Assignment 8 – Implementing a Web Portfolio/Tool Project

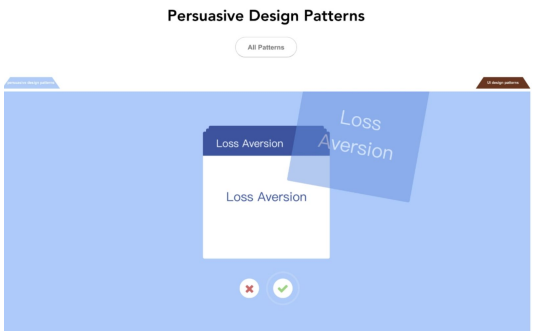
1. Introduction

In assignment 8, I finished a web/ mobile UI Patterns Learning tool. This tool is trying to help users to browse the UI pattern concepts, learning while playing. Targeted users are people who are interested in UI/UX design. Users can identify the UI patterns they are not familiar with, this tool will remember the pattern users are not familiar with and the user can look into the details of each pattern he is not familiar with then redo the test. With reviewing the concept, again and again, the users will have a better understanding of UI patterns. The content is based on http://ui-patterns.com/patterns, and I, myself, generate all of the cards, including 187 concepts. The whole design takes advantage of the card. I divided into two parts, the first is the testing part, the idea here is to show animation ideas for positive (accept) or negative (reject) feedback on a generic card element. I came up with this idea with the inspiration from dating apps like Tinder. This is helping the learner to focus on one concept at a time, they need to figure out what the pattern is or click on the cross to show they don't know this pattern. The other part is the concepts browsing part. I have different versions for people using different platforms. Since the hover function is effective for people using the website, I came up with the concept card wall. While the mobile user needs the function of scrolling, I make it simple and intuitive to browse all of the concepts.

2. Interaction with this UI learning tool

   Take a look at all of the interactions in this tool:
   https://www.youtube.com/watch?v=o7P1IYBEwVw&feature=youtu.be

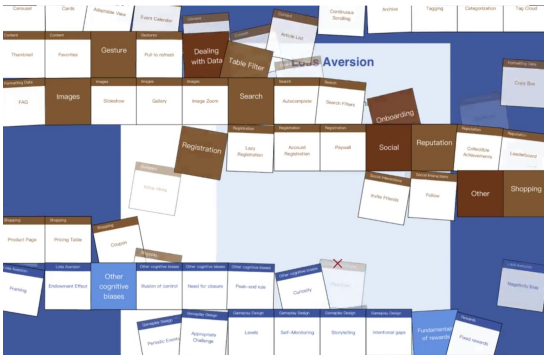
   1. Interactive learning

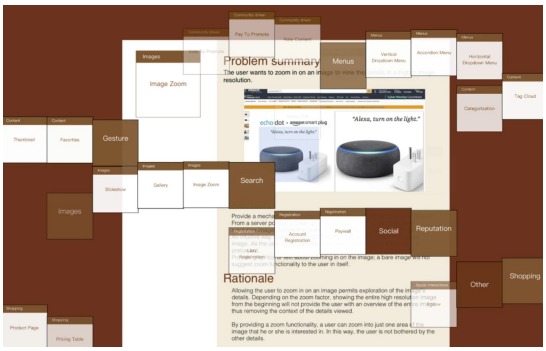


Testing:
On the testing page, users can only see the concept name of each UI pattern, they need to recall the detail of this UI pattern and identify if they are familiar with a pattern or not. When they click on cross, the pattern and its details info will be recorded and put into local storage, they will have access to the pattern they don't know and look into the detail of the pattern they don't know.

