Chart > note time in ms

note > rail (1-4) rail > board/play field > draw/camera

board/play field would likely do the accuracy calculations for notes hit/missed

Rail

-notes[] : Note

-nextNoteToPlay : int : (used to access the correct note in array)

+Rail()

+AddNoteToRail(note : Note) : void

+CheckNoteSpawn(timer : float) : void (sets note to active)

Song

-songTitle : string : loaded from file

-notes[] : Note : loaded from file

-songTime : int : (ms)

-songGraphic : graphic

-songAudio : audio

-previousSocre : float

-HighestScore : float

+Song()

+LoadAudioFile()

Note

+startTime : int : (ms)

+endTime : int : (ms)

+active : bool : (if timer >= start time & <=endTime)

+rail : int : (0-3)

+noteGraphic : graphic : (quad and texture)

User

-health : int

-keyPressed[4] : Rail

-score : ScoreManager

+checkPlayerHealth for all players;

+checkPlayerScore for all players and compare for winner

+checkRailPressed for each key press to determine hit

+update(dt)

PlayField

+static timer : float

-rail[4] : Rail

-scoreManager : ScoreManager

+PlayField(song : Song)

-CreateRails(notes[] : Note) : void

+Update(dt)

+CheckPlayerHit(railHit : int) : void

UPDATE METHOD:

Update delta time;

Update songtimer;

checkNoteSpawn for each rail;

playerInput;

checkPlayerHit (checks what rail they are interacting with, if CheckNoteSpawn returned true and if it did calculate accuracy;

CheckGameOver();

CheckSongEnd();

ScoreManager

-notesTotal : int

-notesHit : int

-combo : int

-score : int

-health : int

-percentageNotesHit : float

-notesMissed : int

+ScoreManager(int : notesTotal)

+IncrementNotesHit() : void

+GetNotesHit() : int

+IncrementCombo(): void

+ResetCombo() : void

+GetScore() : int

+GetPercentage() : float

+IncrementNotesMissed() : void

+CheckSongEnd() : bool

+Accuracy(accuracy : int) : void (method calls changes score based on accuracy and combo)

+CheckGameOver() : bool

+decrementHealth() : void

Flow of Program:

Start Menu

Load all songs from file

Player Input

Song menu, size depending on number of songs loaded.

Create Play field using selected song as a parameter

Player Input

Menu UML diagrams:

Song Menu

-songList[] : Song[]

-songSelector[] : SongMenuDisplay[]

-selectedSong : Song

-selectedPosition : int

-displayedSongs : Song[7]

+SongMenu()

+InitialiseSongList() : Song[] (does not load full song audio file)

+SearchForSong(name : string) : Song

+Draw() : void

+ScrollThroughMenu(keyInput) : void

DRAW METHOD:

Draws the song graphics for the 7 songs in the song selector on the left side of the screen, currently selected song will be drawn larger.

To right of song difficulty options displayed along with an attached score.

SongMenuDisplay

-song : Song

-positionInList : int

-

+SongMenuDisplay(song : Song)

+getSong() : Song

+getPositionInList() : int

+Draw()

DRAW METHOD:

Draw graphic

Start Menu

-backgroundGraphics : graphic

-logo : graphic

+StartMenu()

-PlayerStart() : void

(calls song menu initialise)