Rhythm Master

Generated by Doxygen 1.8.17

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Class Documentation

3.1 ButtonController Class Reference

Inheritance diagram for ButtonController:

3.2 ChangeScene Class Reference

Inheritance diagram for ChangeScene:

Collaboration diagram for ChangeScene:

Public Member Functions

- void BtnChangeScene (string scene_name)
  Changes the scene being displayed.

3.2.1 Member Function Documentation

3.2.1.1 BtnChangeScene()

```
void ChangeScene.BtnChangeScene (
             string scene_name )  [inline]
```

Changes the scene being displayed.

Parameters

scene_name	The name of the scene to display.

The documentation for this class was generated from the following file:

- /mnt/c/users/david/documents/everything/school/3xa3/3xa3__l01_gr16_project/src/Assets/Scripts/Change↩ Scene.cs

3.3 CollisionDetector Class Reference

Inheritance diagram for CollisionDetector:

Collaboration diagram for CollisionDetector:

Public Member Functions

- void NoteHit ()

  Called when a note is hit. Calculates the accuracy of the hit. @detail Finds the distance from the note's centre and the key's centre at the time of hit.
- void NoteMissed ()

  Called when a note is not hit.
- void LongNoteHit ()

  Called when a long note is completed.
- void LongNoteClicked ()

  Called when a long note is hit.

The documentation for this class was generated from the following file:

- /mnt/c/users/david/documents/everything/school/3xa3/3xa3__l01_gr16_project/src/Assets/Scripts/Collision↩ Detector.cs

3.4 CompleteScreen Class Reference

Inheritance diagram for CompleteScreen:

Collaboration diagram for CompleteScreen:

Public Member Functions

- void DisplayCompleteScreen (float normalCount, float goodCount, float perfectCount, float missedCount, float accuracy, float finalScore)

  Display the completion screen to the main display.
- void SaveFinalScore ()

  Saves the player's submitted score to the high score database.

Public Attributes

- GameObject completeScreen

  The screen to be displayed.
- TextMeshProUGUI **normalHitText**
- GameObject submitButton

  The button to submit player score.

Static Public Attributes

- static CompleteScreen instance

  The one instance of CompleteScreen, to be referenced externally.

3.4.1 Member Function Documentation

3.4.1.1 DisplayCompleteScreen()

```
void CompleteScreen.DisplayCompleteScreen (
            float normalCount,
            float goodCount,
            float perfectCount,
            float missedCount,
            float accuracy,
            float finalScore )  [inline]
```

Display the completion screen to the main display.

Parameters

normalCount	The number of normal hits.
goodCount	The number of good hits.
perfectCount	The number of perfect hits.
missedCount	The number of missed hits.
accuracy	The player's hit accuracy, in percent.
finalScore	The final score.

The documentation for this class was generated from the following file:

- /mnt/c/users/david/documents/everything/school/3xa3/3xa3__l01_gr16_project/src/Assets/Scripts/Complete↩ Screen.cs

3.5 DetectKey Class Reference

Inheritance diagram for DetectKey:

Collaboration diagram for DetectKey:

Public Attributes

- string keyPressed

  The key the player selected.
- GameObject key

  The object related to the key.
- string keyColour

  The colour of the key the player selected.
- Text settingsButtonText

  The text reflecting the key the player chose.
- GameObject ui

  The prompt for a key displayed to the user.

The documentation for this class was generated from the following file:

- /mnt/c/users/david/documents/everything/school/3xa3/3xa3__l01_gr16_project/src/Assets/Scripts/Detect←↩ Key.cs

3.6 EffectManager Class Reference

Inheritance diagram for EffectManager:

Collaboration diagram for EffectManager:

The documentation for this class was generated from the following file:

- /mnt/c/users/david/documents/everything/school/3xa3/3xa3__l01_gr16_project/src/Assets/Scripts/Effect←↩ Manager.cs

3.7 Effects Class Reference

Inheritance diagram for Effects:

Collaboration diagram for Effects:

Public Attributes

- float lifetime = 1f

  Time before effect is despawned, in seconds.

The documentation for this class was generated from the following file:

- /mnt/c/users/david/documents/everything/school/3xa3/3xa3__l01_gr16_project/src/Assets/Scripts/Effects.cs

3.8 FileIO Class Reference

Inheritance diagram for FileIO:

Collaboration diagram for FileIO:

Public Member Functions

- string[ ] ReadFile (string pathName, char separator)

  Reads and returns the contents of the specified text file, splitting by the specified delimiter. @detail Note maps are specified by a set of individual entries each representing a set of notes to spawn simultaneously. Each entry's notes are separated by a slash (/) and are numbered from 0 to 4 from left to right. Long notes can additionally be identified by a decimal point, with the number after the decimal giving the long note's length.

