

Overview

Game: Music Mash

Type of Game: Tile/ sprite based game

Summary: This game is based on Candy Crush. However, instead of candy, we'll be using music album covers. When a match is made, a clip of a song from that album will play.

Classes:

- Album Cover Class
 - Stores pictures
 - Stores music
 - Play music method
- Visible Board Class (Controller/GUI)
 - Holds 2D array of Tile Class
- Tile Class
 - Holds an album cover
 - Facilitates switching
- Invisible Board (BackEnd, Lower board)
 - Keeps track of scoring
- Interface
 - numMoves
 - isValidMove
 - reset
 - getScore
 - getPosition

Scoring:

- See <http://candycrush.wikia.com/wiki/Score>

Songs:

That's What I Like by Bruno Mars 0.49- 1.05

<https://www.youtube.com/watch?v=CuFGzAmkoIA>

Castle on the Hill by Ed Sheeran

<https://www.youtube.com/watch?v=u4ZrjuZjzDI>

1.10-1.24

It Ain't Me by Selena Gomez and Kygo

.57-1.08 <https://www.youtube.com/watch?v=YuRAHS9EDY8>

How Far I'll Go - Alessia Cara

https://www.youtube.com/watch?v=i_W8Vfj6ZHA

.59-1.10

It Was Always You - Maroon 5

.28-.47 <https://www.youtube.com/watch?v=7rPZRLThTOg>

There's Nothing Holding Me Back - Shawn Mendes

<https://www.youtube.com/watch?v=A7xzXDStQnk>

.40-1.00

Despacito by Luis Fonsi, Justin Bieber. Daddy Yankee

1:03-1:15 <https://www.youtube.com/watch?v=SlxLJnHu99c>

Slide by Calvin Harris

0.36 - 0.45 0.54-1.04<https://www.youtube.com/watch?v=s2p1IMA6txE>

Class APIs

Grid	
Class Header abstract class Grid	
Class Summary The Grid class is an abstract class that can be implemented whenever a grid (a 2D array of objects) is needed in the game. <ul style="list-style-type: none">- <code>getRows()</code>, <code>getCols()</code>- <code>getPosition(Album a)</code>- <code>setLocation(Position p, Album a)</code>- <code>swap(Position a, Position b)</code>- <code>setScore(int value)</code>, <code>getScore()</code>, <code>incrementScore(int amt)</code>	
Grid() Creates a default Grid (but cannot be created by itself because it is abstract).	
Method Summary	
Return type	Method and description
int	getCols() Returns the number of columns.
int	getRows()

	Returns the number of rows.
Position	getPositionOfAlbum (Album a) Returns the grid position of this Album. Returns null if cannot find album.
void	setPosition (Position p, Album a) Sets the position of an Album on the Grid.
void	swap (Position a, Position b) Swaps the Albums at the given positions and updates the view and score.
int	getScore () Returns the score of the game at the current time.
int	incrementScore (int amt) Increments score by given amount. Returns the new score.
void	setGrid (int rows, int cols) Creates a grid with the designated size.
void	setScore (int score) Sets score to passed in score.
Album	getAlbumAt (Position p) Returns album at position p

Visible Grid
Class Header class VisibleGrid extends Grid
Class Summary The VisibleGrid is a subclass of Grid which will display the board with images; requires a corresponding InvisibleGrid. <ul style="list-style-type: none"> - getAlbum(Position p) - show(Position p), hide(Position p) - setAlbum(Position p, Album a) - getRows(), getCols() - setSize(int rows, int cols, int startRow, int startCol)
Constructors
VisibleGrid (int rows, int cols)

Creates a VisibleGrid with the given amount of rows and columns; the grid is completely empty.

VisibleGrid(int rows, int cols, int startRow, int startCol)

Creates a VisibleGrid with the given amount of rows and columns; the grid is filled starting with startRow and startCol from the corresponding InvisibleGrid.

Method Summary

Return type	Method and description
Album	getAlbum (Position p) Returns the Album at the given position.
void	setAlbum (Position p, Album a) Sets the specified position in the grid to the album passed in.
void	show (Position p) Displays the Album at the given position.
int	getRows() Returns the number of rows in the grid (which represents the visible region of the board)
int	getCols() Returns the number of cols in the grid (which represents the visible region of the board)
void	setSize (int rows, int cols, int startRow, int startCol) Sets the grid size to the number of rows and columns specified by rows and cols respectively and fills the grid with values from InvisibleGrid, starting from startRow and startCol.
void	hide (Position p) Hides the Album at the given position.

Invisible Grid

Class Header

```
class InvisibleGrid extends Grid
```

Class Summary

An InvisibleGrid is a subclass of Grid which contains all the lower board elements of the game.

