another way in which the user can travel from a page to another is by using the buttons for previous and next.  This method is possible because of the if statements present in the function mousePressed(). The reason why the buttons are working is because I have used else if and in this way the changes of the input are also dependent on the page the user is on.



Some of the major features are showcased in relation to the class “OptionsPage”, as mentioned above, I have implemented a slideshow inside the page representing King’s Manor. Depending on the position of the mouse, the width of the canvas is divided in 3, while the mouse is moving, the picture, credit and the filling of the circles below the image will change depending on the position of mouseX. Inside the class OptionsPage, the function called “slideshow”, is changing the filling inside the circles, as well as the correct picture inside the ArrayList in regards to the slidenumber that is passed from the draw function that is inside “Computational_Thinking”.

Inside “Computational_Thinking”

```
  case 19:
    kingsManorSlide=new OptionsPage();
    kingsManorSlide.draw();
    kingsManorSlide.header("King's Manor");
    kingsManorSlide.slideshow(pictureslideshow, slide, "University of York", "Wikipedia", "GPSMYCITY");
    if (mouseX>0 && mouseX<1920/3) {
      slide=0;
    } else if (mouseX>width/3 && mouseX<(width/3)*2) {
      slide=1;
    } else if (mouseX>(width/3)*2 && mouseX<width) {
      slide=2;
    }
    kingsManorSlide.getBack();
    break;
```

Inside the class OptionsPage

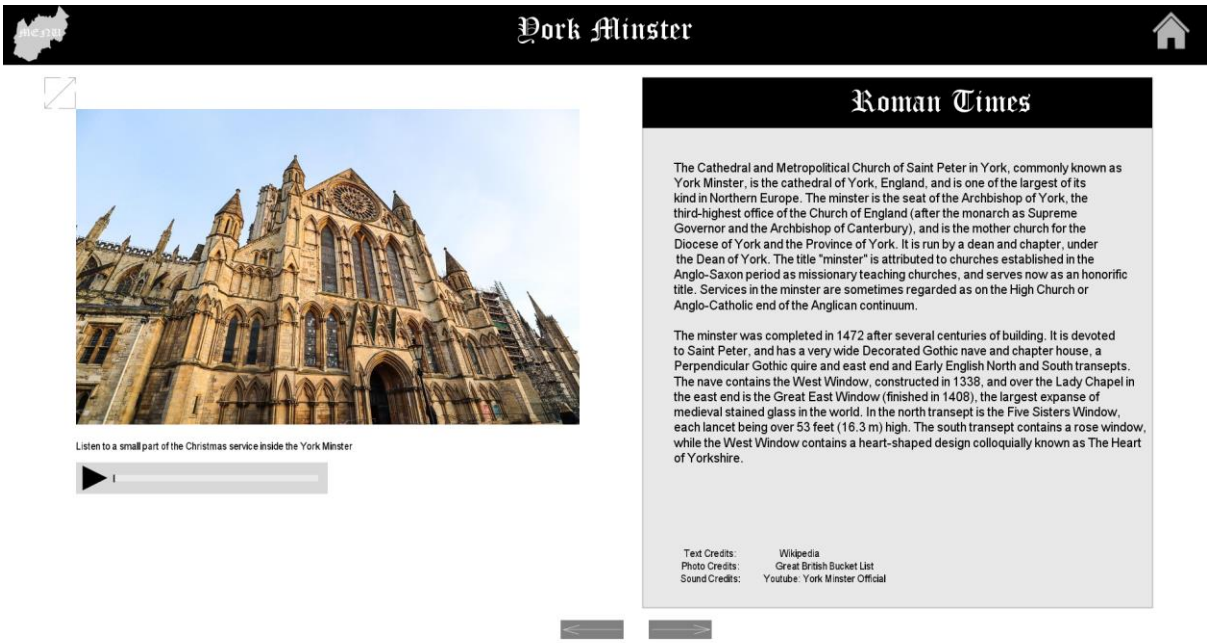
```
void slideshow(ArrayList<PImage> name, int slidenumber,String creditPicture1,String creditPicture2,String creditPicture3) {

  image(name.get(slidenumber), width/2-500, height-910, 1000, 700);
  stroke(colorbutton);
  if (slidenumber==0) {
    color1=#D8D8D8;
    color2=255;
    color3=255;
    credit=creditPicture1;
  }
  if (slidenumber==1) {
    color2=#D8D8D8;
    color1=255;
    color3=255;
    credit=creditPicture2;
  }
  if (slidenumber==2) {
    color3=#D8D8D8;
    color2=255;
    color1=255;
    credit=creditPicture3;
  }
  textFont(textfont);
  fill(0);
  textSize(15);
  text("Credits: ",  width/2-470,height-950);
  text(credit,width/2-350,height-950);
  fill(color1);
  circle(width/2-100, height-170, 10);
  fill(color2);
  circle(width/2, height-170, 10);
  fill(color3);
  circle(width/2+100, height-170, 10);
}
```