Static Public Member Functions

- static List< LeaderboardEntry > GetLeaderboardList ()

  Reads the high scores from storage.
- static void AddEntryToLeaderboard (LeaderboardEntry entry)

  Write a new entry to the high score storage.

Static Public Attributes

- static FileIO instance

  The one instance of FileIO to be referenced globally.

3.8.1 Member Function Documentation

3.8.1.1 AddEntryToLeaderboard()

```
static void FileIO.AddEntryToLeaderboard (
             LeaderboardEntry entry )  [inline], [static]
```

Write a new entry to the high score storage.

Parameters

The	entry to be written.

3.8.1.2 GetLeaderboardList()

```
static List<LeaderboardEntry> FileIO.GetLeaderboardList ( )  [inline], [static]
```

Reads the high scores from storage.

Returns

A list of the entries in the high score list.

3.8.1.3 ReadFile()

```
string [] FileIO.ReadFile (
            string pathName,
            char separator )  [inline]
```

Reads and returns the contents of the specified text file, splitting by the specified delimiter. @detail Note maps are specified by a set of individual entries each representing a set of notes to spawn simultaneously. Each entry's notes are separated by a slash (/) and are numbered from 0 to 4 from left to right. Long notes can additionally be identified by a decimal point, with the number after the decimal giving the long note's length.

Parameters

pathName	The text file's full path, in UNIX format.
separator	The character to split the file by.

Returns

An array of strings containing the file's contents after being split.

The documentation for this class was generated from the following file:

- /mnt/c/users/david/documents/everything/school/3xa3/3xa3__l01_gr16_project/src/Assets/Scripts/FileIO.cs

3.9 GameManager Class Reference

Inheritance diagram for GameManager:

Collaboration diagram for GameManager:

Public Member Functions

- void StartMusic ()

  Start playing music.
- void NormalHit ()

  Called during a normal hit.
- void GoodHit ()

  Called during a good hit.
- void PerfectHit ()

Called during a perfect hit.

- void LongHit ()

  Called after a long note is completed.
- void NoteMissed ()

  Called when a note is missed.
- void LongClicked ()

  Called when a long note has been hit.
- void SetMultiplier (int mult)

  Setter method for multiplier.
- void SetScore (int score)

  Setter method for score.

Public Attributes

- AudioSource song

  The music to play during gameplay.
- AudioSource guitar

  The accompanying music, to play only when the player has hit the most recent note.
- int currentScore

  The current score during the session.
- int currentMultiplier

  The current score multiplier during the session.
- bool songStarted = false

  False if the song has not begun playing.
- float totalNotes = 0

  Total notes spawned.
- float normalHits = 0

  Normal hits by the player.
- float goodHits = 0

  Good hits by the player.
- float perfectHits = 0

  Perfect hits by the player.
- float missedHits = 0

  Missed hits by the player.
- float accuracy = 0

  Net hit accuracy, in percent.
- int scorePerNote = 100

  Baseline score to be awarded per note hit, before multipliers and accuracy bonuses.
- string[ ] noteList

  The complete sequence of notes to spawn for the selected song. @detail Notes from left to right are number from 0 through 4. Long notes are additionally denoted by a decimal following this number.
  Simultaneous notes are separated by a slash (/). For example, 0.2 gives a long leftmost note of length 2.
- TextMeshProUGUI scoreText

  Display text for score.
- TextMeshProUGUI multiText

  Display text for multiplier.

Static Public Attributes

- static GameManager instance

  The instance of GameManager to be referenced globally.

3.9.1 Member Function Documentation

3.9.1.1 SetMultiplier()

```
void GameManager.SetMultiplier (
             int mult )  [inline]
```

Setter method for multiplier.

Parameters

mult	The new multiplier.

3.9.1.2 SetScore()

```
void GameManager.SetScore (
             int score )  [inline]
```

Setter method for score.

Parameters

score	The new score.

The documentation for this class was generated from the following file:

- /mnt/c/users/david/documents/everything/school/3xa3/3xa3__l01_gr16_project/src/Assets/Scripts/Game↩ Manager.cs

3.10 LeaderboardEntry Class Reference

Public Attributes

- string name

  The chosen name for the entrant.
- int score

  The entrant's score.
- string date

  The date the entry was made.

