



DECO 7180
Design Computing Studio 1
Interactive Technology

Final Delivery and Report

TEAM: PROTECTOR

Jian WU
Yawei YANG
Hengshuo LIU
Liqian YOU
Wanli JIANG

Contents

1. Introduction	3	
1.1 Detailed description of the final project		3
1.2 A quick guide to the website	7	
2. Implementation	14	
2.1 Idea generation	14	
2.2 Research	14	
a. Persona	15	
b. Storyboard	18	
c. Scenario	20	
2.3 Prototype	21	
a. Site map	21	
b. Mock-up	22	
c. Low-fidelity prototype		23
d. High-fidelity prototype		24
2.4 Website Development		28
a. Visualization	28	
b. Front-end	33	
c. Emergency map	36	
d. Game	41	
e. Chat robot	58	
2.5 Discuss of process	63	
a. Aesthetic issue	63	
b. Game development issue	64	
c. Database issue	66	
d. Chat robot development issue		67
4. Reflection of success	70	
5. Reference	74	

1. Introduction

The Animal Protector is an interactive website that user could use to have a basic information and understanding of endangered wild animals, get a basic measure of how to protect them through the interactive function of this website. For example, puzzle game in this website is used to help user get the protection information of endangered wild animals through playing the game. This report will contain three parts as description, discussion and reflection. Each part will have an in-depth analysis that how it was implemented on code and how it performances based on this website.

1.1 Detailed description of the final project

The final project our team is a type of interactive website that collects comprehensive data from digital map and wildlife of QLD by using API, to implement multiple functions such as showing recent news about wild animal protection and where are the animal hospitals, chatting with users to answer some basic questions.

As we mentioned in previous proposal, the general purpose of this project is to play an propagandistic and educational role, aiming at helping users to realize the importance of endangered wild animal protection. Be-

sides, users will have a chance to learn basic information of endangered wild animals in QLD, including their inhabitants, description, distribution and potential threats. And there is a puzzle game called flying animal that is provided for user to have these information with a relax state.

According to the Cox's research(2018), the extinction of wild animal in Australia is unfolding in plain sight as there are over than 1800 animals species, plant, and ecological communicates that have a risk of extinction right now. And protecting wild animals is gradually triggering public's concern. However, although more and more people realize that they need to protect endangered wild animal, the most of them don't really know how to take measures. In addition, most websites that are about animal protection only offer the phone number of related protection organization or animal hospital, this is not really helpful sometimes for users, if they encounter the wild animal under emergency. Because, the information is not visible, they need to check lots of things such as location. Based on this, our website will provides lots of emergency measure such as how to close with these animal and some first aid advices. Therefore, the main purpose for this project is to enhance the responsibility of protecting endangered wild animals and help them to build some basic information.

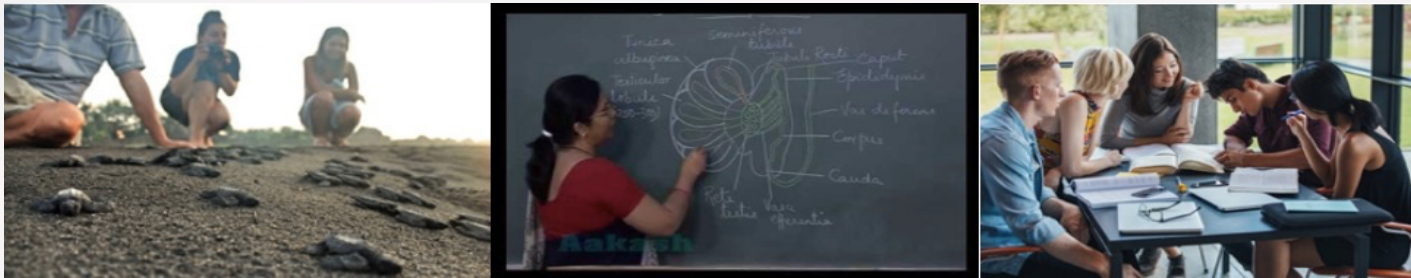


Figure 1 target audience (Google)

Besides, although this website is for public, it is mainly service those target audiences such as zoology teachers, students, wildlife lovers and people who are going to contact with endangered wild animals. And this website is gender-neutral, its target audience have a age range from 14 to 55 years old as people in this range have strong responsibility and ability to help protect endangered wild animals. We will distribute these audience into two part as public that lacks animal common sense and have little knowledge about endangered wild animals and public that have lots of knowledge about wild animals such as wildlife lovers and zoology teacher shown in figure 1. The following will explain the different purpose on the different audience parts.

