Charty Documentation

Hello dear Charty user! In this document I will describe your abilities with Charty. If you have found a bug, or have a suggestion or question, don't hesitate to contact me on official Unity Community forum.

Contents:

1) Vocabulary:

Some things you need to know about Charty first before starting using it.

- Sector – one part(arc) of the diagram.

- Inner radius – the minimum radius of the diagram. Defines starting point to draw.

- Outer radius – the maximum radius of the diagram. Defines ending point to draw.

- Pointers – small lines that go from sectors. Can be customized and show information about sector.

- Infobox – a table of data for your diagram. Shows the data and describes which color is the representation of this data on the diagram.

2)Usage;

To use Charty in your project just make a variable of type “Charty”, and then add Charty component to it. For example:

```
private Charty diagram;

void Start(){
    diagram = gameObject.AddComponent(“Charty”) as Charty;
}
```

You can find an example scene with example script of usage.

3)Variables:

bool outline – defines if outline is enabled.

bool pointers – defines if pointers are enabled.

bool infobox – defines if infox is enabled.

Font pointerFont – the font that pointers use to show information

Font infoFont – the font that infobox uses to show information;

int pointerFontSize – size of the Font that pointers use.

int infoFontSize - size of the Font that infobox uses.

int pointerLength – the length of the pointers.

Color pointerFontColor – color of the pointers font.

Color infoFontColor– color of the infobox font.

string infoTitle - the title used for infobox.

string pointermode – defines what pointers are showing. Has 3 possible options: “percents”, “values”, and “names”;

string infoToShow– defines what infobox is showing. Has 3 possible options: “percents”, “values”, and “names”;

string infoMode – defines either to use per_rows or per_columns variable. Has 3 possible options: “hor”(horizontal), “ver”(vertical);

int per_rows - defines how many sectors are shown in one row of an infobox.

int per_columns - defines how many sectors are shown in one columns of an infobox.

4)Methods:

4.1)Sectors:

AddSector:

```
void  AddSector(string name, float value, Color theColor);
void  AddSector(string name, int value, Color theColor);
void  AddSector(string name, float value);
void  AddSector(string name, int value);
```

Adds a sector to a diagram with specified name, value and color to the diagram.

RemoveSector:

```
void  RemoveSector(string name);
void  RemoveSector(float value);
void  RemoveSector(int value);
```

Removes sector from the diagram with specified name or value.

RemoveSectorByIndex:

```
void  RemoveSectorByIndex(int index)
```

Removes sector from the diagram with specified index;

ClearSectors;

```
void  ClearSectors();
```

Removes all sectors and stops displaying the diagram.

GetIndex;

```
int GetIndex(string name);
int GetIndex(float value);
int GetIndex(int value);
int GetIndex(Color myColor);
```

Returns the index of a sector with given name, value or color.

GetPercent;

```
float GetPercent(int index);
```

Returns the percents of sector with given index.

GetColor;

```
Color GetColor(int index);
```

Returns the color of a sector with given index.

GetValue;

```
float GetValue(int index);
```

Returns the value of a sector with given index.

GetName;

```
string GetName(int index);
```

Returns the name of a sector with given index.

SetColor;

```
void SetColor(Color newCol, int index);
```

Sets the color of a sector with given index.

SetValue;

```
void SetValue(int newVal, int index);
void SetValue(float newVal, int index);
```

Sets the value of a sector with given index.

SetName;

```
void SetName(int index);
```

Sets the name of a sector with given index.

MakeEqual:

```
void  MakeEqual(int count);
```

Will create sectors with equal size sectors of desired amount. Max limit is set 60;

MakeRandom:

```
void  MakeEqual(int count);
```

Will create sectors with random size of desired amount. Max limit is set 60;

SectorsAmount()

```
void Sectors Amount();
```

Returns the amount of the sectors;

GetSum()

```
float GetSum();
```

Returns the sum of all values of the sectors;

4.2)Making the diagram:

MakeDiagram;

```
void  MakeDiagram(int innerRad, int outerRad, int x, int y);
void  MakeDiagram(int innerRad, int outerRad, float x, float y);
```

Will draw the diagram with given inner radius, outer radius, and position, **based on a sectors added before.**

Redraw

```
void Redraw()
```

Will redraw the diagram with already defined inner and outer radiuses, and position.

SetPos

```
void  SetPos(int x, int y);

void  SetPos(float x, float y);

void  SetPos(Vector2 newPos);
```

Used to set the center of diagram.

SetInnerRadius

```
void SetInnerRadius(int rad)
```

Will set the inner radius of the diagram.

SetOuterRadius

```
void SetOuterRadius(int rad)
```

Will set the outer radius of the diagram.

GetCenter()

```
Vector2  GetCenter();
```

Returns center of the diagram.

GetInnerRadius()

```
int GetInnerRadius();
```

Returns inner radius of the diagram.

GetOuterRadius()

```
int GetOuterRadius();
```

Returns outer radius of the diagram.

4.3)Rotation:

RotateCW

```
void RotateCW()
void RotateCW(int value)
```

Rotates Diagram clockwise either one time or desired amount of times.

RotateCCW

```
void RotateCCW()
void RotateCCW(int value)
```

Rotates Diagram counter clockwise either one time or desired amount of times.

SetRotation

```
void SetRotation(int angle)
```

Used to directly set the rotation angle of the diagram.

GetRotation

```
void GetRotation()
```

Returns the current rotation angle of the diagram;

4.4)Movement:

MoveWithRadius

```
void MoveWithRadius(int index, int value);
```

This method is used move the arc with given index with desired amount of units also increasing, decreasing it's size.

Negative amount will move it to the side of Center.

MoveArc

```
void MoveMoveArc(int index, int value, int inLimit, int outLimit);
```

This method is used move the arc with given index with desired amount of units without changing size of the arc.

Negative amount will move it to the side of Center.

4.5)Point on an arc:

Contains

```
int Contains(Vector2 myvec)
```

Will return the index of arc that contains the given point. Will return -1 if no arc contains it.

ArcContains

```
bool ArcContains( Vector2 myvec, int index)
```

Will return true if arc with given index contains given point.

4.6)Outline:

SetOutlineInnerWidth;

```
void SetOutlineInnerWidth(int width)
```

Will set the inner width of outline;

SetOutlineOuterWidth;

```
void SetOutlineOuterWidth(int width)
```

Will set the outer width of outline;

SetOutlineColor;

```
void SetOutlineColor(Color c)
```

Will set the color of outline;

GetOutlineInnerWidth;

int GetOutlineInnerWidth(int width)

Returns the inner width of outline;

GetOutlineOuterWidth;

int GetOutlineOuterWidth(int width)

Returns set the outer width of outline;

GetOutlineColor;

Color GetOutlineColor(Color c)

Returns the color of outline;

4.7)Infobox:

SetInfoBoxPos()

void SetInfoBoxPos(Vector 2 myVec);

Sets the position of infobox.

GetInfoBoxPos()

Vector 2 GetInfoBoxPos(Vector 2);

Returns the position of infobox;

5)Important info:

The maximum sector limit is set to 60 (diagram often looks messy with such amount).

Diagram does not look well on a low antialiasing level.

If you face this problem just use native unity command:

```
QualitySettings.antiAliasing = 8;
```

Hope you will enjoy using Charty! Good luck!