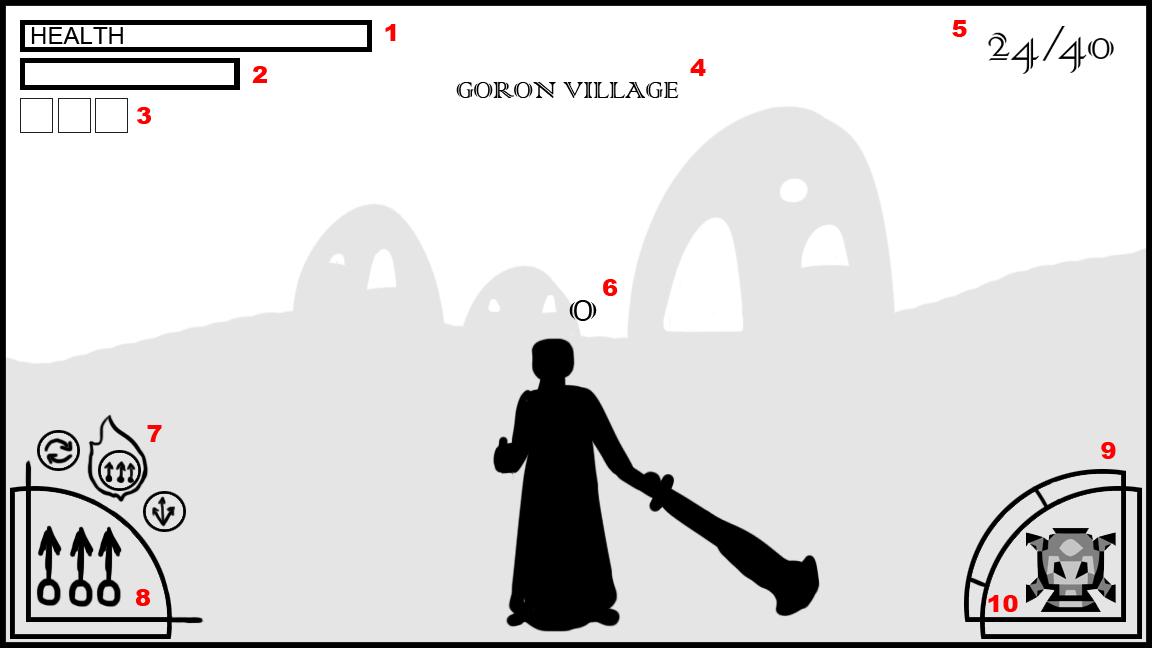
## Player HUD



1. **Player Health:** The flame icons represent the player’s health. Based upon how strong the fire is burning is representative of the player of the player’s current health. Once the flame goes out the player is dead.
2. **Soul Gauge:** For each kill the player gets their soul gauge increases which allows them to summon more troops. Their soul gauge glows when it is full.
3. **Summon Icons:** Icons indicating the units you can summon by spending your Soul Gauge. When the gauge is yellow, the first Light Summon icon is illuminated; when Green, the second Medium Summon icon; when Blue, the third Heavy Summon icon. when the Soul Gauge is glowing, players can summon their Thrall Troops.
4. **Current Map:** Status text informing the player where on the map they are currently at. It pops up for a few seconds giving the player more than just enough time to read it.
5. **Unit Count:** Represents how many allied units out of the initial load-in are still alive on the battlefield.
6. **Targeting Reticle:** The player uses this to help him tell where he is aiming for spell casts, and whether he is targeting the desired unit type. The icon changes based on whether a unit is hovered over, to an oval themed for that unit type. If Ganon is within range for his current range attack and the Reticle is over an enemy it begins glowing with a slight flame effect around it.
7. **Current spells assigned:** Represents the spells currently available to Ganondorf, given the unit type absorbed (will show at half opacity when Ganondorf has an ally unit targeted, but not yet absorbed). Spells which have had their casting-requirement (i.e. a certain number of allies absorbed) met will glow, while those which have not will be dulled. The spell selected will have a flame burning around it, which will jump to the next on the queue when prompted.
8. **Current Spell Selected:** An Icon for the Spell Ganon currently has selected to cast. In the center of the icon a bolded number sits, indicating the **Unit Coun**t cost for casting it.
9. **Relative Troop Percentage:** A dynamic slider showing the relative percentage of different unit types that comprise your current remaining troops. They are color coded, a color also displayed on the **Current Units Absorbed**.
10. **Current Units Absorbed:** The icon in the center of the arc changes to the unit Ganon currently has selected/absorbed. When allies are absorbed, a number will display over the icon to indicate how many were grabbed.

**Pseudo Code**

Void OnGUI()

{

//Use GuiLayout followed by the type of interface object wanted to create multiple

//interface elements with standardized formatting.

//BeginArea lets the program know where to constrain the objects to.

GuiLayout.BeginArea(...);

//Tells Unity to draw the interface horizontally.

GuiLayout.BeginHorizontal(...);

//Let’s Unity know to draw the interface vertically

GuiLayout.BeginVertical(...);

//The Current Spells assigned would be the call here.

GuiLayout.Button(...);

GuiLayout.EndVertical(...);

GuiLayout.EndHorizontal(...);

//All formating calls, Such as BeginArea need a corresponding closing area.

GuiLayout.EndArea();

}

We can use the default Unity type that fits our purposes the best, and override them to get the look we want. This is fairly easy to do. Buttons, labels, toggles, etc each can have a GUISkin assigned to them. In Unity you can create GUISkin and custom looks for each state the GUI elements can be in, such as hover, press, selected. You get the GUIskin to your script, and then use it in the Button or such like so.

public GUISkin CustomSkin;

GUI.skin = CustomSkin

//GUI.Class from here on will use GUI.skin’s custom settings for that type to create their

//look and feel.

GUI.Button(...);

GUI.Toggle(...);

GUI.Label(...);

If you just want to apply a custom look to one piece of the interface use a GUiStyle.

To apply it you first create it and then assign values to it.

GUIStlye customStyle;

//in start or some other function

//Changes color of hover to red.

customStyle.Hover = Color.Red;

//position takes a Rect

GUI.Button(position, text, customStyle);

//The Button will change red when hovered.