



Steampunk UI

Documentation



Hello dear developer!

Thank you for acquiring this asset pack from the Unity Asset Store!

We are a growing art outsourcing agency and are very thankful for your support!
If you like this asset pack, please consider leaving a review on the Unity Asset Store or
recommend us to your friends! This would help us greatly!

If you encounter problems of any kind, feel free to reach out!
We will help you further.

Jacky Martin
CEO - Gentleland
jacky@gentleland.net

Summary

Summary	2
Content listing	3
Scripts Documentation	4
BarsFillAnimations.cs	4
CharakterStat.cs	4
ClockSlider.cs	4
CursorSelector.cs	5
KeyFieldUnSelect.cs	5
Gentleland Settings	6



Content listing

❖ Steampunk UI ❖ include 100+ hand-painted elements to craft your Steampunk UI.
Buttons, Icons, Healthbars, mouse cursors and much more.
Perfect for any games in a steampunk vibe.

Like our package? Please let us a [review](#) and give us feedback.

among the ui elements there are :

- 20 Buttons(with 3 states: normal, highlighted and pressed)
- 34 Button icons(with 4 states: normal, highlighted, pressed and disabled)
- 19 equipment icons
- 16 sliders and bars
- 16 UI frames
- 14 different mouse cursors
- 87 ready to use prefabs
- 12 demo and example scenes

and much more ...

All sprites are delivered on a transparent background as png files.

We included 1 custom made sprite font and 4 fonts under a free license allowing commercial use that we use for demoing purposes. See files in the folder "ThirdPartyLicenses" in the package for details.

We also included 5 free to use for personnel & commercial CCO Environment maps(HDRIs) from <https://hdri-haven.com/> and <https://ambientcg.com/> for demoing purposes.

Scripts Documentation

BarsFillAnimations.cs

Is a class that is used to make sliders, filling bars and circles move in the “Demo Filling Bars And Circles” scene.

It has 4 member variables :

- ★ `Slider[] sliders;` is an array referencing all sliders in the scene.
- ★ `float[] shifts;` is an array containing a shift by slider which allows them to start at different % values.
- ★ `float[] fillTimeInSeconds;` is an array containing the time each slider takes to go from 0% to 100%.
- ★ `float[] durations;` is an array that contains all possible durations in seconds that can be assigned to each slider at startup.

It has 2 functions :

- ★ `void Start()` is a builtin function that is called to initialize MonoBehaviours.
- ★ `void Update()` is a builtin function that is called each frame to update MonoBehaviours.

CharacterStat.cs

Is a class that is used to modify character stats in the Character Selection example scene.

It has 2 member variables :

- ★ `int value;` is the value of the current stat.
- ★ `TextMeshProUGUI textMesh;` is a reference to the TextMeshProUGUIComponent that displays the value.

It has 2 functions :

- ★ `void Start()` is a builtin Editor-only function that is called when a value is changed on the Inspector.
- ★ `void Increment()` is a function used to increment `value`.
- ★ `void Decrement()` is a function used to decrement `value`.

ClockSlider.cs

Is a class that is used to make the arrow of the clock slider move along with the filling circle.

It has 3 member variables :

- ★ `float minZRotation;` is the minimum rotation value of the Z axis.
- ★ `float maxZRotation;` is the maximum rotation value of the Z axis.
- ★ `float m_value;` is the value of the slider

It has 2 functions :

- ★ `void OnValidate()` is a builtin function that is called to initialize MonoBehaviours.
- ★ `void UpdateRotation()` is a function used to set the transform.rotation between `minZRotation` and `maxZRotation` depending on `m_value`.

CursorSelector.cs

Is a class that is used to set the cursor to the texture clicked in the “Demo Cursors” scene.

It has 2 member variables :

- ★ `Vector2 hotspot;` is the offset from the top left of the texture to use as the target point.
- ★ `Texture2D cursorTexture;` is the cursor texture to be set when clicked..

