EasyLayout Documentation

Release 1.2.1

Ilia Novikov

Aug 27, 2019

Contents

CHAPTER 1

EasyLayout

EasyLayout provides different layouts that not available with default layout groups.

- Main Axis

  Determine how elements will be placed.

- Layout Type

  - `Compact`: Compactly places the elements.
  - `Grid`: Places elements in the grid. Cell size is not fixed and depend on elements sizes in the same row and column.
  - `Flex`: Places elements like CSS flexbox layout.
  - `Staggered`: Places elements one-by-one to the shortest column or row depending on the main axis.

- Group Position

  Only for the `Compact` and `Grid` layouts.

  Combination of horizonal (`Left`, `Center`, `Right`) and vertical (`Upper`, `Middle`, `Lower`) positions.

  Elements combine to the group, this option specifies group position relative to the parent.

- Row Align

  Only for the `Compact` layout.

  Element position in the row (`Left`, `Center`, `Right`).

- Inner Align

  Only for the `Compact` layout.

  Column position relative to the group (`Top`, `Middle`, `Bottom`).

- Compact Constraint

  Only for the `Compact` layout.

  - `Flexible`: Rows and columns count depends on the parent size.
  - `Max Column Count`
  - `Max Row Count`

- Compact Constraint Count

  Only for the `Compact` layout.

  Max count of the rows or columns for the `Compact Constraint` option.

- Cell Align

  Only for the `Grid` layout.

  Elements position relative to the cell size. Same as `Group Position`.

- Grid Constraint

  Only for the `Grid` layout.

  - `Flexible`: Rows and columns count depends on the parent size.
  - `Fixed Column Count`
  - `Fixed Row Count`

- Grid Constraint Count

  Only for the `Grid` layout.

  Count of the rows or columns for the `Grid Constraint` option.

- Flex Setting

  Only for the `Flex` layout.

  - Wrap

    If disabled elements will all placed onto one line (row or column).

  - Justify Content

    Alignment along the main axis. Also distribute extra free space on the main axis.

    - `Start`: elements placed at the start of the line.
    - `Center`: elements placed at the center of the line.
    - `End`: elements placed at the end of the line.
    - `Space Between`: first element at the start of the line, last element at the end of the line, other elements placed between them with evenly spacing.
    - `Space Around`: first and last elements are placed with *1n* space from the edges, other elements placed with *2n* space between them.
    - `Space Evenly`: elements are placed so that the spacing between any two element and the space to the edges is equal.

  - Align Content

    Alignment of the lines (columns or rows) along the cross axis. Also distribute extra free space on the cross axis.

    - `Start`: lines placed to the start of the parent.
    - `Center`: lines placed to the center of the parent.

- `End`: lines placed to the end of the parent.
- `Space Between`: first line to the start of the parent, last line to the end of the parent, other lines placed between them with evenly spacing.
- `Space Around`: first and last lines are placed with *1n* space from the edges, other lines placed with *2n* space between them.
- `Space Evenly`: line are placed so that the spacing between any two lines and the space to the edges is equal.

- Align Items

  Define how elements are placed out along the cross axis on the line (column or row).

  - `Start`
  - `Center`
  - `End`

- Staggered Settings

  Only for the `Staggered` layout.

  - Fixed Block Count

    Count of the rows or columns.

  - Blocks Count

- Spacing

  Empty space between elements.

  Can be more than specified value for `Flex` layout.

- Symmetric

  Use symmetric margin.

- Margin

  Empty space from parent edges.

- Skip Inactive

  Do not reserve space for disabled elements.

- Right To Left

  The order of placement of elements.

- Top To Bottom

  The order of placement of elements.

- Children Width

  - `Do nothing`: do not resize elements.
  - `Set Preferred`: set element width to `Preferred Width`.
  - `Set Max From Preferred`: set maximum of the `Preferred Width` from the all elements.
  - `Fit Container`: increase elements width proportionally `Flexible Width` to fit parent width.
  - `Shrink On Overflow`: decrease elements width if summary width more than parent width with margin.

- Children Height

  Similar to `Children Width`

- Settings Changed

  Event, raised after any setting was changed.

CHAPTER 2

Layout Switcher

Allow to create different layouts with same GameObjects for different screen sizes and aspect ratios. Used when anchros, pivots and layout group not enough to create layout with different aspect ratios support.

Save values of the position, size, anchors, pivot, rotation, scale, active/disable state for each layout.

Options

- *Objects*: list of the controlled objects.
- *Default Display Size* (inches): display size to use when actual display size cannot be detected.
- *Layouts*: list of the layouts.
  - *Name*: layout name.
  - *Aspect Ratio*: aspect ratio for this layout.
  - *Max Display Size* (inches): maximum size of the display for this layout (layout will not be used if display size more than specified).

CHAPTER 3

Changelog

3.1 Release 1.2.1

- fixed FitContainer option

3.2 Release 1.2.0

- added Flex layout type
- added Staggered layout type
- renamed Stacking to MainAxis
- reduced memory allocations
- EasyLayout namespace renamed to EasyLayoutNS to avoid problems with Unity 2018.2 and later
- fixed "dirty" scene bug when using FitContainer or ShrinkOnOverflow
- added Shrink on Overflow option
- added CompactConstraint and CompactConstraintCount options
- added row/column constraint for Grid layout

3.3 Release 1.1.1

- improved compatibility with Unity 2017.x

3.4 Release 1.1.0

- bug fixes
- performance improvements

3.5 Release 1.0.5

- bug fixes.

3.6 Release 1.0.4

- improved performance
- bug fixes.

3.7 Release 1.0.3

- renamed to EasyLayout