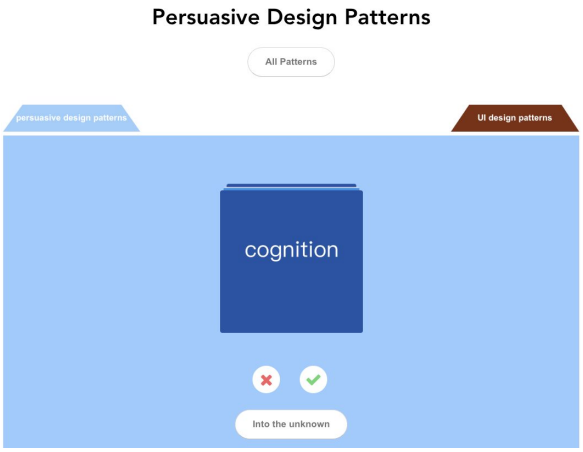
2. UI patterns wall
   This is a staggering effect that will hide the small cards with only the concept name on it and show the content of the specific UI pattern underneath. When you click on the small card, it will "fall down"(pic on the left). When closing the view, all the small cards will show again by moving them upwards(pic on the right).

   Another feature is the follow-through, The whole design is not a classic scrolling with a mouse, the web page itself will follow your mouse and it will identify where your mouse is hovering. This is helping to give users a feeling of intelligent and more interactive. You can close the detail view with only click the page you are in, and it will be back to the original UI pattern wall.



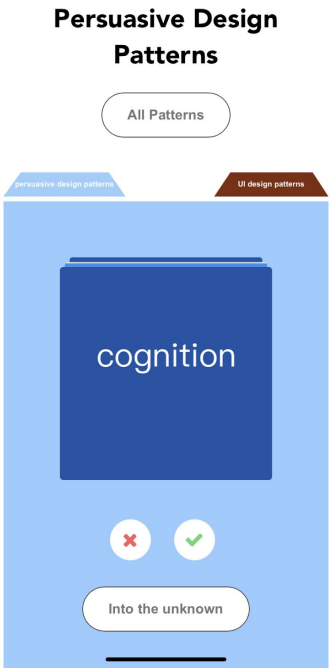


3. Choosing a folder of patterns

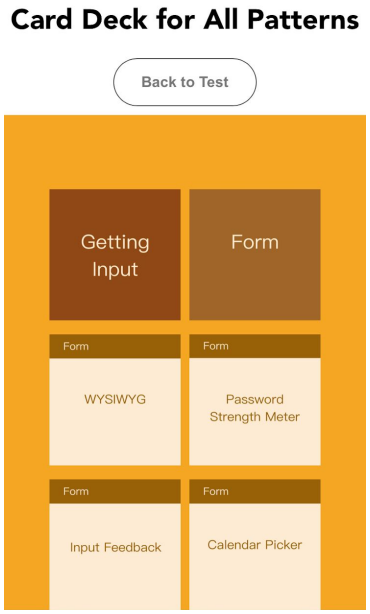


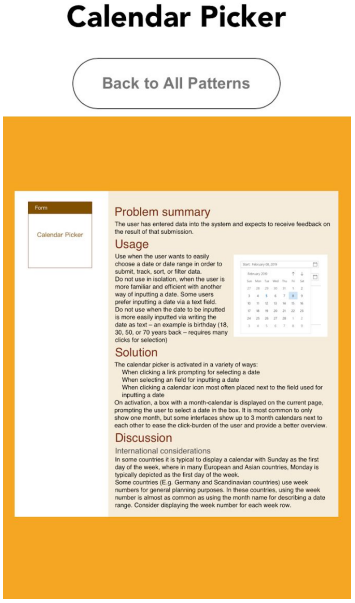
Another interactive design here is the folder. Since the UI patterns have been divided into basic UI patterns and persuasive design patterns. I imagined them as different cards in different folders, so, when you click on the different tab, you will be redirected to another folder of cards.