It has 2 functions :

- ★ `void Awake()` is a builtin function that is called to initialize MonoBehaviours.
- ★ `public void OnPointerClick(PointerEventData eventData)` is a function from IPointerClickHandler interface that is called when the UI element is clicked.

KeyFieldUnSelect.cs

Is a class that is used to unselect a KeyInputField in the Settings Menu example scene.

It has 1 member variable :

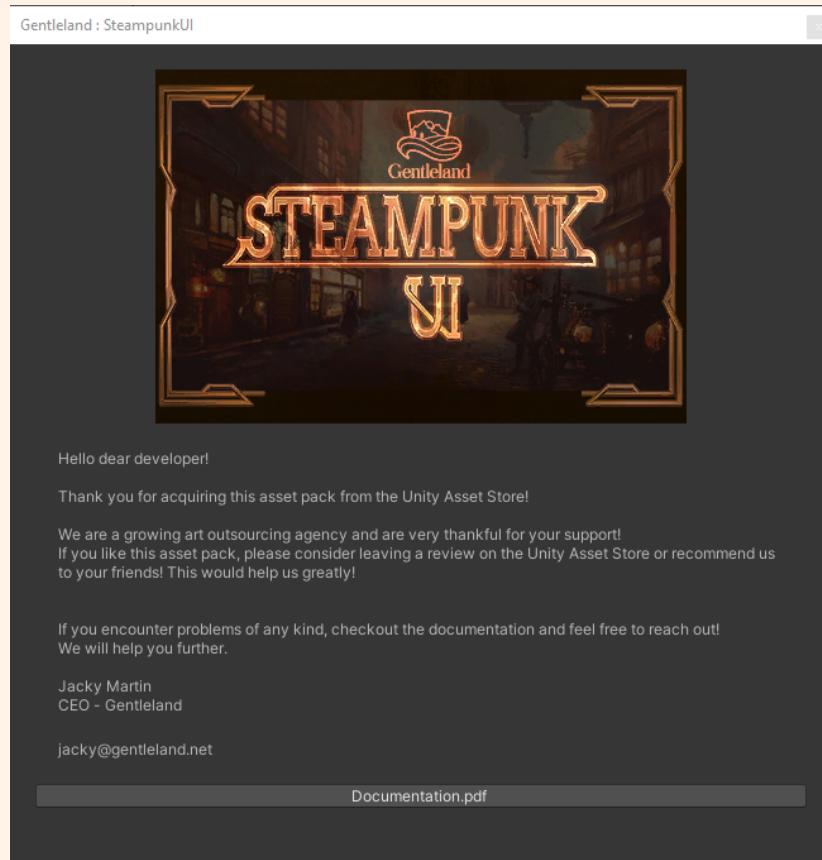
- ★ `Selectable selectable;` is the selectable that will be selected to unselect the KeyInputField

It has 1 function :

- ★ `public void UnSelect()` is the function called to unselect the KeyInputField and select the `selectable` member.

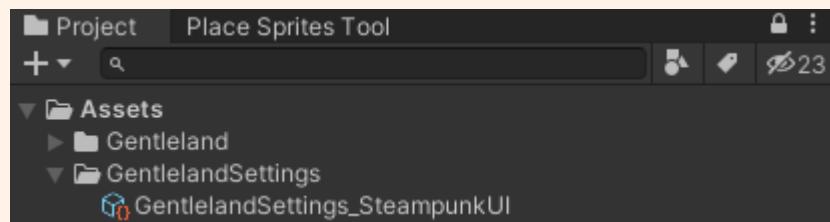
Gentleland Settings

As you will probably notice when you import the asset a window shows up.



Welcome window

At first load we create a folder called “GentlelandSettings” containing a file called “GentlelandSettings_SteampunkUI.asset” which tracks whether it is the first time the asset is loaded or.



Project window with the GentlelandSettings folder

If you delete this folder it will be recreated. The proper way to get rid of it is to delete the Utils folder under ‘Gentleland/SteampunkUI/Utils’ and then delete the “GentlelandSettings” folder.