Unit 5 Reflective Journal - Javascript Proposal

Based on your personas and scenarios developed for Unit 1, in your learning diary identify ways that JavaScript elements might improve your site. It is important that the code you develop meets the learning outcomes for this unit as well as being justified and useful in the context of the site.

Idea #1

Gladiator Defence

The concept is that a gladiator is attacking you, and you have to defend yourself. You're shown an image of a gladiator attacking you for 2000ms, and you have to click on the appropriate defence or counter move before the time runs out. You have to defeat your enemy or you lose. This is meant to excite Horatio the Hobbyist in Scenario #6 to get him more engaged in Roman culture.

Variables

```
var attackType
var defenceType
var battlePairs =
{“stab”:
        {“block”: “shield”,
        “counter”: “sweeplegs”,
        “image”: “images/stab.png”},
“highswing”:
        {“block”: “sidestep”,
        “counter”: null,
        “image”:”images/highswing.png”}
} etc.
var enemyHealth
var playerHealth
var reactionTimer
var battleOver
```

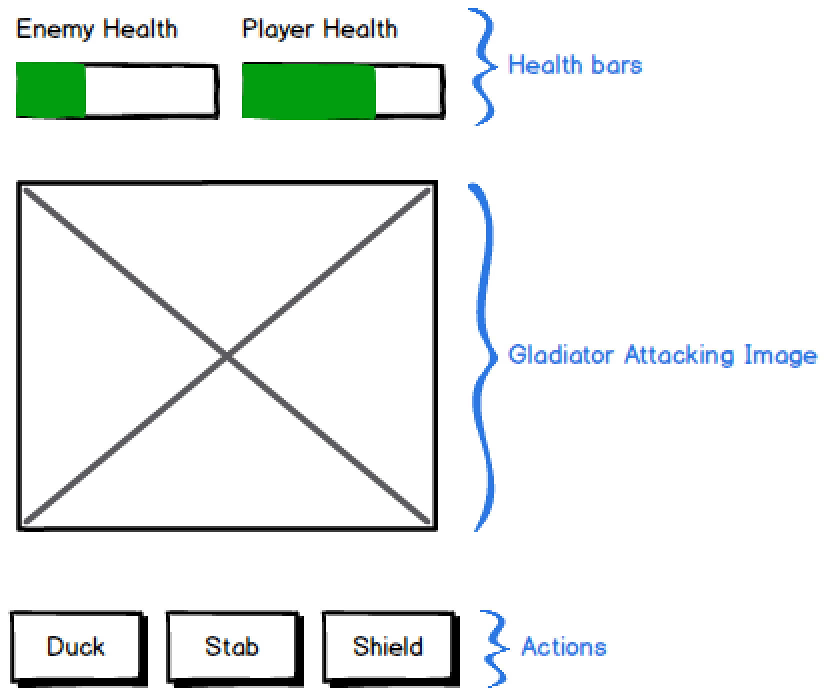
Add #gladiator-img id for the image of the gladiator
Add .action class to buttons
Gladiator health bars done with CSS classes. A div inside another div. A health bar having 33% width .health-1 with background green represents 1 health of max 3.

Battle Function

```
Every second until battleOver is true
        Pick a random key from battlePairs, set to attackType
        Render the image to the id #gladiator-img
        Render 3 defenceType buttons, the block and counter, and another random option.
        switch
```

```
            case: the defenceType is the value of the block inside the attackType object,
no-one loses health
            case: the defenceType is the value of the counter inside the attackType object,
enemy loses -1 health
            case: if the defenceType is neither value of the block or the counter of the
attackType object, player loses -1 health
            default: (the timer runs out before a defence is clicked), then player loses -1
      health
      render health classes for gladiator and player
            1 health = .health-1 = healthbar width: 33%
```

Mockup



Idea #2

Currency converter

The high-level idea is that you are exchanging your money (in CAD) to the Roman currency at the time, Denarii. The converted uses an exchange rate to determine how much you'll have to spend. This provides some classroom content for Prem the Professor in Scenario #5 and engages Horatio the Hobbyist in Scenario #6 by making Roman history relevant to his life. The tool will be on the tools page.
http://student.athabascau.ca/~zacharyle/unit4/tools.html

Variables

```
var exchangeRate
var startAmount
var startCurrency
var endCurrency
var endAmount
var errors = []
```

On #convert-button click

```
    if validation function returns true, run convert currency function
    else loop through errors array and insert errors array strings into an unordered list
#currency-errors
```

Check Validation Function validation(startAmount)

```
switch
case: amount is not a number, return false and add “Please input a number” to errors array
case: amount is < 0, return false and add “Please input a number > 0” to errors array
case: amount has > 2 decimal places, return false and add “Max 2 decimal places” to error array
default: return true
```

Convert Currency Function

```
if startCurrency is CAD
    endAmount = startAmount / exchangeRate
    endCurrency = denarii
else
    endAmount = amount * exchangeRate
    endCurrency = CAD
Insert into id #end-amount the endAmount
Insert into id #end-currency the endCurrency
```

Idea #3

Tabs for Categories Page

The tabs are for presenting the content in a neater way. This is useful for consuming large amounts of content as you are more focused on the content that is revealed. The user can thus learn more. It would help the persona #2 Silvia the Student in Scenario #4 be able to get a better general grasp of the information because she’d be more likely to read it that if the page were really long.
http://student.athabascau.ca/~zacharyle/unit4/categories.html

Flow

```
Give tab buttons a class of .tab-button
Give tab buttons ids relating to their names #tab-history
Give tab contents a class of .tab-content
Give tab contents ids relating to their names #content-history
On .tab-button click, run an anonymous function:
    get id of element var tabName = #tab-history
    split strings tab-history by - into an array var tabNameSplit = [“tab”, “history”]
    concatenate string “#content-” + tabNameSpit[1]
    display:none all .tab-content by adding class .hidden (CSS .hidden {display:none})
    display:block specific tab-content with id #content-history by adding class .visible (CSS .visible {display:block})
    add class .active to #tab-history for aesthetics
```