Slide 0

Slide 1

Slide 2

The page for the York Minster has the feature that was the hardest to implement, the controller for the sound.



The design of it is quite simple, the play and pause buttons are made by me in illustrator, while the rectangles are made inside processing. In regards to the code, inside the class OptionsPage I have implemented only the visual features of the controlor which are controlled by the press of the mouse.

```
void sound(String text, String credSound) {
  textFont(textfont);
  textSize(15);
  fill(0);
  text(text, 340, height-370);
  fill(#D8D8D8);
  rect(120, height-350, 400, 50);
  fill(#EAEAEA);
  rect(180, height-330, 325, 10);
  if (pressedPlay==false) {
    image(play, 130, height-345, 40, 40);
  } else if (pressedPlay==true) {
    image(pause, 130, height-345, 40, 40);
  }
  fill(0);
  text("Sound Credits: ",width-790,height-160);
  text(credSound,width-610,height-160);
}
```

Inside “Computational_Thinking” I have implemented the code that makes it functional. Because I wanted the music to sound better I decided to use Minim.

Computational_Thinking | First_page

```
1 import ddf.minim.*;
2 import ddf.minim.analysis.*;
3 import ddf.minim.effects.*;
4 import ddf.minim.signals.*;
5 import ddf.minim.spi.*;
6 import ddf.minim.ugens.*;
7 Minim minim;
8 AudioPlayer serviceMinster;
```

Here I have assigned the sound file to the variable.

```
minim=new Minim(this);
serviceMinster=minim.loadFile("York Minster Advent and Christmas 2018.mp3");
```

Here inside case 12, I have added, apart from the common functions, the function sound() which as seen above helps in drawing the controller.

```
case 12:
  yorkminsterpage=new OptionsPage();
  yorkminsterpage.draw();
  yorkminsterpage.header("York Minster");
  yorkminsterpage.body(picturesfortheoptions.get(2),"Roman Times",yorkminstertext);
  yorkminsterpage.credits("Wikipedia","Great British Bucket List");
  yorkminsterpage.sound("Listen to a small part of the Christmas service inside the York Minster", "Youtube: York Minster Official");
  stroke(0);
  float position = map( serviceMinster.position(), 0, serviceMinster.length(), 180, 180+330 );
  fill(0);
  rect(180, height-330, position-180, 10);
  if (pressedPlay==true ) {
    serviceMinster.play();
  } else {
    serviceMinster.pause();
  }
  yorkminsterpage.nextButton();
  yorkminsterpage.previousButton();
  break;
```

I have used map() function in order to create a more visual approach to the sound controller, and map() is a more elegant method of doing it. Inside map(), there is the variable that is always changing, “serviceMinster.position”, the minimum of the original range “0” and the maximum of that range “serviceMinster.length()” and then I have assigned the variables of the new range that represent the rectangle that needs to be filled in regards to how far the song went on. The reason why the new range is in regards to the rectangle is because the width of the that rectangle moves as far as the position goes. In order to make the controller more interactive, inside the function mousePressed() there is this line of code.

```
597   if (mouseX>180 && mouseX<180+330 && mouseY>height-350 && mouseY<height-300) {
598     int position=int(map(mouseX, 180, 330+180, 0, serviceMinster.length()));
599     serviceMinster.cue(position);
600   }
```

The role of this line of code is to allow the user to skip to different parts of the song as long as they are in the right area of the screen. This time the variable that is always changing is the value of mouseX, the original range is between the beginning of the rectangle and the beginning plus the width of it, and the new range is between zero and the length of the song. The serviceMinster.cue(position) is resposible for setting the position from which the song should continue in regards to where the mouseX is equal to when the mouse has been pressed.