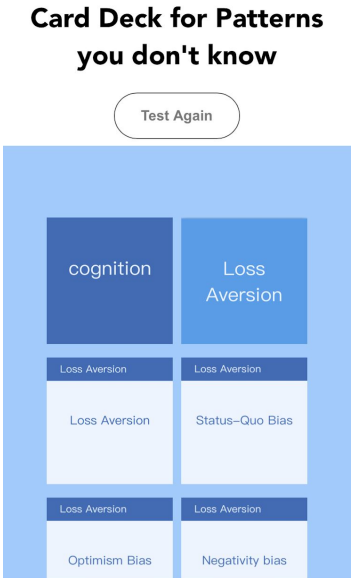
4. Responsive Design



The responsiveness in this UI pattern tool is not only responsive to the screen size, but it is also responsive functionally. One of the features of website browsing is the mouse hovering, while people will not have the hovering effect on mobile. People will use scrolling a lot. Therefore, the solution is redirecting the user to another page when they are using mobile. Pics on the left is what the user will see on their mobile devices.

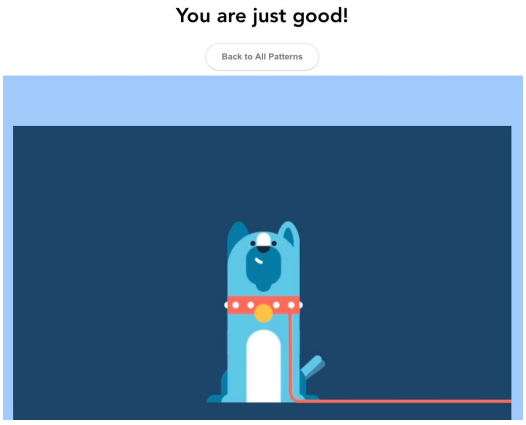


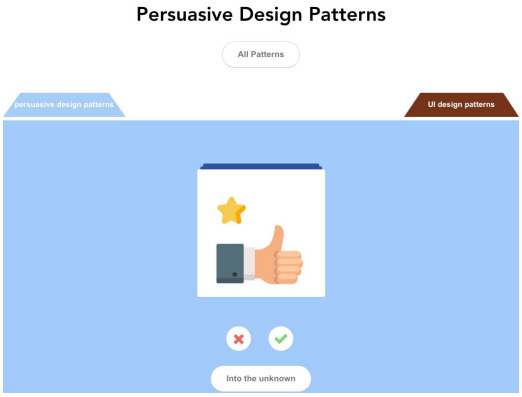




5. Game Mechanics

   The tool is more like gamification of the previous UI pattern tool web page. The most typical feature of the website is the level design. Since there are a lot of patterns, and users will be bored when they are learning. I was trying to encourage them by adding levels to show their completion.





3. External tools

Javascript library:

Dynamics.js - the animation of card browsing by clicking on accept or reject.

```
var krisna = new Stack(document.getElementById('stack_Persuasivedesignpatterns'), {
stackItemsAnimation : {
  duration: 800,
  type: dynamics.spring
}
});
```

```
        delay: anime.stagger(350, {grid: this.gridDef, from: this.pos})
    });
    this.animation.play();
    this.animation.finished.then(() => {
        // Pointer events class
        this.DOM.el.classList.add('grid-wrap--hidden');
        this.isAnimating = false;
    });
```

Anime.js - the UI pattern wall with “falling down” and refill of the wall.

Classie.js - the radical action when I am clicking on the accept/ reject button

```
radialAction.style.left = Number(offset.left - boxOffset.left) + 'px';
radialAction.style.top = Number(offset.top - boxOffset.top) + 'px';

classie.add(radialAction, classie.has(bttn, 'button--reject') ? 'material-circle--reject' : 'material-c
classie.add(radialAction, 'material-circle--active');
onEndAnimation(radialAction, function() {
  classie.remove(radialAction, classie.has(bttn, 'button--reject') ? 'material-circle--reject' : 'mater
  classie.remove(radialAction, 'material-circle--active');
});
```