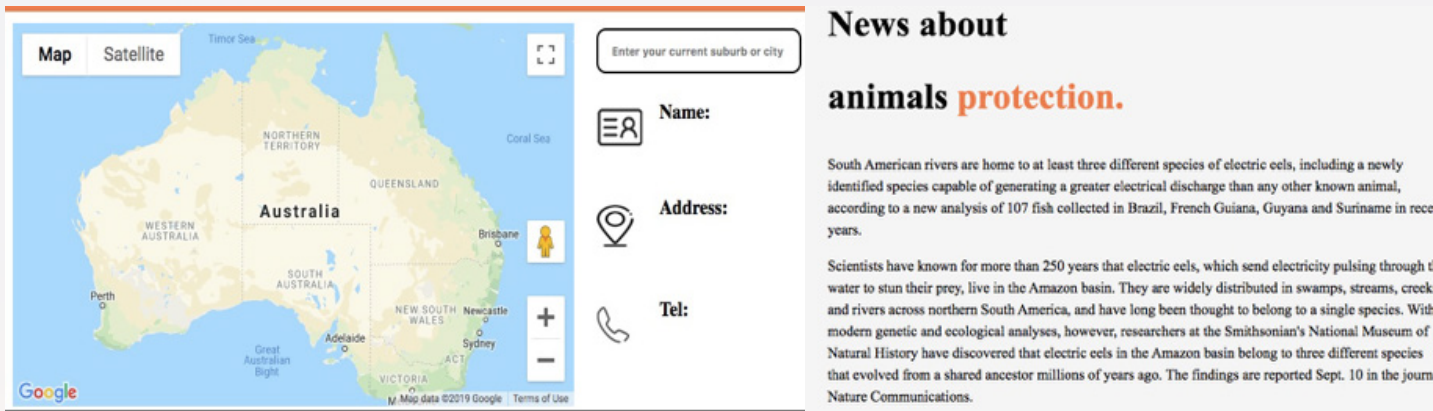


Figure 2 emergency and education part

For the public who are going to learn knowledge about wild animals for their tourism or curiosity, they only want to learn the very explicit and basic information about the wild animals. Thus, education part containing the basic introduction of animals and precautions and game part will attract this part audience. What's more, not only they could play the interesting game, but the game will show lots of protection information for the specific endangered animal. These two parts of website will satisfy this target audience's needs.

For zoology teachers and wildlife lovers that are familiar with wild animals, they need to build long-term contact with wild animals and need some professional information from research or article or information that is helpful for protecting the endangered wild animals. Therefore, Game, emergency, education part of this website shown in figure 2 could attract this part target audience as news from education part will help update their knowledge. Besides, the map from emergency part not only could help them find the nearby animal hospital, but displays the distribution of all endangered wild animals. This will provide the very useful information for them. In addition, Robot chatting part could also help them get some tips such as how to do the first aid for some injured animals.