Another feature I have implemented is the timer that restarts the programme, which means when the user has been on a page for approximately 1 minute and has not pressed the mouse, they will be redirected to the page with the menu. The way the timer works is showcased in the picture below.

```
if (input>8) {
  timedout--;
}
if (timedout==0) {
  input=8;
}
println(timedout);
```

These lines of code are inside the draw() function inside “Computational_Thinking” because this process has to function like a loop, as seen inside the mousePressed function, in the picture below, the timedout variable is restarted back to 2000 when the mouse has been pressed anywhere on the screen. The reason why I have required the input to be bigger than 8 is because the inputs for the pages that represent options from the menu are bigger than 8, therefore I was not required to name all of them.

```
if (input>8 && mouseX>0 && mouseX<width && mouseY>0 && mouseY<height) {
  timedout=2000;
}
//next buttons for the page with the options
```

In order to create a more aesthetic design, I have decided to create two main fonts, one for the text and credits and the other one for the titles. The font for the text and the credits is a more readeble one, “Arial”, while the font for the titles is “OldEnglishText”. Another feature in regards to the text is the one I have implemented for the word “York” at the beginning of the programme. I have decided I do not want the title to be fully opaque because I wanted both

the image and the title to be visible to the user.

```
void elementsFirst() {
  fill(255, 200);
  textSize(150);
  textFont(text);
  text("YORK", width/3+50, height/3);
```

The code above shows the approach I have implemented, usually when filling a text, I use a code such as “#EAEAEA” and that means the object added to the screen will be fully opaque and filled with that color. By adding another number which is called the alpha number, the programmer is allowed to set a different level of transparency. The level 255 means the object is going to be fully opaque, after testing the code a few times, I decided the best value for the alpha number is 200.

Reflective section

The button inside the page for King’s Manor is not working completely. The only way the user can acces it is by pressing on the square representing it in the menu and read what it says on the dialogue circle, however in order for the picture to be clickable they have acces the page by the buttons next and previous. The reason why this method is working is because the buttons next and previous are redirecting the user to the input of the page King’s Manor where the user is able to press the image in order to be redirected to the extended version of that page.

Another existing bug I have found is inside the page York Minster. Although the controller for the music is working, if the user leaves the page while the music is still on, they will still hear the music even though they are on the menu page or any other page inside the programme. I have tried implementing a method that would stop this bug, but it was not successful. The method I have tried was consists in checking if the music is playing and if the user was still on that page, if the music was playing and the user is no longer on the page, a boolean variable would change its value which will result in making the music stop by muting the sound. This method did not work because the user would not be able to listen to that song anymore and the sound of it is going to be muted until the user restarts the programme. In addition to this bug, another problem I have faced with the controller of the music is the one that icon for pause would not turn on when the music reached its end. I have not found a solutio in that sense.

Regarding the possible improvements, apart from solving the existing bugs that are present now, I would also want to add more animations. When the user is clicking on a new page, it would be nice to add a small animation, that would behave like a GIF and it would symbolise the content they will see on the page they are trying to open, or maybe it can be the same animation for every page. The way this animation would work is by integreting a timer that would start when the input representing the menu would be switched to another one that is representing one of the pages and while the timer is for example less then 20 seconds, an array of images will be loaded on the screen and they would not be fully opaque. Another improvement I would want to implement is to make the programme fit every window in fullsize by using displayWidth and displayHeight.