Animation

Gif, hover, and CSS animation.

```
{
    const MathUtils = {
        lineEq: (y2, y1, x2, x1, currentVal) => {
            // y = mx + b
            var m = (y2 - y1) / (x2 - x1), b = y1 - m * x1;
            return m * currentVal + b;
        },
        lerp: (a, b, n) => (1 - n) * a + n * b
    };

    // Window size
    var newheight = window.innerHeight*1.2;
    var calcWinsize = () => winsize = {width: window.innerWidth , height: newheight};
    calcWinsize();
    window.addEventListener('resize', calcWinsize);

    const getMousePos = (ev) => {
        let posx = 0;
        let posy = 0;
        if (!ev) ev = window.event;
        if (ev.pageX || ev.pageY) {
            posx = ev.pageX;
            posy = ev.pageY;
        }
        else if (ev.clientX || ev.clientY)  {
            posx = ev.clientX + body.scrollLeft + docEl.scrollLeft;
            posy = ev.clientY + body.scrollTop + docEl.scrollTop;
        }
        return {x: posx, y: posy};
    }
    // Track the mouse position
    let mousePos = {x: winsize.width/2, y: winsize.height/2};
    window.addEventListener('mousemove', ev => mousePos = getMousePos(ev));

    // Custom mouse cursor.
    class CursorFx {
        constructor(el) {
            this.DOM = {el: el};
            this.DOM.toggle = this.DOM.el.querySelector('.cursor__inner--circle');

            this.DOM.title = this.DOM.el.querySelector('.cursor__inner--text');
            this.bounds = {
                toggle: this.DOM.toggle.getBoundingClientRect(),
                title: this.DOM.title.getBoundingClientRect()
            };
            this.lastMousePos = {
                toggle: {x: mousePos.x - this.bounds.toggle.width/2, y: mousePos.y - this.bounds.toggle.height/2},
                title: {x: mousePos.x - this.bounds.title.width/2, y: mousePos.y - this.bounds.title.height/2}
            };
```

```
@-webkit-keyframes krisnaReject {
  to {
    -webkit-transform: translate3d(-25vw,0,0) translate3d(-60%,0,0) rotate3d(0,0,1,-5deg);
    transform: translate3d(-25vw,0,0) translate3d(-60%,0,0) rotate3d(0,0,1,-5deg);
  }
}

@keyframes krisnaReject {
  to {
    -webkit-transform: translate3d(-25vw,0,0) translate3d(-60%,0,0) rotate3d(0,0,1,-5deg);
    transform: translate3d(-25vw,0,0) translate3d(-60%,0,0) rotate3d(0,0,1,-5deg);
  }
}

.stack--krisna .stack__item--accept {
  -webkit-animation: krisnaAccept 0.5s forwards;
  animation: krisnaAccept 0.5s forwards;
}

@-webkit-keyframes krisnaAccept {
  to {
    -webkit-transform: translate3d(25vw,0,0) translate3d(60%,0,0) rotate3d(0,0,1,5deg);
    transform: translate3d(25vw,0,0) translate3d(60%,0,0) rotate3d(0,0,1,5deg);
  }
}

@keyframes krisnaAccept {
  to {
    -webkit-transform: translate3d(25vw,0,0) translate3d(60%,0,0) rotate3d(0,0,1,5deg);
    transform: translate3d(25vw,0,0) translate3d(60%,0,0) rotate3d(0,0,1,5deg);
  }
}

/* ...when content has 100% viewport width */
@media screen and (max-width: 60em) {
  @-webkit-keyframes krisnaReject {
    to {
      -webkit-transform: translate3d(-50vw,0,0) translate3d(-60%,0,0) rotate3d(0,0,1,-5deg);
      transform: translate3d(-50vw,0,0) translate3d(-60%,0,0) rotate3d(0,0,1,-5deg);
    }
  }

  @keyframes krisnaReject {
    to {
      -webkit-transform: translate3d(-50vw,0,0) translate3d(-60%,0,0) rotate3d(0,0,1,-5deg);
      transform: translate3d(-50vw,0,0) translate3d(-60%,0,0) rotate3d(0,0,1,-5deg);
    }
  }
```

Follow through - calculating the mouse position and let the webpage follow the mouse