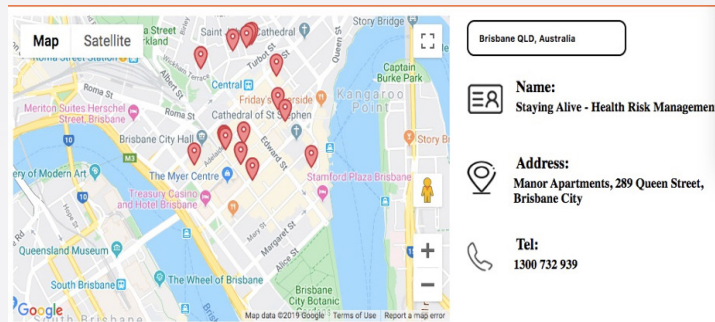


Figure 3 animal hospital search

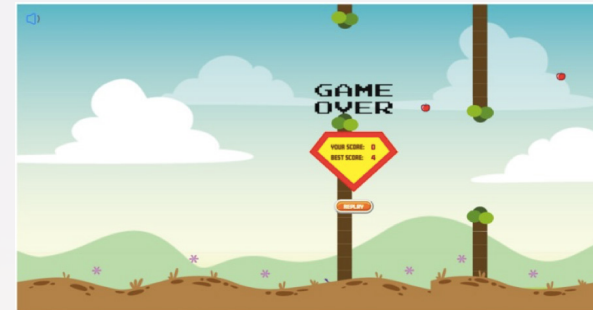
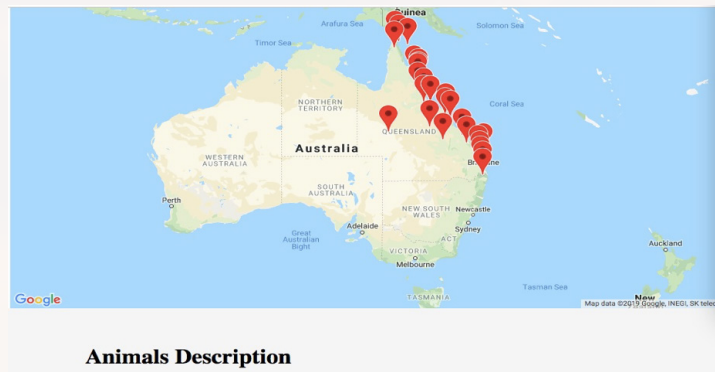


Figure 5 flying animal game part



Animals Description

Figure 4 wildlife distribution

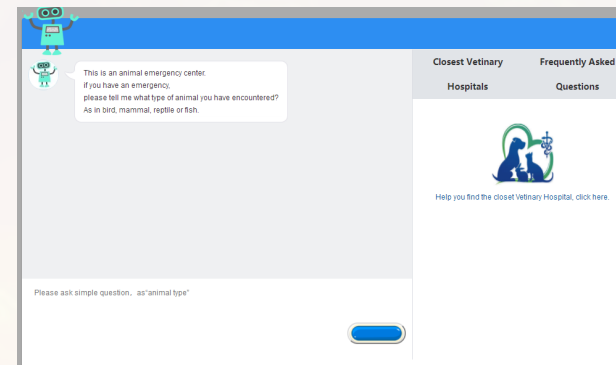


Figure 6 chat robot

For approaching the given data sets, initially, we combine the data of digital map and google maps, use their database to do the search based on animal hospital type. Thus, users could directly input their current position or specific position to find all animal hospital, there will be a window to show the hospital's name, address, telephone number. It works as figure 3. Besides, there is another map in discovery page which use wildlife database in SLQ. We need sort the data of this database to find all wildlife distribution, and provides some basic information introduction. it works as figure 4. Apart from these two interaction, users also could play the game to get the wildlife protection knowledge. User use keyboard to control the game. And when they get enough score, the flying animal will be changed. It works as figure 5. Lastly, figure 6 is the robot chatroom, users could type whatever basic question, the robot will answer it at now. And there will some priority question shown as hint when they are typing.

1.2 A quick guide to the website

If you want to access our website, the first step is to type in the following address in your browser:
<https://deco7180-c01t15.uqcloud.net/>

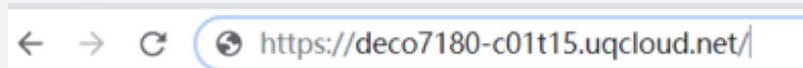


Figure 7 Website link

Tab enter on the keyboard, it will lead you to the homepage of our website.

The topic of the website is about animal protection. We decide to offer services in the following aspects: General information, detailed introduction about endangered species, news about animals, interactive game and a chat robot to assist anyone who is dealing with an animal emergency. The general information section is just included in the homepage, which you have already opened.

The users just need to scroll down to see the briefing of the websites:

1. the news briefing on the top:

News about

animals protection.

South American rivers are home to at least three different species of electric eels, including species capable of generating a greater electrical discharge than any other known animal. A recent analysis of 107 fish collected in Brazil, French Guiana, Guyana and Suriname in recent

Figure 8 News section

2. The research briefing in the middle:



Figure 9 Information section

3. And the contact information at the bottom:

After read through the general information of the homepage. The user would probably, take a look at the detailed information of the websites.

