CSC309 Alpha Release

EasySlides is a slideshow library, which takes images, types of transitions and other configurations as inputs and create a slideshow. The purpose of the library is to provide an abstraction to build a slideshow that responds to various user interactions and has different types of transitions. The originality of the project comes from enabling various transition effects and user interaction. The existing slideshow libraries do not support many transition types and user interactions. For example, swiper and flickity only support fade transition type, and they do not allow users to add paragraphs of descriptions on the current slide. These features are useful for building web applications that serves different purposes, including recipe sharing and presentation creation. In a recipe sharing app, developers can use this library to display top recipes and users can hover over the food images to see the required ingredients and decide whether they have all the ingredients to make it. In a presentation creation app, developers can make use of different transition effects provided by the library to showcase their products and users can have a better idea of what they will be able to get out of it.

There are 11 features implemented for the Alpha Release. The first feature is adding dots to indicate the current slide the user is on. Users can click on the dots to switch slide. The second feature is enabling slide switching through mouse click and/or keyboard press. The third feature is adding previous and next arrows to switch slides. The fourth feature is enabling different transition effects, which are fade, shift, and zoom. The fifth feature is enabling “autoplay” mode and allowing users to customize the time interval for slide switching. The fifth feature is enabling users to add descriptions to each slide. The descriptions will show up when users hover over the slide and the animation will stop if the autoplay mode is enabled. The sixth feature is enabling thumbnails. The eighth feature is allowing users to create vertical slides. The ninth feature is enabling the use of 3D spin slides. Developers can use this view to showcase demonstrate all the slides at once. It can be beneficial for a presentation creation webapp where the developers want to showcase their products. The tenth feature is allowing users to use a thumbnail only view for the slides thus saving developers more space to put other contents to the page. When users hover over the images on the thumbnail, users will be able to see a larger view of the slide. The eleventh feature is a thumbnail split view where users can specify two slides to view and put them side by side to compare the two slides. Developers can also enable the on-hover descriptions such that users can compare the text on the slides as well. Users can also deselect a slide by clicking on the slide in the thumbnail.

Here is an example webpage: https://protected-shore-59731.herokuapp.com/examples.html. This web page will provide a few examples of features that I implemented. The first example is a slide show with dots, previous and next arrows. Users can left click and use left and right key to switch slides. Example two, three, and four showcase three transition effects that can be applied to the slides. They are fade, shift, and zoom. The fifth example is a slideshow with autoplay mode enabled. The sixth example is a slideshow with descriptions on each slide. When users hover over the slide, the animation will stop, and users can see the descriptions associated with the slide. The sixth example is a slideshow with thumbnails. Users can click on the image in thumbnails to switch slide. The seventh example is a thumbnail only view. The eighth example is a thumbnail split view, where users can specify the two images and view them side by side. The ninth example is a vertical slideshow. The last example is a 3D view of the slides.

There is one JS object, “EasySlides”, associated with each slideshow instance. The object has 7 properties: carousel, options, active_idx, timer, carouselItems, splitview, second_active. The carousel property is the reference to the slideshow object in the DOM. Adding and manipulating the slideshow object involves changing the slides pointed to by the reference. The options property contains a list of configurations which are dots, arrows, thumbnail, width, height, onHoverDescription, onClick, keyboardSwitch, transition, vertical, thumbnailonly, thumnailsplit, and spin. The active_idx property indicates the current slide that is shown. The timer is used to track the timer in autoplay mode. It is used to stop the timer when the users hover over images to see descriptions. CarouselItems is a copy of all the slides in the slideshow. It is used when the thumbnail only view is enabled. The library stores a copy of the slides such that they can be restored later when user wants to see a larger view of the slide. Splitview and second_active properties are used when the split view is enabled. It is used to indicate whether the slideshow has two active slides and what the second active index is.

Here is an example of an instance of the EasySlides object:

```
{
 carousel: “<div class=“easyslides”> … </div>”, // reference to the slideshow object in the DOM
 options: {
          speed: 2000,
          dots: true,
          arrows: true,
          width: 800, // in px
          height: 600, // in px
          onHoverDescription: true,
          onClick: true,
          keyboardSwitch: true,
          transition: “”,
          thumbnail: true,
          thumbnailonly: false,
          thumbnailsplit: false,
          vertical: true,
          spin: false,
 }
 active_idx: 0 // the current slide that is displayed
 timer: timerIdx, // this is used for autoplay mode
 splitview: false
 second_active = -1
 carouselItems = []
}
```

Here is an example to create an EasySlide object:

```
const slides = document.querySelector(‘.easyslides’);
const options = {
        dots: true,
        arrows: true,
        thumbnail: true,
}
const easySlidesInstance = new EasySlides(slides, options)
```

Here is a list of API functions developers can access. All the methods are called on an EasySlide object:

getOptions

Arguments: none

Returns the options set to the EasySlide object

----

getCurrentSlideIdx

Arguments: none

Returns the current slide index

----

nextSlide

Arguments: none

Navigates to the next slide

----

prevSlide

Arguments: none

Navigates to the previous slide

----

gotoSlide

Arguments: int: slide number

Navigates to a slide by index

----

getEasySlide

Arguments: none

Get the EasySlide Object

For the final submission, I will be brainstorming and implementing more features that can be beneficial for the use cases described in the first paragraph. The features could potentially include enabling more transition effects, making different types 3D slideshows, making slideshow editable in real time, allowing users to drag and drop a slide, and making thumbnail split view compatible with dots and arrows. The challenges come from making the slideshow library original since there are many existing slideshow libraries online. I will also be documenting the library along the way and creating a unique style for my library.