Final Project Plan Assignment

**Team Name:** Unity

**Members:** Shawn Kang, Royce Weigel, Mateen Saghafi, Neel Pochareddy

**Team Song:** Fast Car by Tracy Chapman

**Goal:**  A remix video, animation.

Audio playing on the background.

Import videos and pictures to create animation.

Tell the story of the song.

**Requirements:**

Five classes (Main class included)

Each class should have at least four methods (Main class excluded)

**Classes:**

Main Method - control the sequence

The main method will call all the classes and use the methods in the classes to create the final video to our satisfaction. The next paragraph will describe how each class will be used in the main method.

The main class will first open a Window class. Then, we run the Lyrics and display the lyrics on the window, and play Music class on the background. The main class will call the Image and video classes on corresponding lyrics. We will have a scheduler to keep track of the order of the displayed image and video.

By being able to coordinate the other four classes, this main method will allow a well transitioned, continuous flow. The audio along with the video and images will all be in sync. Everything will be kept in order and will properly lead on to the next part of the song.

Music - import and play the audio //Neel

* importMusic // put the song into the video file(accept song name, void)

This method imports the whole song into the video so we can use the song to

* setMusicLength// crop certain parts of the song if needed such as the ending or filler parts

If the audio file that we put in has breaks or non music parts in it then we have to crop the audio file so that the video continues smoothly without any random pauses in it.

* setVolume // set the volume of the music (accept integer, void)

This method is very self explanatory. If the imported video’s audio is too loud or not to our satisfaction, this method will edit it to our liking

* pauseMusic //pause music when pause video(void)

If for whatever reason the video is paused, the music should pause with it simultaneously

Lyrics - display the verses and chorus// Royce

* importLyrics // import text file that contains the lyrics of the song(accept string, void) This method imports the text file of our chosen song. Importing the lyrics is important because if we do not import the text file of the lyrics then there will be nothing to display when calling the method in lyricDisplay.
* fontSize// set appropriate font length(void) This method determines and controls the size of the font. This is useful because it gives us the ability to make our font to a size that best serves the classroom setting, so all words are easily readable.
* lyricDisplay // display lyrics on screen(void). This method displays the lyrics on the screen. This is important because the project is based around our chosen song “Fast Car” and displaying the lyrics is essential to the success of this project.

Scheduler - keep track of the flow // Shawn

* currentSongTime // keep track of the time of the current song (return integer)
* playSong // start to play the song (void)// The playSong class will start to count how much time has passed.
* checkSwitch // check if the designated time reaches, if yes, the new image or video will be displayed (return boolean)
* pauseSong // pause song if needed (void) Stop counting the time.

Window - create a window frames //Shawn

* + setWindowSize// Set Window size (accept integer, void)
  + setWindowLocation// Set Window location on the screen (void)
  + setWindowFrames // Set up window icon and feature(void)
  + setWindowBackground // set up a wallpaper on the background (accept image, void)

Window Buttons - Create window buttons //Shawn

* + setButtonLocation // set button location (accept integer1, integer2, void)
  + setButtonSize // set button size (accept integer, void)
  + setButtonColor // set button color (accept string, void)
  + setButtonFont // set button font (accept string, integer, void)
  + setButtonName // set the tag name of the button (accept string, void)

Image - import and edit images //Mateen

* ImageBackground //The image being imported, will be used to control what appears as we progress through the song. Images will appear that correlate to the song and will stay on for a brief moment.
* ImageSelf // the image file (return image) When called upon, the desired image will be returned, meaning that it will show up during the part of the song that we want it. This will be interacting with other classes in order for it to appear at the correct part of the song.
* ImageColor //color of the image appearing(accept string, void) This determines the color of the image that appears. The color must be accepted in order for it to be able to have the correct image color. This will interact with ImageSelf in order for the correct picture to have the desired color.
* ImageLength //Duration on screen(accept integer, void). Is used to set the amount of time for the image to be displayed. The images are to be in sync with the lyrics, so we must be able to set a length where it correctly works with those lyrics.
* ImageSize //How large the image will appear(accept integer, void)//Is used to set how large the image should appear on the screen. The desired size will be implemented through this and is used to accept the size we desire and allow ImageSelf to properly return the image.