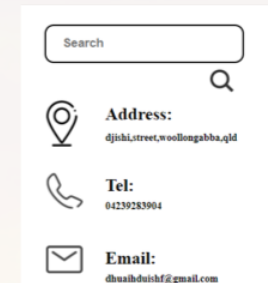


Figure 10 Search

The following is the head bar for users to choose which subject they are most interested in:

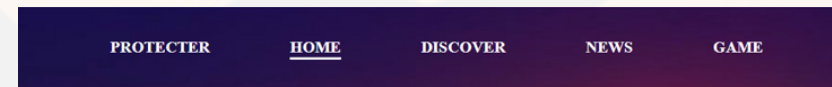


Figure 11 Navigation bar

Using the head bar, the user can access all above mentioned functions except the chat robot.

If the user clicks the discover link, a journey to explore all the magnificent animals in Queensland will begin:

Here, we provide information about the endangered animals in Queensland. This includes their survival situation, their location of habitat, and it also lists about the local policy of recovery for all species specifically.

(http://www.environment.gov.au/cgi-bin/sprat/public/publicspecies.pl?taxon_id=1802)

As it is shown, this page presents information according to the classes of animal, as in: Bird, Reptile, Mammals, Fish, Amphibians. Each class, has a sub list of the endangered species within the class:

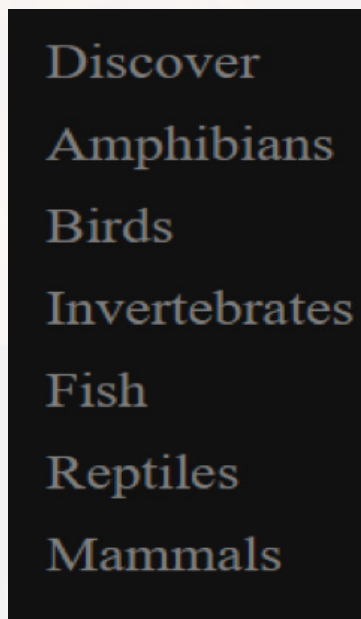


Figure 12
Navigation bar

Some of the species has a link that can lead to Queensland state database, which will present the user more information of the species' recovery plan and current reservation status:

Common name	Scientific name
Bare-rumped sheathtail bat	Saccolaimus saccolaimus nudicluniatus
Black-tailed antechinus	Antechinus arktos
Bridled nailtail wallaby	Onychogalea fraenata
Dusky hopping-mouse	Notomys fuscus
Ghost bat	Macroderma gigas
Greater bilby	Macrotis lagotis
Greater large-eared horseshoe bat	Rhinolophus philippinensis
Julia Creek dunnart	Julia Creek dunnart
Mahogany glider	Petaurus gracilis
Northern bettong	Bettongia tropica
Northern hairy-nosed wombat	Lasiorhinus krefftii
Plains rat	Pseudomys australis
Proserpine rock-wallaby	Petrogale persephone
Semon's leaf-nosed bat	Hipposideros semoni
Silver-headed antechinus	Antechinus argentus
Spectacle flying-fox	Pteropus conspicillatus
Spotted-tailed quoll (northern subspecies)	Dasyurus maculatus gracilis

Figure 13 Animal list

EPBC Legal Status and Documents	
EPBC Act Listing Status	Listed as Vulnerable
Approved Conservation Advice	Threatened Species Scientific Committee (2016). Conservation Advice Saccolaimus saccolaimus nudicluniatus http://www.environment.gov.au/biodiversity/threatened/species/pubs/66889-conservation-advice-07122016.pdf
Listing Advice	Listing assessment information may be available in the approved Conservation Advice
Recovery Plan Decision	Recovery Plan required, included on the Commenced List (1/11/2009).

Figure 14 QLD database

The news link on head bar can lead the user to the local animal news:

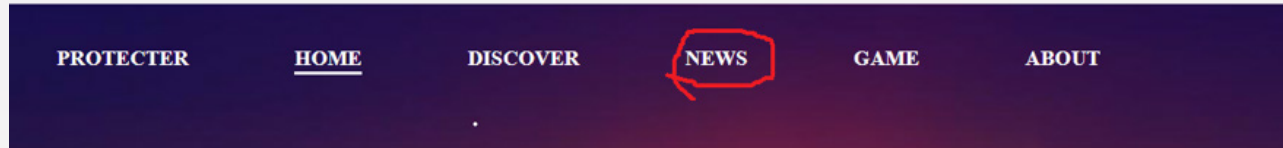


Figure 15 Navigation

To:

News about animals

protection.