- `shift(int col, int num, int beginAt)`
- `setVisibleRange(int leftBound, int rightBound, int upperBound, int lowerBound)`
- `getVisibleLeftBound(), getVisibleRightBound(), getVisibleUpperBound(), getVisibleLowerBound()`
- `setAlbum(Position p, Album a), getAlbum(Position p)`
- `isMatch(Position p1, Position p2)`
- `hasMatches()`

Constructors

InvisibleGrid(int rows, int cols)

Creates an InvisibleGrid with the given number of rows and columns.

InvisibleGrid(int rows, int cols, int leftBound, int rightBound, int upperBound, int lowerBound)

Creates an InvisibleGrid with the given number of rows and columns which also has a defined visible range specified by the bound parameters.

Method Summary

Return type	Method and description
void	shift (int col, int num) This method shifts down a column by the given number of spots, “deleting” the row specified by beginAtRow .
boolean	setVisibleRange (int leftBound, int rightBound, int upperBound, int lowerBound) Sets the bound values to the values specified in the parameters. Returns true if the bound values are all valid values. Returns false if one or more of the bound values is invalid and does not set any of the bound values to the parameter-specified values.
int	getVisibleLeftBound () Returns the column value of the left visible region boundary.
int	getVisibleRightBound () Returns the column value of the right visible region boundary.
int	getVisibleUpperBound () Returns the row value of the upper visible region boundary.
int	getVisibleLowerBound ()

	Returns the row value of the lower visible region boundary.
void	setAlbum(Position p, Album a) Sets the value of position p in the grid to album a
Album	getAlbum(Position p) Gets the album value of position p in the grid
boolean	isMatch(Position p1, Position p2) Returns true if swapping the albums at p1 and p2 is a match
boolean	hasMatches() Returns true if there are more matches available on the grid, and fall within the bounds of the visible range.
boolean	isGameOver() Determines if game is over.
int	getScoreThreshold() Returns score threshold that score must be greater than or equal to to win the game.
void	shuffle() Shuffles the grid.
boolean	hasWon() Determines whether the player has won or not
void	totalScore() Increments score after every match

Album
Class Header class Album extends ImageView
Class Summary Album is a class that represents one album cover that will be put into our 2D grids. This class keeps track of a audio, image, and name.

- playMusic(), setMusic(Parameter)
- getImage(), setImage(Image img)
- getAlbumName(), setAlbumName(String name)
- getArtist(), setArtist(String artist)

Constructors

Album (Image img, AudioClip audio, String name, String artist)
Creates an Album with the parameters passed in.

Method Summary

Return type	Method and description
void	playMusic() Plays music corresponding to this album.
AudioClip	getMusic() Return the song corresponding to this album.
void	setMusic (AudioClip clip) Sets the song to the param
Image	getImage() Returns the image of the album cover of the object.
Image	setImage (Image img) Sets the album cover image.
String	getArtist() Returns the name of the artist
void	setArtist (String artist) Sets the name of the artist
String	getAlbumName() Returns the name of the album cover.
void	setAlbumName (String name) Sets the name of the album cover.

Position

Class Header class Position	
Class Summary Position keeps track of the row and column of an object in a grid. <ul style="list-style-type: none"> - <code>getRow()</code>, <code>setRow(int row)</code> - <code>getCol()</code>, <code>setCol(int col)</code> 	
Constructors	
Position (int row, int col) Creates a Position object with the given row and column.	
Method Summary	
Return type	Method and description
int	getRow() Returns the row of the grid that the object is in.
void	setRow (int row) Sets the row to the parameter
int	getCol() Returns the column of the grid that the object is in.
void	setCol (int col) Sets the album cover image.

GameView	
Class Header class GameView	
Class Summary Represents the GUI that creates the actual game board. Allows for swaps (based on mouse click events), displays images, and deals with the front-end, graphical aspects of the game. <ul style="list-style-type: none"> - <code>updateBoard()</code> - <code>updateScore()</code> - <code>onMouseClicked()</code> - <code>onMouseDown()</code> - <code>lastAlbumCleared()</code> - <code>populateGrid()</code> 	

<ul style="list-style-type: none"> - setValue() - getValue() - playMusic() - calculateScore() 	
Constructors	
Not Applicable	
Method Summary	
Return type	Method and description
void	updateBoard() Updates the images in the GUI grid to the values in the InvisibleGrid
void	updateScoreImage() Updates the GUI image of the score
void	onMouseClicked() Responds to clicks on menu items
void	onMouseDown(int col) Responds to mouse drags used to swap two album covers
Album	lastAlbumCleared() Returns the last album that was cleared out on the grid (to be used to select the audio clip played)
void	populateGrid(Grid grid) Repopulates the grid to the values in the grid passed in. Updates InvisibleGrid to this grid.
void	setValue(Position pos, Album a) Sets the grid image at the position passed in to the album a.
Album	getValue(Position pos) Returns the album at the position passed in.
void	playMusic() Plays audio clip of the last album cleared
void	calculateScore() Calculates the new score after a swap is made (may need additional methods to fully implement)

void	transfer () Transfers values from InvisibleGrid to VisibleGrid
------	--