The documentation for this class was generated from the following file:

- /mnt/c/users/david/documents/everything/school/3xa3/3xa3__l01_gr16_project/src/Assets/Scripts/Leaderboard↩ Entry.cs

3.11 LoadSettings Class Reference

Inheritance diagram for LoadSettings:

Collaboration diagram for LoadSettings:

Public Attributes

- Text **greenButtonSettings**
- Text **redButtonSettings**
- Text **yellowButtonSettings**
- Text **blueButtonSettings**
- Text **pinkButtonSettings**

The documentation for this class was generated from the following file:

- /mnt/c/users/david/documents/everything/school/3xa3/3xa3__l01_gr16_project/src/Assets/Scripts/Load↩ Settings.cs

3.12 MainMenu Class Reference

Inheritance diagram for MainMenu:

Collaboration diagram for MainMenu:

Public Member Functions

- void PlayGame ()

  Called when the user starts a new game.

The documentation for this class was generated from the following file:

- /mnt/c/users/david/documents/everything/school/3xa3/3xa3__l01_gr16_project/src/Assets/Scripts/Main↩ Menu.cs

3.13 MMSettings Class Reference

Inheritance diagram for MMSettings:

Collaboration diagram for MMSettings:

The documentation for this class was generated from the following file:

- /mnt/c/users/david/documents/everything/school/3xa3/3xa3__l01_gr16_project/src/Assets/Scripts/MM↩ Settings.cs

3.14 NoteObject Class Reference

Inheritance diagram for NoteObject:

Collaboration diagram for NoteObject:

Public Attributes

- KeyCode keyToPress

    The key that the user must press to hit this note.
- GameObject key

    The in-game key associated with keyToPress.

The documentation for this class was generated from the following file:

- /mnt/c/users/david/documents/everything/school/3xa3/3xa3__l01_gr16_project/src/Assets/Scripts/Note↩ Object.cs

3.15 NoteScroller Class Reference

Inheritance diagram for NoteScroller:

Collaboration diagram for NoteScroller:

Public Attributes

- float beatTempo

    The speed at which the note moves down the screen.
- bool hasStarted

    True if notes have started spawning.

The documentation for this class was generated from the following file:

- /mnt/c/users/david/documents/everything/school/3xa3/3xa3__l01_gr16_project/src/Assets/Scripts/Note↩ Scroller.cs

3.16 NoteSpawner Class Reference

Inheritance diagram for NoteSpawner:

Collaboration diagram for NoteSpawner:

Public Attributes

- GameObject[ ] notes = new GameObject[5]

    The five possible short notes that can be spawned at once.
- GameObject[ ] longNotes = new GameObject[5]

    The five possible long notes that can be spawned at once.
- GameObject[ ] keys = new GameObject[5]

    The five possible keys to hit.
- float BPM

    The tempo of the song, in beats per minute.
- string[ ] noteList

    The list of notes to spawn. @detail Notes from left to right are number from 0 through 4. Long notes are additionally denoted by a decimal following this number.

    Simultaneous notes are separated by a slash (/). For example, 0.2 gives a long leftmost note of length 2. An entry in noteList is spawned once every beat.

The documentation for this class was generated from the following file:

- /mnt/c/users/david/documents/everything/school/3xa3/3xa3__l01_gr16_project/src/Assets/Scripts/Note↩ Spawner.cs

3.17 PauseMenu Class Reference

Inheritance diagram for PauseMenu:

Collaboration diagram for PauseMenu:

Public Member Functions

- void Resume ()

    Resume gameplay after it has been paused.
- void Pause ()

    Pause gameplay in progress. Stops note spawning, movement, and music playing.
- void Settings ()

    Displays settings menu UI.

Public Attributes

- GameObject PauseMenuUI

    The pause menu display.
- GameObject settingsMenuUI

    The settings menu display.

Static Public Attributes

- static bool Paused = false

  True if the game is paused.
- static bool SettingsShown = false

  True if settings are being shwon.

The documentation for this class was generated from the following file:

- /mnt/c/users/david/documents/everything/school/3xa3/3xa3__l01_gr16_project/src/Assets/Scripts/Pause←↩ Menu.cs

3.18 PopulateLeaderboard Class Reference

Inheritance diagram for PopulateLeaderboard:

Collaboration diagram for PopulateLeaderboard:

Public Attributes

- Text textTemplate

  The text on the leaderboard screen.

