

# **Software Requirements and Design Document**

**For**

**Group #14**

Version 2.0

## **Authors:**

Joseph Schmitt, jcs20fh, JoeSchmitt-2  
Tyler Starling, jts20cz, Tetherly  
Talal Mahmoudi, tkm16c, Talal-mahmoudi  
Sergio Pantoja, sp19p, sergiopan19  
Nathan Zhao, nyz18, nyz18

## 1. Overview (5 points)

Keyboard Racer – Game with multiple modes where the user needs to match the timing of falling indicators to keyboard input to score with Piano themed layout. There will be a base mode like guitar hero or piano tiles, faster game mode (medium, hard), an accuracy mode, and a blitz mode where speed is based off user input and not falling indicators.

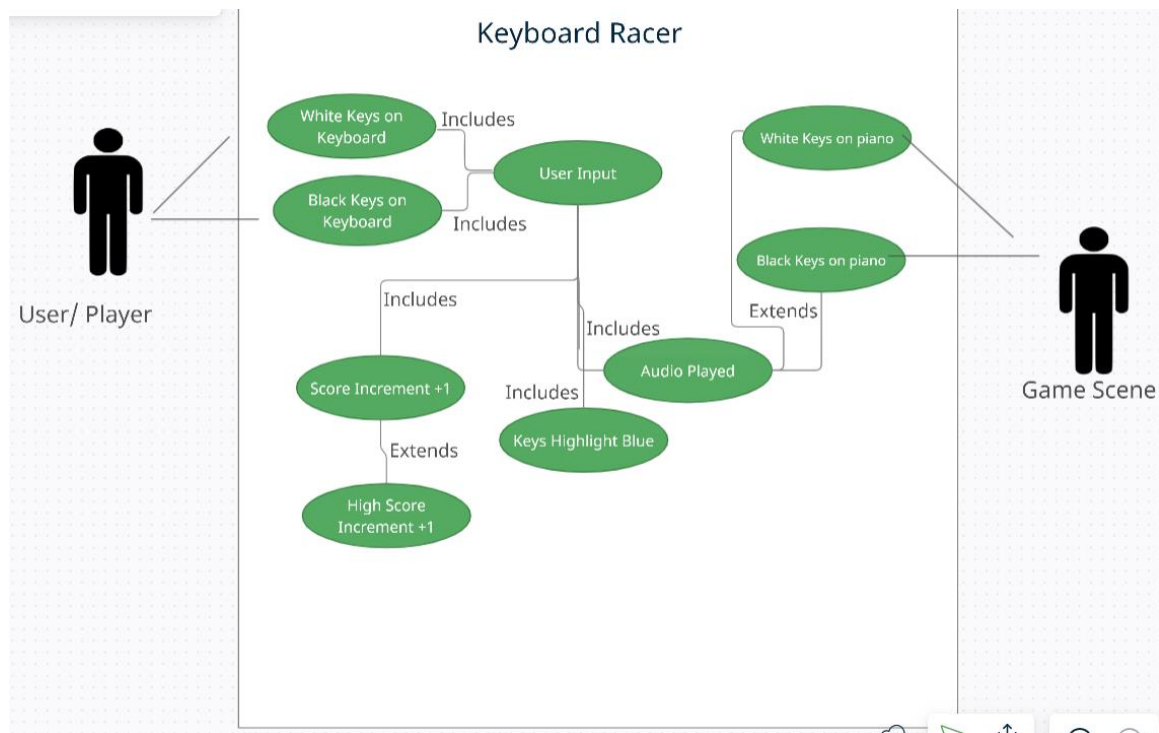
## 2. Functional Requirements (10 points)

When the user presses down the corresponding key must play the sound associated with that key, changing the color to blue and incrementing the score by one. Must spawn correct indicator to the associated key that must be pressed. The spawn indicator must spawn accurately and at the correct times.

