Initial Design

Morales, Goldenfeld, Karthikeyan

KGS rhythms

UML

Class - Project_Dancing (main)

Instance Variables:

All of these methods will be used to control which “screen” is to be displayed

- Boolean startscreen,
- Boolean songselect,
- Boolean dancing,
- Boolean resultscreen,

Methods:

- setup(), sets up the dimension of the screen, loads the song shelf and other necessary data that will be used by the program.
- draw(), this method is what will actually display the desired “screen”, this method will be constantly refreshed during runtime and will utilize the Boolean instance variables to determine which screen the player should be on
- mousePressed(), this method will be constantly listening for mouse input and if the mouse if pressed while over a song then it will initiate the “game” phase based off of the song that was on the shelf

Class - Analyzer

Instance Variables:

- Not sure?

Methods:

-

Class - Song

Instance variables:

- String songname, song's name
- Double songlength, songs length
- Int [] beats, an array that will contain the information for when a shape will be "thrown" onto the screen, the sequence will be based off how the Analyzer analyses the song (the Analyzer is a separate independent class)
- Mov video/lyricvideo, will hold the mp4 file that goes along with the song

  Methods:

  - getSong(), returns song stored in instance variable
  - getDuration(), returns songs duration
  - getAudio(), returns the compressed rhythm sequence
  - getVideo(), returns the mp4 video file

Class - Song_Shelf

Instance Variables:

- Song [] songs, this array will contain all of the songs that can be played in the game, the Project_Dancing (main game) class will have a method mousePressed() or utilize a controller class that will be used to determine which song the player has selected based on where the mouse is hovering

Class - Animations

Instance Variables:

- ?, not sure if any are necessary, still determining

Methods:

- startPulsation(), on the start screen one of the animations will be allow the “press to start” text to fade in and fade out, a nice little effect
- shelfSlide(), when a user hovers there mouse over a song the shelf should pop out, a segment of the song should be played so the user can hear what they will be playing

- shapeOverlay(), this method will cast an overlay on top of the video and will display regions where the player will need to tap the buttons in unison with the shape being displayed
- throwShape(), this method will “throw” shapes on to the screen by grabbing values that have been sequenced inside of the Int [] beats array located in the song object which will be stored inside of the Song_Shelf , it will then throw it 2-3 seconds earlier and enter the overlay region where the user should then tap the corresponding button
- displayScore(), this method will display the players score that has been accumulated throughout the session
- incrementScore(), when a shape has been “thrown” on the screen and if the button is pressed while the shape is within the region the shape will be “popped” and points will be added to the score, if the button is pressed to early or too late then the players score will not increase at all
- displayCombo(), if the player gets more than 5 beats correct in a row then a combo counter for consecutive correct beats will be displayed, if they get a beat wrong the combo counter will disappear

Class - Analyzer_Main

Instance Variables:

- ?

Methods:

- main(), will construct a Analyzer object with the song that will be

Class - Analyzer

Instance Variables:

- Song sample, will hold the song that is going to be “analyzed”

Methods:

-

adw