I also use a little bit of the jquery in the transition

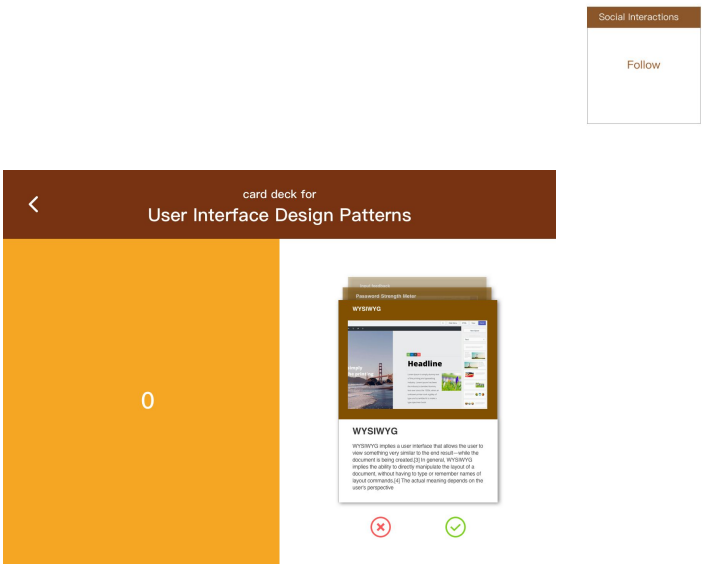
```
function init() {

  $pages.each( function() {
    var $page = $( this );
    $page.data( 'originalClassList', $page.attr( 'class' ) );
  } );

  $pages.eq( current ).addClass( 'pt-page-current' );

  $( '#dl-menu' ).dlmenu( {
    animationClasses : { in : 'dl-animate-in-2', out : 'dl-animate-out-2' },
    onLinkClick : function( el, ev ) {
      ev.preventDefault();
      nextPage( el.data( 'animation' ) );
    }
  } );

  $iterate.on( 'click', function() {
    if( isAnimating ) {
      return false;
    }
    if( animcursor > 67 ) {
      animcursor = 1;
    }
    nextPage( animcursor );
    ++animcursor;
  } );

}
```

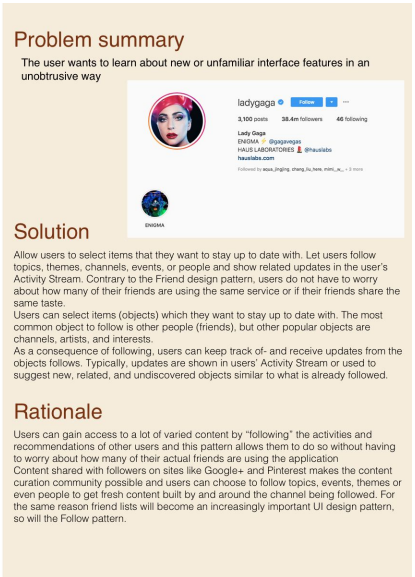
4. Iteration

1. One function I added into this version is the “into the unknown”, this function is letting users have the access to see what they don’t know when they are doing the test when they have not finished the test. The previous version allowed the users to see what they do not know when they finish all of the patters. However, there are too many patterns. This version is for people have little knowledge of UX/ UI design. The users who are not able to go through all the concepts, they can start from some patterns and gradually finish all of them.

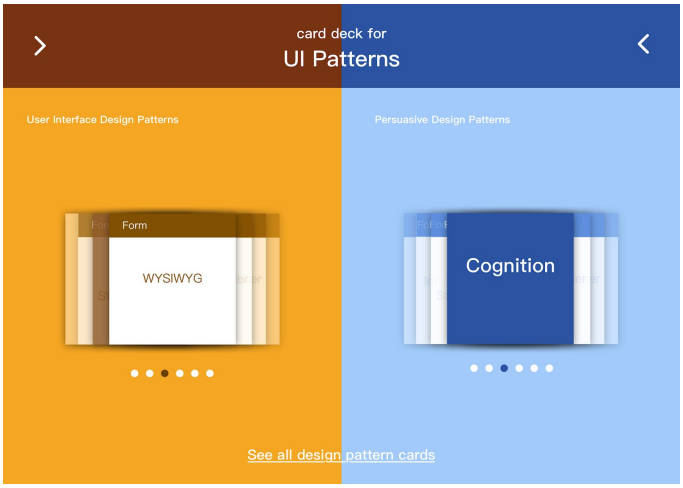
2. Only testing with small cards. Big cards with detail info are for browsing and reading.