Word count: 2512

Reference List

//Inside the slideshow from the beginning

1. Peter K Burian, 2018. A portion of the ruins of St Mary's Abbey [image online]. Available at: < https://en.wikipedia.org/wiki/York > [Accessed 16 May 2021].
2. The Shambles Old Street [image online]. Available at: < https://www.visitbritain.com/ae/en/england/northern-england/york> [Accessed 24 April 2021].
3. York Minster. [image online]. Available at:< https://docbrown.info/docspics/yorkscenes/yspage02.htm >[Accessed 24 April 2021].
4. Roman Walls. . [image online]. Available at: < https://www.bbc.com/news/uk-england-york-north-yorkshire-43448590 >[Accessed 24 April 2021].

Inside the house page

//The images for the image buttons at the bottom of the page

5. Anthony Chappel-Ross, 2019. Jorvik Viking Festival. Available at: < https://yorkmix.com/nine-pix-spectacular-fireworks-and-big-crowds-at-the-viking-fest-weekend/ >[Accessed 27 April 2021].
6. Ruth Downie, 2017. Eboracum Roman Festival [image online]. Available at: <https://ruthdownie.com/2017/05/29/have-shoes-will-travel/ >[Accessed 27 April 2021].
7. Tom Blackwell, 2010. The Shambles, York: The Most Medieval Street in England [image online]. Available at: < https://www.amusingplanet.com/2012/09/the-shambles-york-most-medieval-street.html >. [Accessed 27 April 2021].

The images inside the pages from history

8. JORVIK Viking Centre. [image online]. Available at: < http://letsgowiththechildren.co.uk/places-to-go/jorvik-viking-centre/ >[Accessed 28 April 2021].
9. York Photo Images, 2016.The Romans invade York again! More than 50 storm the city centre. [image online]Available at:< https://yorkmix.com/the-romans-invade-york-again-more-than-50-storm-the-city-centre/ >[Accessed 28 April 2021].
10. Medieval York, the Cathedral, and Vikings Day Trip from Manchester. [image online]. Available at: < https://www.klook.com/activity/20924-medieval-minster-cathedral-vikings-york-trip-manchester/ >[Accessed 28 April 2021].

//Images for the menu

11. Tim Green, 2016. Front facade of King's Manor. [image online]. Available at: < https://en.wikipedia.org/wiki/King%27s_Manor > [Accessed 28 April 2021].
12. An illustration from 1807 during the reign of King George III showing the Multangular Tower and the city walls. [image online]. Available at: < https://en.wikipedia.org/wiki/York_city_walls#:~:text=History-,Roman%20walls,part%20of%20the%20fort's%20defences. >[Accessed 28 April 2021].
13. York Minster. [image online] Available at: < https://www.theguardian.com/commentisfree/2016/oct/14/bellringer-york-minster-wrong-sack-us-silence-church-bells >[Accessed 28 April 2021].
14. Anglian Tower. [image online]. Available at: < http://www.bollington.eu/York.html >[Accessed 28 April 2021].
15. Horace, 2012. Clifford’s Tower [image online] Available at: < https://www.jorvik.co.uk/cliffords-tower/ > [Accessed 28 April 2021].
16. Rept0n1x, 2010. Railway Station. [image online]. Available at: < https://en.wikipedia.org/wiki/York_railway_station >[Accessed 8 May 2021].

