Fortune Wheel Doc

Generated by Doxygen 1.9.2

Chapter 1

Welcome to Fortune Wheel Documentation!

This is your guide to use the tool efficeintly.

1.0.1 Sections:

- System Overview
- Setup Steps
- API References

1.1 System Overview

The tool helps you create fortune wheel easily for your game to satisfy a specific functionality by manipulating every detailed aspect.

1.1.1 Package Dependencies

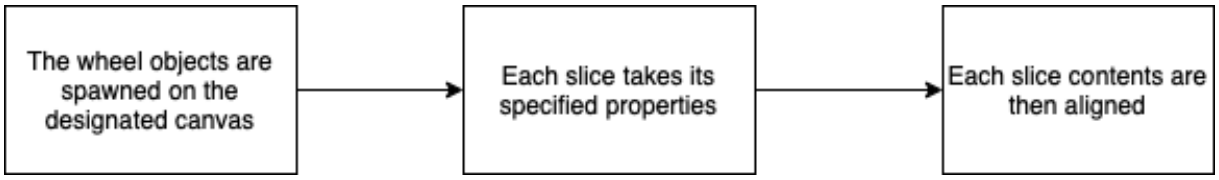
- TextMeshPro

1.1.2 Technical features

- Add unlimited number of components to the wheel
- Every component has:
  - Label
  - Icon
  - Fill color
- Control wheel speed
- Choose and make you own wheel pointer style
- Some assets included

1.1.3 How does it work?

The below figure illustrates how it works.



Work flow of the script

1.1.4 Wheel behavior:

When the wheel script is added, a custom inspector is created for a better work experience. The wheel contains two slices at least. Note that the wheel only works on canvas.
If you want to try out the changes, just run the game and see your customizations.

1.2 Setup Steps

You can create a new fortune wheel just straight forward. After you create a Canvas, create a new empty Game←Object under it and follow along.

Steps to make a wheel

Note that the effects while changing properties in inspector aren't real-time, you have to run the game, so don't worry if you don't get any feedback from your changes, feedback system is coming soon.

1.2.1 Make it via script

An alternative of creating it from the inspector is to write a script. Please refer to the **FortuneWheel** (p. **10**) constructor.

1.3 API References

Please refer to the Chapter 2 to get started.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

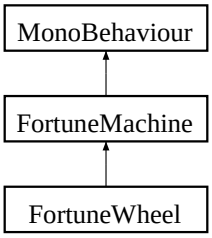
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

Class Documentation

4.1 FortuneMachine Class Reference

The parent class that **FortuneWheel** that inherits from and implements its functions. Inheritance diagram for FortuneMachine:



Public Member Functions

- abstract void **CreateFortune** ()

  *Abstract function responsible for instantiating one of the **FortuneMachine** (p.**8**) and its components.*

- abstract IEnumerator **StartFortune** ()

  Starts moving the FortneMachine. The machine then starts to decelarate gradually until it stops on the selected choice. This function works asynchronously so call it in a coroutine.

  Exmaple:

- abstract string **GetLatestResult** ()

  Get the label result of the latest fortune spin.
  You can use if after you make at least one spin.

- abstract int **GetLatestResultID** ()

  Get the ID result of the latest fortune spin.
  You can use if after you make at least one spin.

Public Attributes

- bool **doubleChance** = false

  If you want to double the instances number of the elements to get the selection chance doubled. Set to false by default.

4.1.1 Detailed Description

The parent class that **FortuneWheel** (p. **10**) that inherits from and implements its functions.

4.1.2 Member Function Documentation

4.1.2.1 CreateFortune() abstract void FortuneMachine.CreateFortune ( ) [pure virtual]

Abstract function responsible for instantiating one of the **FortuneMachine** (p. **8**) and its components. Implemented in **FortuneWheel** (p. **10**).

4.1.2.2 GetLatestResult() abstract string FortuneMachine.GetLatestResult ( ) [pure virtual] Get

the label result of the latest fortune spin.

You can use if after you make at least one spin.

Returns

The label of the result.

Implemented in **FortuneWheel** (p. **10**).

4.1.2.3 GetLatestResultID() abstract int FortuneMachine.GetLatestResultID ( ) [pure virtual]

Get the ID result of the latest fortune spin.

You can use if after you make at least one spin.

Returns

The ID of the result.

Implemented in **FortuneWheel** (p. 10).

4.1.2.4 StartFortune() abstract IEnumerator FortuneMachine.StartFortune ( ) [pure virtual]

Starts moving the FortneMachine. The machine then starts to decelarate gradually until it stops on the selected choice. This function works asynchronously so call it in a coroutine.

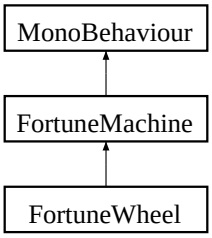
Exmaple:

```
IEnumerator CoroutineExamle() { yield return StartCoroutine(
StartFortune() ); }
```

Implemented in **FortuneWheel** (p. **10**).

4.2 FortuneWheel Class Reference

Create and customize your own Fortune Wheel. Inheritance diagram for FortuneWheel:



Public Types

- enum **PointerPosition** {
  Top , **Botton** , **Right** , **Left** ,
  Center }

Public Member Functions

- override void **CreateFortune** ()

  *Abstract function responsible for instantiating one of the **FortuneMachine** (p.**8**) and its components.*

- override IEnumerator **StartFortune** ()

  Starts moving the FortneMachine.
  The machine then starts to decelarate gradually until it stops on the selected choice.
  This function works asynchronously so call it in a coroutine.

