

Unit 5 Program Design

The next step is to identify the data, objects, functions, methods, and properties you will need to assemble in your program and, once you have gathered these together, to design the logical process model.

Idea #1 (only doing first idea)

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 3000ms, 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 =
[
    {
        "attack" : "stab",
        "block" : "shield",
        "counterWorks" : true,
        "counter" : "stab",
        "mistake" : "swing low",
        "image" : "images/stab.png"
    },
    {
        "attack" : "highswing",
        "block" : "shield",
        "counterWorks" : false,
        "counter" : "sidestep",
        "mistake" : "duck",
        "image" : "images/highswing.png"
    }
]
var gameVars = {
    "enemyHealth" : 100,
    "playerHealth" : 100,
    "enemyHealthBar" : enemyHealthBar,
    "playerHealthBar" : playerHealthBar,
    "battleOver" : false,
```

```

        "responseButtons" : [
            buttonOne,
            buttonTwo,
            buttonThree,
        ],
    };

```

Add #gladiator-img id for the image of the gladiator

Add .action class to buttons

Battle Function

Every 3 seconds until battleOver is true

Pick a random key from battlePairs

Render the attacking image to the id #gladiator-img

Render 3 buttons: block, counter, and dodge.

Decrease Health

Renders the health CSS bars, uses the playerHealth and enemyHealth to set the width

Decreases the gameVars health bars for either the player or the enemy

50 health = 50% width of the health bar

Determine Result

Takes in the player move and checks it against the battle pair

switch

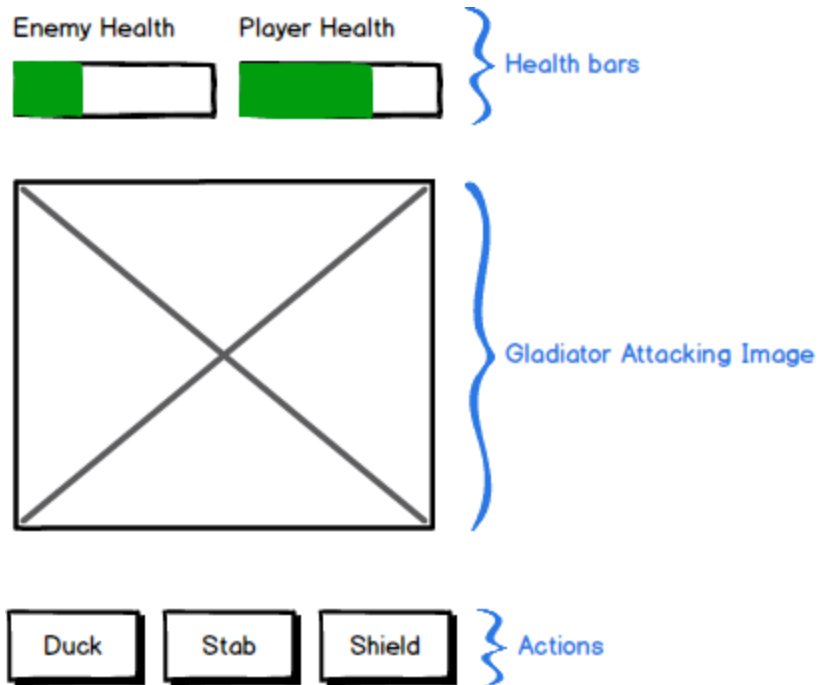
case: the defenceType is the value of the block inside the attackType object, no-one loses health

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

Mockup



Flow