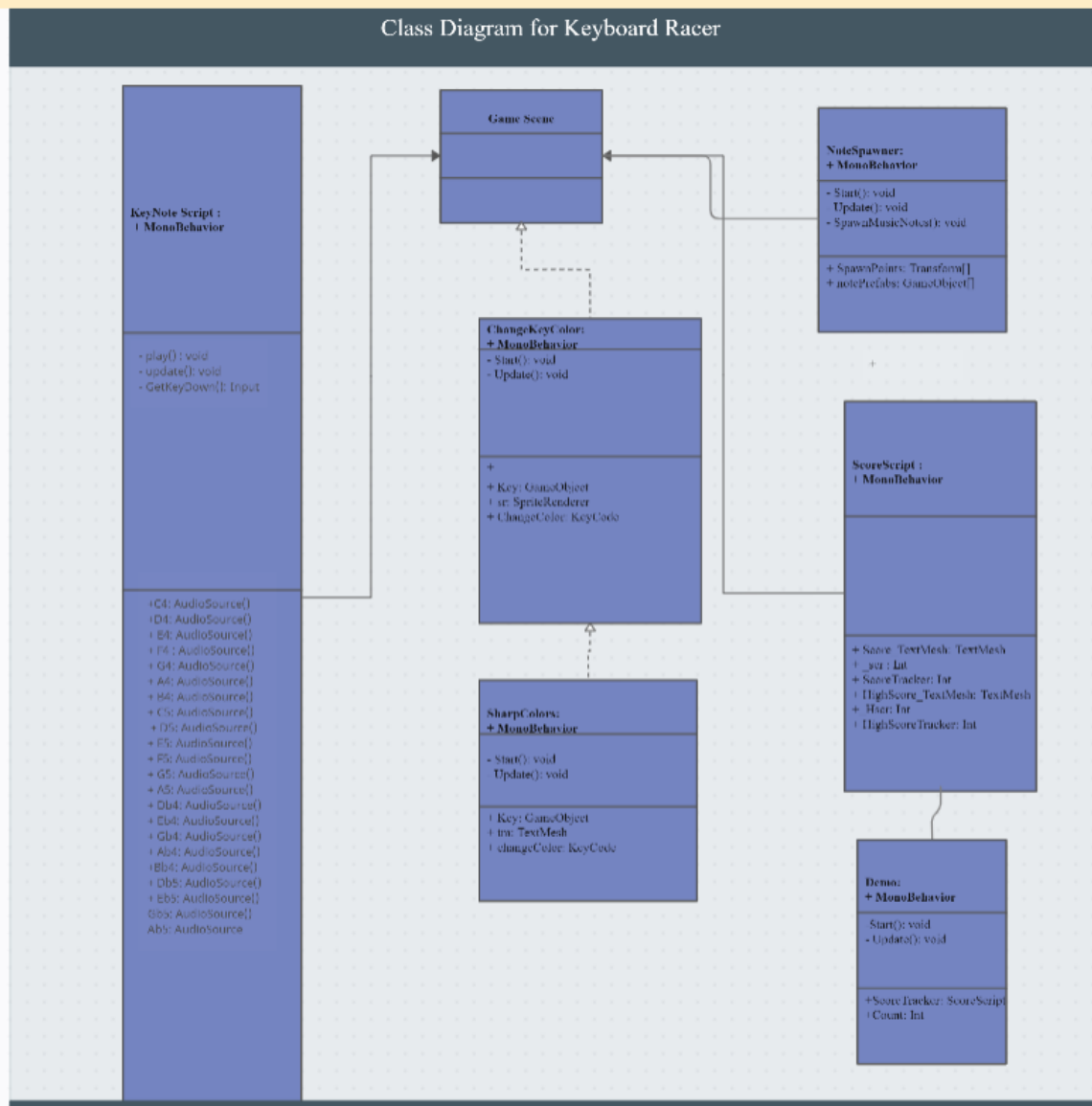
## 3. Non-functional Requirements (10 points)

The game must be secure. The game mustn't lag for a variety of user's systems. Autogenerated sound doesn't have to be generated if a user misses an indicator.

## 4. Use Case Diagram (10 points)



### 5. Class Diagram and/or Sequence Diagrams (15 points)



## **6. Operating Environment (5 points)**

The game, Keyboard Racer, is intended to be played on PC with development done in Unity and C#.

## **7. Assumptions and Dependencies (5 points)**

*An assumed factor when playing the game is that the user is attempting to play the game on a 'QWERTY' keyboard.*