- override string **GetLatestResult** ()

  Get the label result of the latest fortune spin.
  You can use if after you make at least one spin.

- override int **GetLatestResultID** ()

  Get the ID result of the latest fortune spin.
  You can use if after you make at least one spin.

- **FortuneWheel** ()

  Create a default instance from the fortune wheel, two slices minimum.

- **FortuneWheel** (**WheelElement**[ ] elements, float **wheelSpeed**=200, float **wheelSize**=3, bool **double**←-
  Chance=false, float labelOrientation=0, float **labelOffset**=0.5f, float **iconOffset**=0.5f, float **slice**←**Thickness**=0, PointerPosition pointerPosition=PointerPosition.Top, float pointerSize=1) *A much more detailed constructor to make a **FortuneWheel** (p.**10**).*

Public Attributes

- float **wheelSize** = 3f

  Scale size of the wheel.

- float **wheelSpeed** = 200f

  Controller of the wheel rotating speed.

- float **sliceThickness** = 0f

  The outline thickness around each slice in the wheel.

- float **labelRotation** = 0f

  Specify the orientation of labels on wheel.
- float **iconOffset** = 2f

  The offset of the icons on wheel from the origin till the end of the wheel.
- float **labelOffset** = 1f

  The offset of the labels on wheel from the origin till the end of the wheel.
- float **pointerScaleSize** = 1 *The scale size of the wheel pointer.*
- bool **createOnStart** = true

  Create the wheel automatically at the start of the game.
- Color **sliceOutlineColor** = new Color(1, 1, 1, 1)
- Sprite **pointerIcon**
- TMP_FontAsset **font**
- int **pointerPositionIndex** = 0

  The postion of the wheel pointer. Note that the pointer doesn't change its rotation relative to its postion.
- List< **WheelElement** > **slices**

  The elements in the wheel.

4.2.1 Detailed Description

Create and customize your own Fortune Wheel.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 FortuneWheel()

FortuneWheel.FortuneWheel (
WheelElement[] *elements,* float *wheelSpeed = 200,* float *wheelSize = 3,*
bool *doubleChance = false,* float *labelOrientation = 0,* float *labelOffset = 0.5f,* float *iconOffset = 0.5f,* float *sliceThickness = 0,*
PointerPosition *pointerPosition = PointerPosition.Top,*
float *pointerSize = 1*) [inline]

A much more detailed constructor to make a **FortuneWheel** (p. **10**).

Parameters

wheelSpeed	Controller of the wheel rotating speed.
elements	The elements in the wheel.
wheelSize	Specify the scale size of the wheel.
doubleChance	If you want to double the instances number of the elements to get the selection chance doubled.
labelOrientation	Specify the orientation of labels on wheel.
labelOffset	The offset of the labels on wheel from the origin till the end of the wheel.
iconOffset	The offset of the icons on wheel from the origin till the end of the wheel.
sliceThickness	The outline thickness arround each slice in the wheel.

pointerPosition	The postion of the wheel pointer. Note that the pointer doesn't change its rotation relative to its postion.
pointerSize	The scale size of the wheel pointer.

4.2.3 Member Function Documentation

4.2.3.1 CreateFortune()

override void FortuneWheel.CreateFortune ( ) [inline], [virtual]

Abstract function responsible for instantiating one of the **FortuneMachine** (p. **8**) and its components. Implements **FortuneMachine** (p. **8**).

4.2.3.2 GetLatestResult()

override string FortuneWheel.GetLatestResult ( ) [inline], [virtual] Get

the label result of the latest fortune spin.

You can use if after you make at least one spin.

Returns

The label of the result.

Implements **FortuneMachine** (p. **8**).

4.2.3.3 GetLatestResultID()

override int FortuneWheel.GetLatestResultID ( ) [inline], [virtual] Get

the ID result of the latest fortune spin.

You can use if after you make at least one spin.

Returns

The ID of the result.

Implements **FortuneMachine** (p. **8**).

4.2.3.4 StartFortune()

override IEnumerator FortuneWheel.StartFortune ( ) [inline], [virtual]

Starts moving the FortneMachine. The machine then starts to decelarate gradually until it stops on the selected choice. This function works asynchronously so call it in a coroutine.

Exmaple:

```
IEnumerator CoroutineExamle()

{ yield return StartCoroutine( StartFortune() ); }
```

Implements **FortuneMachine** (p. **8**).

4.3 WheelElement Class Reference

The unit object of the wheel elements. Contains all
the slice properties.

Public Attributes

- string **label**

  The label written on the wheel element. This label is also used as a reference for the wheel element. Default value is "New Slice".

- Color **labelColor**

  The color of the label on the wheel.
  Default color is white.

- float **fontSize**

  The font size of the wheel label. It ranges between 4 and 10.

- bool **showLabelInWheel** = true

  Determines whether the label will be visible on the wheel element or not.
  Default value is true.

- Color **fillColor**

  The main color of the slice.
  Default color is black.

- Sprite **icon**

  The icon on the wheel element.
  This icon size is adapted to the wheel element size with few controls in the size. If you leave it blank, it will not be drawn.

4.3.1 Detailed Description

The unit object of the wheel elements.
Contains all the slice properties.