# **GUI DESIGN – Ah My 2 Legs**

**By:** Daniel Sohler

Contents

[GUI DESIGN – Ah My 2 Legs 1](#_Toc69103900)

[Document overview 3](#_Toc69103901)

[Project GUI Requirements 3](#_Toc69103902)

[Leading GUI requirements 3](#_Toc69103903)

[UI Library Usage 3](#_Toc69103904)

[Itemised GUI elements 3](#_Toc69103905)

[Identify GUI & UI Events 3](#_Toc69103906)

[UI Events 4](#_Toc69103907)

[GUI Events 4](#_Toc69103908)

[Design GUI 4](#_Toc69103909)

[GUI Design Goals 4](#_Toc69103910)

[Identify GUI structure 4](#_Toc69103911)

[Initial GUI layout 4](#_Toc69103912)

[Design GUI Functionality 5](#_Toc69103913)

[<Energy Meter> 5](#_Toc69103914)

[Implement & Iterate GUI Design 5](#_Toc69103917)

[Initial layout Implementation 5](#_Toc69103918)

[Functionality Review 6](#_Toc69103919)

[External Feedback 6](#_Toc69103920)

[Planned Response to Review & Feedback 6](#_Toc69103921)

[Iterated Implementation 6](#_Toc69103922)

Document overview

The purpose of this document is to show the planning, design, implementation of GUI scripting for a project, showing the full design loop of at least one area of the GUI. The document is broken up into phases to go through in order.

Project GUI Requirements

### Leading GUI requirements

* Shows player impact of movement system on a recharging resource.
* Be clearly visible, and not obstructed/obstructive of environmental details.
* Only be a concern if players ae haphazard with control system, encourage careful attitude to play.
* Not hinder players willing to test limits of the Player Controller.

### UI Library Usage

* UnityEngine (Used for general functions, comes along with scripting in Unity)
* UnityEngine.UI (Used for specifically adjusting variables relating to UI elements such as a slider value, properties of an image that fills a meter ect.)

### Itemised GUI elements

The primary focus is for the Energy Meter GUI element, is to be easily viewable no matter the level environment, and to have it function be concise.

**Screens & UI elements**

* **Screen 1** - Main screen the game is intended to be played on.
  + **Slider Meter -** A slider GUI element

### Identify GUI & UI Events

#### UI Events

* A timer, that can stop at a finish line
* An input counter, that shows the player their required inputs to finish a level

#### GUI Events

* **Energy Meter-Slider**
  + Remove meter on consumption.
  + Passively regenerate meter.