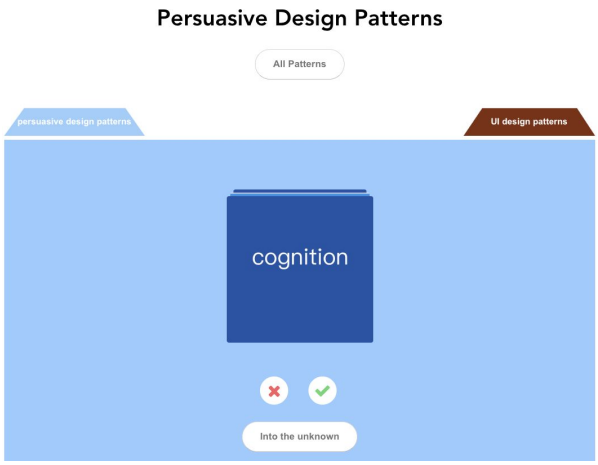
   The big card has also been redesigned to cover more info. The part to show the patterns users don’s know has also been removed which will discourage the users. Only achievements will be shown off.
   (pic on the left is the previous, pic on the right is the latest version)





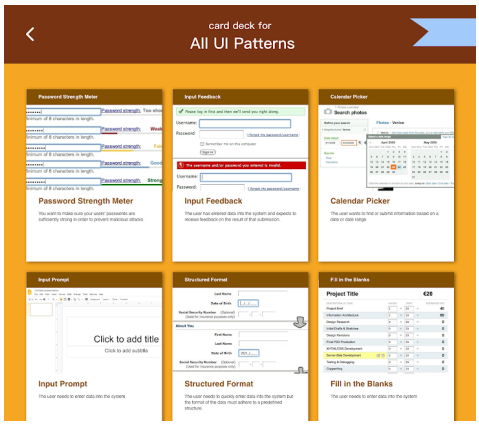
3. Only testing with small cards. Big cards with detail info are for browsing and reading. (pic on the left is the previous, pic on the right is the latest version)

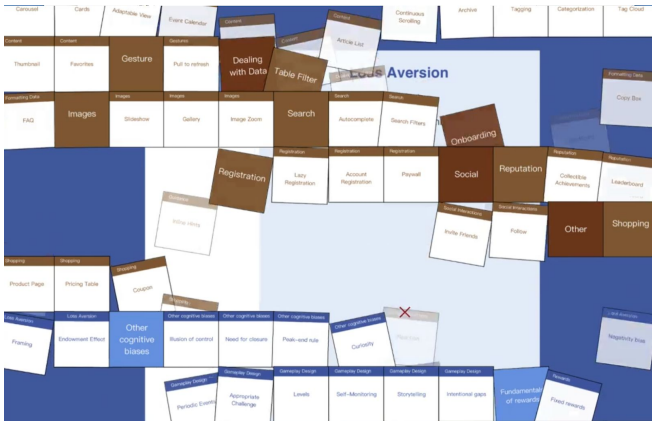




My roommate playtests this website mockup and feels a little bit confused with the previous version which shows two card decks on the same page. She feels distractive. I finally change to the version with 2 tabs(folders)

4. UI pattern wall, it will be a more interactive way to browse all of the patterns (pic on the left is the previous, pic on the right is the latest version)





5. Challenges

The first challenge is having an idea of the tool I want to build, which is both usable, useful and desirable. I google a lot of other creative web designs that are really inspired. Another challenge is how to dig the other js library, open-source scripts and put them into use. The last one is to keep modifying and adjusting to make it nice to use on my own website.

Links:
You can go to my Github repo page through this link:
https://github.com/RanCuiRebecca/RanCuiRebecca.github.io/tree/master/uipatterntool

If you want to see the assign8 website directly, please click this link:
https://rancuirebecca.github.io/uipatterntool/index.html