Today (October 3), Airbnb is taking action to protect animals by launching a new category of Animal Experiences. The new offering is guided by a premiere animal welfare policy, created in consultation with World Animal Protection.

The online marketplace is now offering more than 1,000 animal-friendly Animal Experiences that consider the well-being of animals across 58 countries.

Airbnb Animal Experiences are activities organised and led by local hosts, including

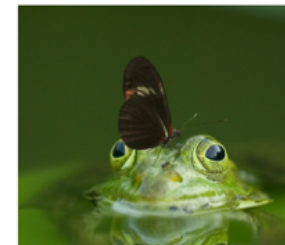


Figure 16 News section

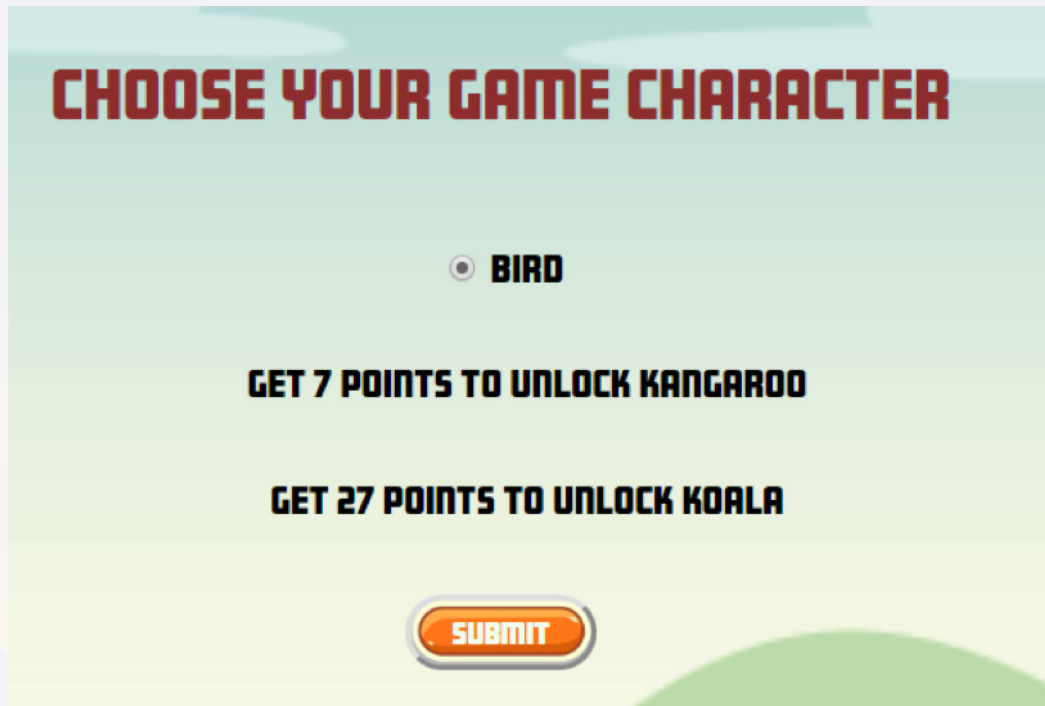


Figure 17 Game section

This is a game similar to flappy bird. Initially, the user will use a bird icon:



. But after accumulating some points the player can unlock the kangaroo icon and koala icon sequentially.

