M04 Simple Doc

Isaac Lohnes

Prog 2500

Below is what my xaml for menu looks like:

```
<MenuItem Header ="Edit Hair">
    <MenuItem Header="Next" Command="{Binding NextHairCommand}" InputGestureText="ctrl + L"></MenuItem>
    <MenuItem Header="Previous" Command="{Binding LastHairCommand}" InputGestureText="ctrl + K"></MenuItem>
</MenuItem>
<MenuItem Header ="Edit Nose">
    <MenuItem Header="Next" Command="{Binding NextNoseCommand}" InputGestureText="ctrl + P"></MenuItem>
    <MenuItem Header="Previous" Command="{Binding LastNoseCommand}" InputGestureText="ctrl + O"></MenuItem>
</MenuItem>
<MenuItem Header ="Edit Mouth">
    <MenuItem Header="Next" Command="{Binding NextMouthCommand}" InputGestureText="ctrl + I"></MenuItem>
    <MenuItem Header="Previous" Command="{Binding LastMouthCommand}" InputGestureText="ctrl + U"></MenuItem>
</MenuItem>
<MenuItem Header ="Edit Eyes">
    <MenuItem Header="Next" Command="{Binding NextEyesCommand}" InputGestureText="ctrl + J"></MenuItem>
    <MenuItem Header="Previous" Command="{Binding LastEyesCommand}" InputGestureText="ctrl + H"></MenuItem>
</MenuItem>
enuItem>
```

I’ve separated my logic for radio button clicks:

```
/*Logic for button clicks:*/

private void hairLogic(string Hair)
{
    switch (Hair)
    {
        case "hair_1":
            currentHair = 0;
            updatePNG(hair1, 0);
            break;
        case "hair_2":
            currentHair = 1;
            updatePNG(hair2, 0);
            break;
        case "hair_3":
            currentHair = 2;
            updatePNG(hair3, 0);
            break;

    }
}
```

Button checked method now calls <bodypart>Logic() function, snipped below:

```
private void hairRadioButton_Checked(object sender, RoutedEventArgs e)
{
    RadioButton rb = sender as RadioButton;
    if (rb != null)
    {
        string hairName = rb.Tag.ToString();
        hairLogic(hairName);
    }
}
```

Commands are set to call actions using your supplied CommandHandler class:

```
HairNextCommand = new CommandHandler(nextHairAction, true);
HairLastCommand = new CommandHandler(lastHairAction, true);
EyesNextCommand = new CommandHandler(nextEyesAction, true);
EyesLastCommand = new CommandHandler(lastEyesAction, true);
MouthNextCommand = new CommandHandler(nextMouthAction, true);
MouthLastCommand = new CommandHandler(lastMouthAction, true);
NoseNextCommand = new CommandHandler(nextNoseAction, true);
NoseLastCommand = new CommandHandler(lastNoseAction, true);

DataContext = new {
    NextHairCommand = HairNextCommand,
    LastHairCommand = HairLastCommand,
    NextEyesCommand = EyesNextCommand,
    LastEyesCommand = EyesLastCommand,
    NextMouthCommand = MouthNextCommand,
    LastMouthCommand = MouthLastCommand,
    NextNoseCommand = NoseNextCommand,
    LastNoseCommand = NoseLastCommand

};
```

Each action calls nextLoad or lastLoad:

```
/*Actions:*/

private void nextHairAction() {
    nextLoad("hair");
}
private void lastHairAction()
{
    lastLoad("hair");
}
private void nextEyesAction()
{
    nextLoad("eyes");
}
private void lastEyesAction()
{
    lastLoad("eyes");
}
private void nextNoseAction()
{
    nextLoad("nose");
}
private void lastNoseAction()
{
    lastLoad("nose");
}
private void nextMouthAction()
{
    nextLoad("mouth");
}
```

Sample from lastload:

```
private void lastLoad(string bodyPart)
{

    switch (bodyPart)
    {
        case "mouth":
            int newIndexM = currentMouth - 1;
            if (newIndexM < 0)
            {
                newIndexM = mouthArr.Count -1;
            }
            currentMouth = newIndexM;

            updatePNG((BitmapImage)mouthArr[newIndexM], 300);
            break;

        case "hair":
            int newIndexH = currentHair - 1;
            if (newIndexH < 0)
            {
                newIndexH = hairArr.Count -1;
            }
            currentHair = newIndexH;

            updatePNG((BitmapImage)hairArr[newIndexH], 0);
            break;

        case "eyes":
            int newIndexE = currentEyes - 1;
```

Sample output: