Hextris

Generated by Doxygen 1.8.13

Contents

1 Hierarchical Index **1**

1.1 Class Hierarchy 1

2 Class Index **3**

2.1 Class List 3

3 File Index **5**

3.1 File List 5

4 Class Documentation **7**

4.1 HexagonBehaviour Class Reference 7

4.2 LevelManager Class Reference 8

4.3 MusicPlayer Class Reference 8

4.4 PauseManager Class Reference 9

4.5 Spawner Class Reference 9

4.6 TrapezoidCollision Class Reference 10

4.7 TrapezoidTransform Class Reference 10

4.7.1 Member Function Documentation 11

4.7.1.1 Start() 11

4.7.1.2 Update() 11

5 File Documentation **13**

5.1 HexagonBehaviour.cs File Reference . . . 13

5.1.1 Detailed Description . . . 13

5.2 LevelManager.cs File Reference . . . 13

5.2.1 Detailed Description . . . 14

5.3 MusicPlayer.cs File Reference . . . 14

5.3.1 Detailed Description . . . 14

5.4 PauseManager.cs File Reference . . . 14

5.4.1 Detailed Description . . . 15

5.5 Spawner.cs File Reference . . . 15

5.5.1 Detailed Description . . . 15

5.6 TrapezoidCollision.cs File Reference . . . 15

5.6.1 Detailed Description . . . 16

5.7 TrapezoidTransform.cs File Reference . . . 16

5.7.1 Detailed Description . . . 16

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

MonoBehaviour
- HexagonBehaviour . . . 7
- LevelManager . . . 8
- MusicPlayer . . . 8
- PauseManager . . . 9
- Spawner . . . 9
- TrapezoidCollision . . . 10
- TrapezoidTransform . . . 10

Chapter 2

Class Index

2.1 Class List

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

HexagonBehaviour 7
LevelManager 8
MusicPlayer 8
PauseManager 9
Spawner 9
TrapezoidCollision 10
TrapezoidTransform 10

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

HexagonBehaviour.cs
: The purpose of this file is to implement the behaviour of the Hexagon, mostly the physical rotation of the Hexagon 13

LevelManager.cs
: The purpose of this file is to manage the loading of the game 13

MusicPlayer.cs
: The purpose of this file is to manage the music and make sure the music is playing 14

PauseManager.cs
: The purpose of this file is to implement the pause funcitonality of the game 14

Spawner.cs
: The purpose of this file is to implement the spawning functionality for the trapezoids 15

TrapezoidCollision.cs
: The purpose of this file is to detect collisions between a gameObject and either the center black hexagon, or with a child of the hexagon (i.e. a trapezoid that is already on top of the hexagon) . 15

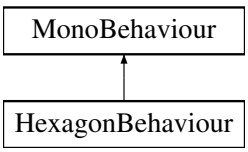
TrapezoidTransform.cs
: The purpose of this file is to define the behaviour for each color of trapezoid. This includes both the motion, scaling, and position of the trapezoids 16

Chapter 4

Class Documentation

4.1 HexagonBehaviour Class Reference

Inheritance diagram for HexagonBehaviour:



Public Attributes

- float **transitionSpeed** = 0.3f
- float **degree** = 0f
- bool **timeStay** = false

Private Member Functions

- void Start ()

  The purpose of this function is to initialize the hexagon game object. It finds the GameObject and binds it to blackHexagon.
- void Update ()

  The purpose of this function is to implement the rotation functionality of the hexagon. It checks if the gmae is paused, and if it isn't then it checks if the user pressed either the left or right arrow key. On the key press, it rotates the hexagon 60 degrees to the direction of the key.

Private Attributes

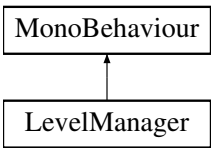
- GameObject **blackHexagon** = null

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

- HexagonBehaviour.cs

4.2 LevelManager Class Reference

Inheritance diagram for LevelManager:



Public Member Functions

- void LoadLevel (string name)

  The purpose of this function is to load the level and basically boot the program.
- void QuitRequest ()

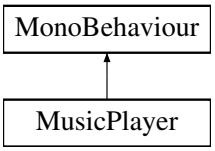
  The purpose of this function is to handle the quit function when it is requested.

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

- LevelManager.cs

4.3 MusicPlayer Class Reference

Inheritance diagram for MusicPlayer:



Private Member Functions

- void Awake ()

  This function loads and plays the music file.
- void **Start** ()
- void **Update** ()

Static Private Attributes

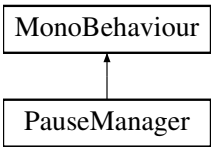
- static MusicPlayer **instance** = null

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

- MusicPlayer.cs

4.4 PauseManager Class Reference

Inheritance diagram for PauseManager:



Private Member Functions

- void Start ()

  The Start() function is not needed because no values need to initialized.
- void Update ()

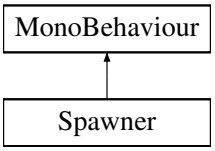
  The purpose of this function is to detect if the space-bar is pressed, and if it is pressed, then pause the game by setting the timeScale value to 0.

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

- PauseManager.cs

4.5 Spawner Class Reference

Inheritance diagram for Spawner:



Public Attributes

- GameObject [ ] **coloredTrapezoids**

Static Public Attributes

- static float **angle** = 0

Private Member Functions

- void Start ()

  The Start() function is used to initialize all the variables and GameObjects needed to spawn a trapezoid.
- void spawnOne ()

  This function randomly generates one trapezoid. It randomly chooses one of 6 places to spawn and it also randomly chooses the colour between the four colours.

Private Attributes

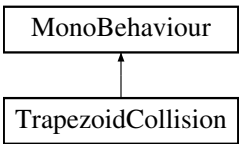
- GameObject **hexagon**
- GameObject **canvas**

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

- Spawner.cs

4.6 TrapezoidCollision Class Reference

Inheritance diagram for TrapezoidCollision:



Static Public Attributes

- static Vector3 **areaZero** = new Vector3(214.5f, 211.77f, 0f)

Private Member Functions

- void OnCollisionEnter2D (Collision2D coll)

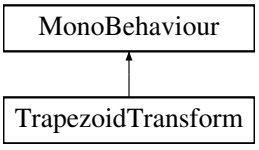
  The purpose of this function is to detect a collision between two different rigidbodies with colliders inside of the child rectangles for the particular trapezoid. This detects whether a collision with the black hexagon or any of the child trapezoids take place. If the gameObject that is hit is the black hexagon, the gameObject that this script is referring to will become a child of the hexagon. If not, then the.

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

- TrapezoidCollision.cs

4.7 TrapezoidTransform Class Reference

Inheritance diagram for TrapezoidTransform:



Public Member Functions

- void **LeftSideStick** (GameObject leftSide)
- void **RightSideStick** (GameObject rightSide)

Public Attributes

- int **side** = 0
- int **row** = 0
- bool **move** = true
- float **pixelsPerUnit** = 100f
- float **rectangleHeightPixels** = 271f
- float **rectangleWidthPixels** = 373f
- float **hexagonHeightPixels** = 480f
- float **scalePerUnit** = 0
- float **halfHexagonWorldUnit** = 0
- float **hexagonSideWorldUnit** = 0
- float **rectangleScaleOnSide** = 0
- string **objectType**

Private Member Functions

- void Start ()

  This is the intial setup the script does. One time run.
- void Update ()

  This runs every frame.

Private Attributes

- GameObject **rectangle**
- GameObject **leftSide**
- GameObject **rightSide**
- GameObject **hexagon**

4.7.1 Member Function Documentation

4.7.1.1 Start()

```
void TrapezoidTransform.Start ( )  [inline], [private]
```

This is the intial setup the script does. One time run.

The start method, which instantiates all trapezoid gameObjects with specific positions relative to the player.

4.7.1.2 Update()

```
void TrapezoidTransform.Update ( )  [inline], [private]
```

This runs every frame.

The update function defines the scaling factor that is required for every world unit of distance the rectangle is away from the player hexagon. It also defines the position of the two side triangles of yellow trapezoid to ensure that it they are always to the left and right of the rectangle.

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

- TrapezoidTransform.cs

Chapter 5

File Documentation

5.1 HexagonBehaviour.cs File Reference

The purpose of this file is to implement the behaviour of the Hexagon, mostly the physical rotation of the Hexagon.

Classes

- class HexagonBehaviour

5.1.1 Detailed Description

The purpose of this file is to implement the behaviour of the Hexagon, mostly the physical rotation of the Hexagon.

Author

Jason Li, Yousaf Shaheen, Scott Williams

Date

November 8th, 2017

5.2 LevelManager.cs File Reference

The purpose of this file is to manage the loading of the game.

Classes

- class LevelManager

5.2.1 Detailed Description

The purpose of this file is to manage the loading of the game.

Author

Jason Li, Yousaf Shaheen, Scott Williams

Date

November 8th, 2017

5.3 MusicPlayer.cs File Reference

The purpose of this file is to manage the music and make sure the music is playing.

Classes

- class MusicPlayer

5.3.1 Detailed Description

The purpose of this file is to manage the music and make sure the music is playing.

Author

Jason Li, Yousaf Shaheen, Scott Williams

Date

November 8th, 2017

5.4 PauseManager.cs File Reference

The purpose of this file is to implement the pause funcitonality of the game.

Classes

- class PauseManager

5.4.1 Detailed Description

The purpose of this file is to implement the pause funcitonality of the game.

Author

Jason Li, Yousaf Shaheen, Scott Williams

Date

November 8th, 2017

5.5 Spawner.cs File Reference

The purpose of this file is to implement the spawning functionality for the trapezoids.

Classes

- class Spawner

5.5.1 Detailed Description

The purpose of this file is to implement the spawning functionality for the trapezoids.

Author

Jason Li, Yousaf Shaheen, Scott Williams

Date

November 8th, 2017

5.6 TrapezoidCollision.cs File Reference

The purpose of this file is to detect collisions between a gameObject and either the center black hexagon, or with a child of the hexagon (i.e. a trapezoid that is already on top of the hexagon).

Classes

- class TrapezoidCollision

5.6.1 Detailed Description

The purpose of this file is to detect collisions between a gameObject and either the center black hexagon, or with a child of the hexagon (i.e. a trapezoid that is already on top of the hexagon).

Author

Jason Li, Yousaf Shaheen, Scott Williams

Date

November 8th, 2017

5.7 TrapezoidTransform.cs File Reference

The purpose of this file is to define the behaviour for each color of trapezoid. This includes both the motion, scaling, and position of the trapezoids.

Classes

- class TrapezoidTransform

5.7.1 Detailed Description

The purpose of this file is to define the behaviour for each color of trapezoid. This includes both the motion, scaling, and position of the trapezoids.

Author

Jason Li, Yousaf Shaheen, Scott Williams

Date

October 26, 2017