7 points for kangaroo



. 27 points for koala



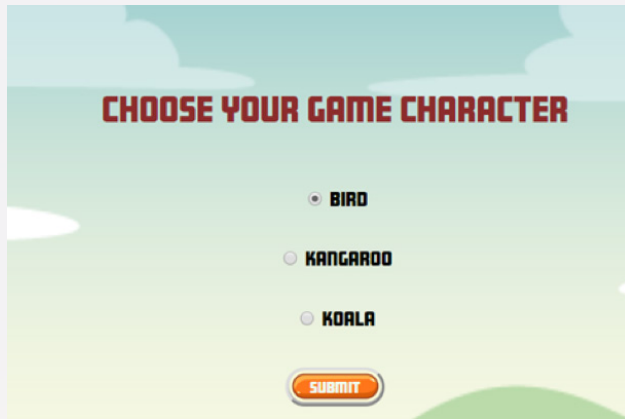


Figure 18 Game Angry Animal

While playing, the user can use space or left click to control the animal. One click will cause the icon to jump up for some distance, but the animal is set to falling constantly, if the player stops clicking. The obstacle will be coming right at the player, and the player needs to control the icon to dodge all the obstacles:

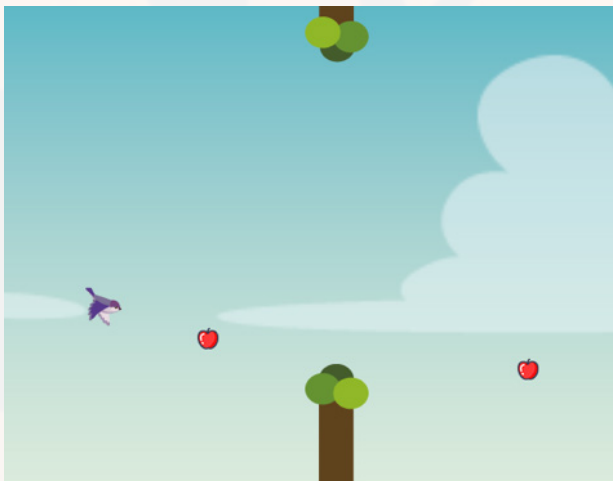


Figure 19 Game Angry Animal

This creates a effects that the animal is the one moving and there is gravity in the game, the animal needs to constantly “flap hard ” to keep float in the air. Every time the animal pass through an obsta- cle, the left side scoreboard will update your new score:

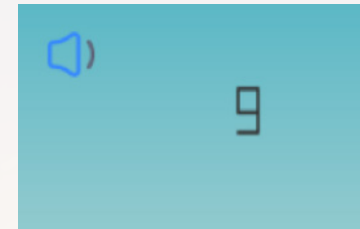
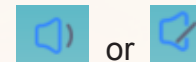


Figure 20 Game Scoreboard

The above shown speaker symbol is also in charge of the game’s sound control. Click once to mute the sound(including background music and game action sound), click it twice to reactivate it.



Another important feature of the game is that be- sides dodging the obstacles, user can choose to collect the apples as a bonus side mission. When- ever the user bumps into an apple, a pop out win- dows will appears, and give the users some fun facts about the wild lives in Queensland, the information, is gathered from the www.data.qld.gov.au and se- lected randomly into the tips:



Figure 21 Game tips

If you are tired of reading all these information. You could play some game:

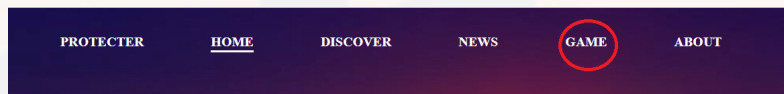


Figure 22 Navigation bar

Just Click it, you will reach the gaming center.

After all the information and functionality on display, Where can we find the chat robot? No worries, in order to makes it more accessible for people who have an emergency. Our dear chat robot is always waiting to help you in the corner:

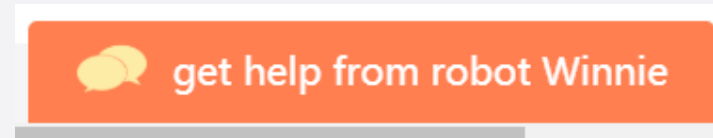


Figure 23 Contact Button

The above shown tag will always stick to the bottom right, just like a loyal robot. Waiting for your request.

