Assignment 6

Milestones, Timeline, and Effort Matrix

Rob Kelly, Stephanie Mullins, Athulya Ganesh

**Milestones:**

*Computer Vision*

* + Fully workable gesture recognition using live video.
    - Be able to use a live video feed for gesture detection and getting near perfect recognition of what the gesture is.
  + Fully workable body detection using live video.
    - Be able to use a live video feed for body detection and getting a fully working skeletal detection of a human.
  + Fully workable body recognition using live video.
    - Be able to use a live video feed for body recognition matching to a given pose. Static poses only.
  + Fully workable body recognition while moving and using live video.
    - Be able to use a live video feed for moving body poses that can be followed along with the algorithm to see if the human movement matches the given movement.
  + Near real-time recognition when using live video.
    - Reduce any lag from processing and computation as much as possible to make it as close to real-time movements.
  + Compute movement score.
    - Determine how much the human movements match the given movements and calculate the score.

*Web Interface*

* + Technology Selection
    - Research and select appropriate frontend and backend technologies.
    - Decide on suitable APIs for integrating necessary features.
  + UI Design
    - Create mock UI interfaces detailing the layout and features for each page.
    - Develop page-by-page wireframes and prototypes for the web app.
  + Environment Setup
    - Set up the development environment, including configuring chosen frameworks.
    - Prepare the server and database environment, ensuring proper access and security settings.
  + Web App Skeleton
    - Develop the initial structure of the web app with basic navigation.
    - Create a landing page where users can choose songs and performance options.
  + Performance Interfaces
    - Design separate interfaces for audio-only and audio-video performances.
    - Implement performance recording and playback functionality.
  + Lyrics Module
    - Develop the module to display lyrics on the screen in sync with the chosen song.
    - Ensure a user-friendly interface for scrolling through lyrics.
  + Gesture Guidance
    - Create a module that lists gestures for users to follow along during performances.
    - Implement features to highlight or track user gestures in real-time.
  + Scoring and Restart
    - Develop a scoring system to evaluate user performances.
    - Enable users to view their scores and provide an option to restart or try another performance.
  + Leaderboard and User Management
    - Design and implement a leaderboard feature to display user scores.
    - Develop user registration and login pages to save user information securely.
  + Database Setup
    - Set up the backend database, either locally or using a cloud service.
    - Establish appropriate data structures and relationships.
  + Testing and Refinement
    - Perform comprehensive testing at each development stage, including unit testing, integration testing, and user testing.
    - Collect feedback and refine the web app based on user and team input.
  + Deployment and Maintenance
    - Deploy the web application within the specified timeline using a hosting service.
    - Ensure ongoing maintenance to keep the application running smoothly.

*Audio Processing*

* Research Audio Processing Libraries
  + Investigate and determine which audio processing library or libraries to use for audio processing.
* Implement Selected Audio Processing Library
  + Using the selected library or libraries, set up a functional proof-of-concept for processing audio files so that the proper information needed can be utilized.
  + After proof-of-concept is built out, implement the audio processing method into the main project.
* Research Audio to Text Transcription Methods
  + Investigate and select which audio-to-text transcription method is best for the project.
  + Provide proof-of-concepts for each potential method so that the team can decide which is best for the project.
* Research Pitch Detection Methods
  + Investigate and determine which method of pitch detection is best for the project.
  + Provide proof-of-concepts for each method and discuss with the team which method is best.
* Implement Lyrical and Pitch Accuracy Features
  + After the selected method for lyrical and pitch detection is completed, implement the strategy into the project, making sure to seamlessly incorporate it with the rest of the project features.
* Compute Composite Accuracy Algorithm
  + Research and investigate various methods of computing composite scores for situations similar to the project’s use case.
  + Select the method that fits the team’s needs best.
  + Once a method has been selected, implement the final algorithm into the codebase for the project to calculate a user’s lyrical and vocal accuracy.

**Timelines:**

Computer Vision Audio Processing Web Interface

| **October** | **November** | **December** | **January** | **February** | **March** |
| --- | --- | --- | --- | --- | --- |
| Research software and start setting up an environment. Begin gesture recognition | Continue and finish gesture recognition. Start full body still images skeletal detection. | Continue and finish still image full body detection. Work on live video with skeletal detection. | Work on recognizing human body movements with inputted movements. | Continue and finish fully body skeletal movements. Finish scoring of user movements. | Combine computer vision aspects to the rest of the app. |
| Research Audio Processing Libraries | Implement Selected Audio Processing Library | Research Audio to Text Transcription and Pitch Detection Methods | Implement Lyrical and Pitch Accuracy Features | Compute Composite Accuracy Algorithm | Combine Audio Processing Portion of Project with the Rest of the App |
| Technology Selection and UI Design and Environment Setup | Web App Skeleton and Database Setup | Performance Interfaces and Lyrics Module and Gesture Guidance | Scoring and Restart and Leaderboard and User management | Testing and refinement | Deployment and Maintenance |

**Effort Matrix:**

| **Task Description** | **Primary Responsibility** | **Stephanie (%)** | **Rob (%)** | **Athulya (%)** |
| --- | --- | --- | --- | --- |
| **Computer Vision Tasks (Stephanie)** | | | | |
| Research technologies | Stephanie | 30% | 30% | 30% |
| Implement gesture recognition | Stephanie | 35% | 30% | 35% |
| Expand gesture recognition to tracking | Stephanie | 35% | 30% | 35% |
| Research skeleton recognition and tracking | Stephanie | 30% | 30% | 40% |
| Implement skeleton detection | Stephanie | 35% | 30% | 35% |
| Expand skeleton detection to tracking | Stephanie | 35% | 30% | 35% |
| Implement skeletal tracking correctness | Stephanie | 30% | 30% | 40% |
| **Audio Processing Tasks (Rob)** | | | | |
| Research audio processing libraries | Rob | 30% | 40% | 30% |
| Implement selected audio processing library | Rob | 40% | 30% | 30% |
| Research audio to text transcription methods | Rob | 30% | 40% | 30% |
| Research pitch detection methods | Rob | 30% | 40% | 30% |
| Implement lyrical and pitch accuracy features | Rob | 40% | 30% | 30% |
| Compute composite accuracy algorithm | Rob | 40% | 30% | 30% |
| **Web Interface Tasks (Athulya)** | | | | |
| Research technologies | Athulya | 30% | 30% | 30% |
| Design mock UI interfaces | Athulya | 35% | 30% | 35% |
| Environment setup | Athulya | 30% | 35% | 35% |
| Web app skeleton | Athulya | 35% | 30% | 35% |
| Performance interfaces | Athulya | 35% | 30% | 35% |
| Lyrics module | Athulya | 35% | 30% | 35% |
| Gesture guidance | Athulya | 35% | 30% | 35% |
| Scoring and restart | Athulya | 35% | 30% | 35% |
| Leaderboard and user management | Athulya | 30% | 35% | 35% |
| Database setup | Athulya | 35% | 30% | 35% |
| Testing and refinement | Athulya | 30% | 30% | 40% |
| Deploy and maintain | Athulya | 30% | 30% | 40% |
| **Combined Tasks** | | | | |
| Hook audio processing into web app | Team Effort | 30% | 40% | 30% |
| Combine computer vision with audio and web app | Team Effort | 30% | 40% | 30% |
| Implement composite performance algorithm | Team Effort | 30% | 40% | 30% |