17. Mike Peel, 2008. Cold War Bunker. [image online]. Available at: < https://en.wikipedia.org/wiki/York_Cold_War_Bunker > [Accessed 8 May 2021].
18. Paul Johnson and Jonathan Sharman, 2006. The Red Tower. [image online] Available at: < http://www.yorkshireguides.com/red_tower.html >[Accessed 3 May 2021].
//Images inside the pages with the options
19. DAVID ROSS. King's Manor. [image online]. Available at: < https://www.britainexpress.com/cities/york/kings-manor.htm > [Accessed 3 May 2021].
20. York Minster cathedral. [image online]. Available at: < https://www.greatbritishbucketlist.com/york-minster-facts/ > [Accessed 3 May 2021].
21. Clifford's Tower. [image online]. Available at: < https://www.gpsmycity.com/attractions/cliffords-tower-31455.html >[Accessed 3 May 2021].
22. Chris Venne. York station [image online]. Available at: < https://www.gazettelive.co.uk/news/teesside-news/york-station-power-cut-closure-15500941 >[Accessed 8 May 2021].
23. Shutterstock. York Cold War Bunker. [image online]. Available at: < https://www.historyhit.com/locations/york-cold-war-bunker/ > [Accessed 10 May 2021].
24. Red Tower [image online]. Available at: < http://redtoweryork.org.uk/red-tower-history/#:~:text=Fishergate%20Postern%20Tower%20was%20started,waters%20of%20the%20King's%20Fishpond. > [Accessed 3 May 2021].
//extensions of the pages
25. Multiangular Tower. [image online] Available at: < http://www.yorkshireguides.com/multiangular_tower.html >[Accessed 28 April 2021].
26. King's Manor Refurbishment. [image online]. Available at: < https://www.york.ac.uk/campus-investment/current/kings-manor-refurbishment/ >[Accessed 8 May 2021].
27. Rodhullandemu, 1987. Kings Manor Lawn [image online]. Available at: < https://en.m.wikipedia.org/wiki/File:Kings_Manor_Lawn,_York.jpg > [Accessed 8 May 2021].
28. Chopton. The King's Manor. [image online]. Availabe at: < https://www.gpsmycity.com/attractions/the-kings-manor-27999.html >. [Accessed 8 May 2021].
// sound inside the page for York Minster
29. York Minster Official, 2018. York Minster Advent and Christmas 2018. [online] Available at: < https://youtu.be/xAopaZsikh4 >. [Accessed 8 May 2021].
//text
30. History of York. VisitYork. [online]. Available at: < https://www.visityork.org/general-and-tourist-info/history >[Accessed 26 April 2021].
31. Viking History. VisitYork. [online]. Available at: < https://www.visityork.org/general-and-tourist-info/history/viking >[Accessed 26 April 2021].
32. Roman History. VisitYork. [online]. Available at: < https://www.visityork.org/general-and-tourist-info/history/roman >[Accessed 26 April 2021].
33. Medieval History. VisitYork. [online]. Available at: < https://www.visityork.org/general-and-tourist-info/history/medieval >[Accessed 26 April 2021].

34. Wikipedia. York city walls [online]. Available at: < https://en.wikipedia.org/wiki/York_city_walls >. [Accessed 2 May 2021].
35. Wikipedia. York Minster [online]. Available at: < https://en.wikipedia.org/wiki/York_Minster > [Accessed 2 May 2021].
36. David Ross. Anglian Tower. [online]. Available at: < https://www.britainexpress.com/counties/yorkshire/properties/anglian-tower-york.htm#:~:text=But%20when%20was%20it%20built,are%20educated%20guesses%20at%20best.&text=The%20Anglian%20Tower%20originally%20stood,but%20only%20one%20storey%20remains. > [Accessed 10 May 2021].
37. Wikipedia. York Castle. [online]. Available at: < https://en.wikipedia.org/wiki/York_Castle > [Accessed 7 May 2021].
38. ICE. York railway station. [online]. Available at: < https://www.ice.org.uk/what-is-civil-engineering/what-do-civil-engineers-do/york-railway-station#:~:text=Designed%20by%20former%20ICE%20president,station%20buildings%20and%20a%20hotel. > [Accessed 4 May 2021].
39. Wikipedia. King's Manor. [online]. Available at: < https://en.wikipedia.org/wiki/King%27s_Manor >[Accessed 3 May 2021].
40. Wikipedia. York Cold War Bunker. Available at: < https://en.wikipedia.org/wiki/York_Cold_War_Bunker >[Accessed 8 May 2021].
41. Red Tower York. Red Tower. [online]. Available at: < http://redtoweryork.org.uk/red-tower-history/#:~:text=Before%20his%20defeat%20in%201485,the%20loyalty%20of%20the%20city. > [Accessed 10 May 2021].