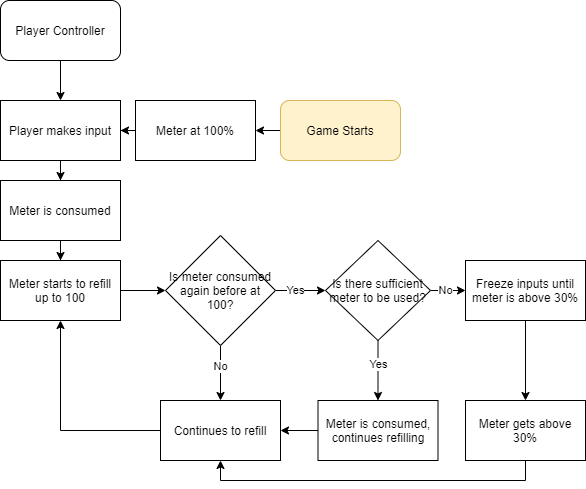
Design GUI

### GUI Design Goals

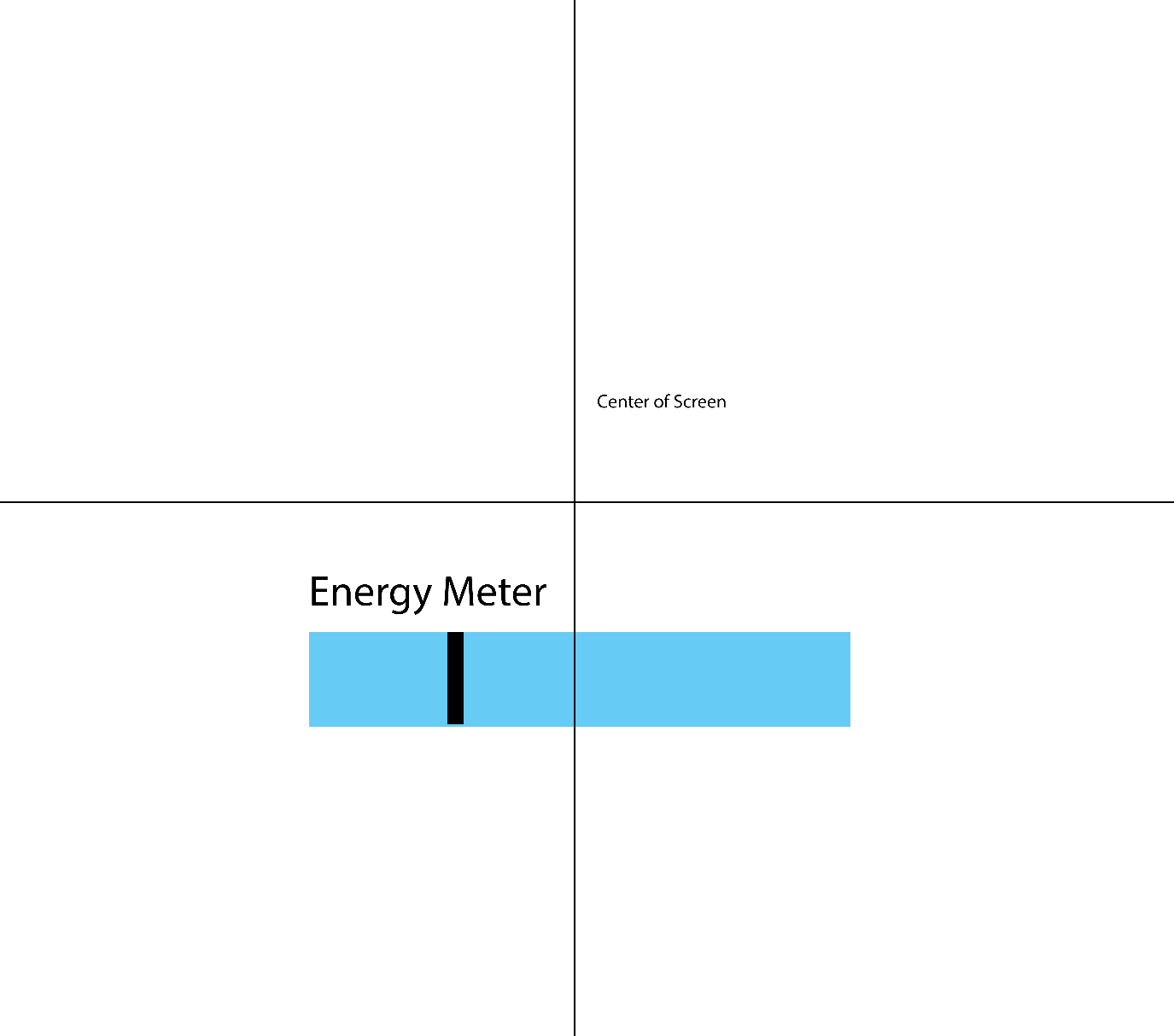
* Be concise with its function with little explanation.
* Not scare the player of consuming meter.
* Have a clean and consistent refill time.

### Identify GUI structure

When the game starts, the meter should be at max. The meter will remain until the player makes an input. The amount of meter consumed is equal to the scale of the input. If the input is consumed below the required amount, it freezes until it gets above 30%. From here it loops back to refilling.



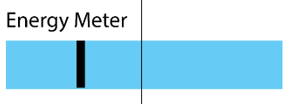
### Initial GUI layout



* The meter sits below the middle of the screen, making it easily visible and in the peripheral view of the player.
* Keeping it out of the corner allows it to be more important, as it’s not relegated to the side.
* It is situated below the centre enough that it will not block view of the player.

### Design GUI Functionality

#### Energy Meter

**Appearance:** 

**Functionality:** The meter will start at full. It will be consumed when the player launches themselves. It will be consumed to an amount equal to the force of the launch. It will always refill. If more is consumed than necessary, it freezes consumption until it has more than 30%.

* Can consume meter depending on player inputs.
* Black bar represents 30% it needs to pass.
* Constantly refills.

Implement & Iterate GUI Design

### Initial layout Implementation

The script worked as intended. It consumed meter depending on the amount of force applied to the player.   
  


### Functionality Review

The entire GUI functions as intended, only requiring some research into coroutines for the recharge mechanic. It was not a very complicated script in retrospect, and I could have had higher expectations. My main issue is how flat the penalty to consuming too much energy was, it only took a few seconds and did not punish the player in any meaningful way.

### External Feedback

* Ui is ‘super clean’.
* The screen is rather barren, more visual details would be nice.
* Try adding a timer and an input counter.

### Planned Response to Review & Feedback

* Attempt implementing a timer without milliseconds.
* Attempt an input counter.

### Iterated Implementation

* Timer works as intended, only starts after 1 input is made. When in single digits, it doesn’t have an extra zero, so some early timers seems off.
* Input tracker works as intended; no bugs are visible so far.