The documentation for this class was generated from the following file:

- /mnt/c/users/david/documents/everything/school/3xa3/3xa3__l01_gr16_project/src/Assets/Scripts/Populate←↩ Leaderboard.cs

3.19 QuitGame Class Reference

Inheritance diagram for QuitGame:

Collaboration diagram for QuitGame:

Public Member Functions

- void **ExitGame** ()

The documentation for this class was generated from the following file:

- /mnt/c/users/david/documents/everything/school/3xa3/3xa3__l01_gr16_project/src/Assets/Scripts/Quit←↩ Game.cs

3.20 ScoreCalculator Class Reference

Inheritance diagram for ScoreCalculator:

Collaboration diagram for ScoreCalculator:

Public Member Functions

- float NormalHit (int currentScore, int currentMultiplier)

  Called when a normal hit has been made.
- float GoodHit (int currentScore, int currentMultiplier)

  Called when a good hit has been made.
- float PerfectHit (int currentScore, int currentMultiplier)

  Called when a perfect hit has been made.
- float LongHit (int currentScore, int currentMultiplier)

  Called when a long hit has been made.
- float NoteMissed ()

  Called when a note has been missed.
- float AccuracyCalculation (float normalHits, float goodHits, float perfectHits, float totalNotes)

  Calculates the user's net accuracy.

Public Attributes

- int multiplierTracker

  The number of notes going toward the next multiplier.
- int[ ] multiplierThresholds

  The number of notes that must be hit for the next multiplier.
- int scorePerNote = 100

  The score for a normal hit, before multipliers.
- int scorePerGoodNote = 125

  The score for a good hit, before multipliers.
- int scorePerPerfectNote = 150

  The score for a perfect hit, before multipliers.
- int scorePerLongNote = 200

  The score for a long hit, before multipliers.

Static Public Attributes

- static ScoreCalculator instance

  The instance of ScoreCalculator to be referenced globally.

3.20.1 Member Function Documentation

3.20.1.1 AccuracyCalculation()

```
float ScoreCalculator.AccuracyCalculation (
          float normalHits,
          float goodHits,
          float perfectHits,
          float totalNotes )  [inline]
```

Calculates the user's net accuracy.

Parameters

normalHits	The number of normal hits made.
goodHits	The number of good hits made.
perfectHits	The number of perfect hits made.
totalNotes	The number of notes spawned.

Returns

The user's hit accuracy, in percent.

3.20.1.2 GoodHit()

```
float ScoreCalculator.GoodHit (
             int currentScore,
             int currentMultiplier )  [inline]
```

Called when a good hit has been made.

Parameters

currentScore	The current gameplay score.
currentMultiplier	The current gameplay multiplier.

Returns

1. Used to increment the number of good hits logged.

3.20.1.3 LongHit()

```
float ScoreCalculator.LongHit (
             int currentScore,
             int currentMultiplier )  [inline]
```

Called when a long hit has been made.

Parameters

currentScore	The current gameplay score.
currentMultiplier	The current gameplay multiplier.

Returns

1. Used to increment the number of long hits logged.

3.20.1.4 NormalHit()

```
float ScoreCalculator.NormalHit (
            int currentScore,
            int currentMultiplier )  [inline]
```

Called when a normal hit has been made.

Parameters

currentScore	The current gameplay score.
currentMultiplier	The current gameplay multiplier.

Returns

1. Used to increment the number of normal hits logged.

3.20.1.5 NoteMissed()

```
float ScoreCalculator.NoteMissed ( )  [inline]
```

Called when a note has been missed.

Returns

1. Used to increment the number of misses logged.

3.20.1.6 PerfectHit()

```
float ScoreCalculator.PerfectHit (
            int currentScore,
            int currentMultiplier )  [inline]
```

Called when a perfect hit has been made.

Parameters

currentScore	The current gameplay score.
currentMultiplier	The current gameplay multiplier.

Returns

1. Used to increment the number of perfect hits logged.

The documentation for this class was generated from the following file:

- /mnt/c/users/david/documents/everything/school/3xa3/3xa3__l01_gr16_project/src/Assets/Scripts/Score↩ Calculator.cs

3.21 SettingsMenu Class Reference

Inheritance diagram for SettingsMenu:

Collaboration diagram for SettingsMenu:

Public Member Functions

- void SetVolume ()

  Sets game volume based on the current slider value.
- void GreenPressed ()

  Green button to be rebound.
- void RedPressed ()

  Red button to be rebound.
- void YellowPressed ()

  Yellow button to be rebound.
- void BluePressed ()

  Blue button to be rebound.
- void PinkPressed ()

  Pink button to be rebound.

Public Attributes

- GameObject **uiGreen**
- GameObject **uiRed**
- GameObject **uiBlue**
- GameObject **uiYellow**
- GameObject **uiPink**
- Slider slider

  Volume slider.

The documentation for this class was generated from the following file:

- /mnt/c/users/david/documents/everything/school/3xa3/3xa3__l01_gr16_project/src/Assets/Scripts/Settings↩ Menu.cs

Index