If the user clicks the button:

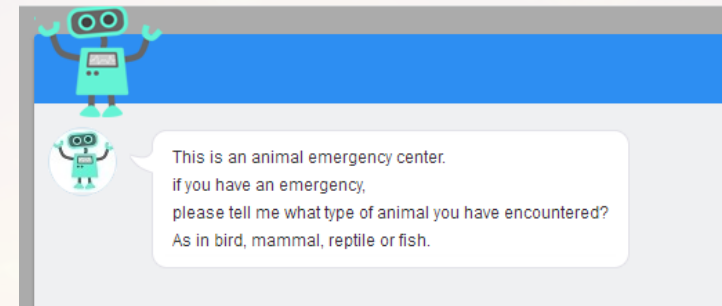


Figure 24 AI robot

You will meet friendly robot Winnie. Winnie is specialized in dealing with emergency situation. He tell you the most suitable hospital according to the species of the wounded animal. Winnie also has some sense of humour, if you tell him certain things, he will choose to joke about it.

Basically, users' input goes into here:

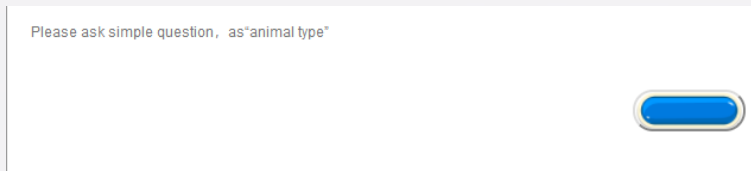


Figure 25 Type-In Area

If you type more than 2 letters, a suggestion will pop out:

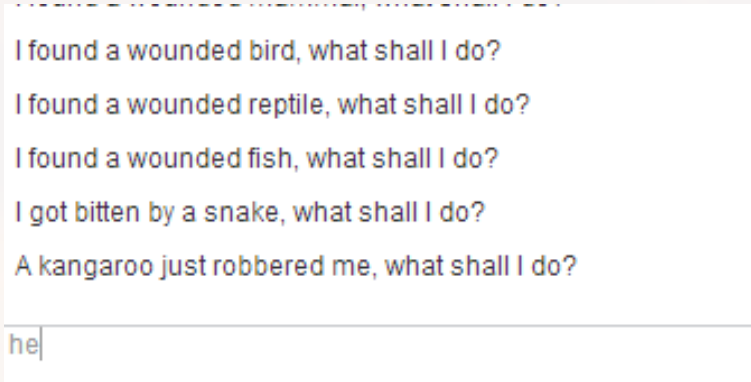


Figure 26 Suggestions

The suggestions actually offers the sum of all the questions Winnie will respond to.

If the user choose to type themselves, Winnie will read between lines to find the key words and offer you a related answers. Only if the key word isn't there Winnie will insist his primary occupation is to deal with animal emergency. A very stubborn robot indeed.

On the right of the chatting area:

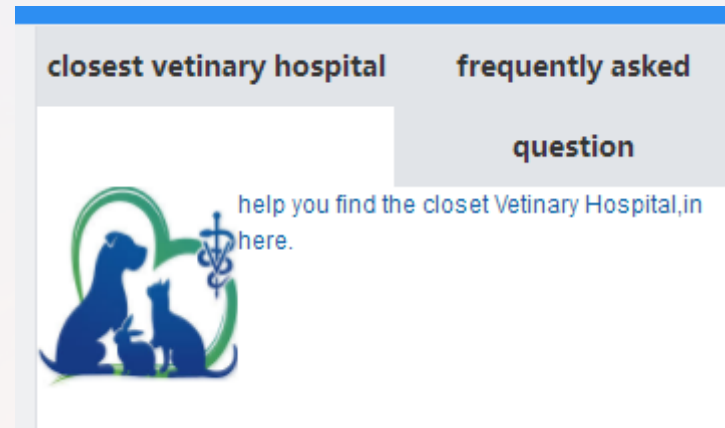


Figure 27 Help column

A help column is there to help anyone who can't find the veterinary event if the address is provided by Winnie. A frequently asked and answer panel is also available.

Finally, the website provide an about function:

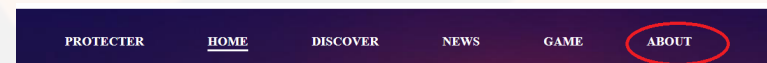


Figure 28 Navigation bar

If you click it, it tells you about us, and how you can find us.

2. Implementation

2.1 Idea generation

At the beginning, considering the ideas from all team members to our new group project, we generated our individual pitch and domain research in the first five weeks. Three members' topic related to the environment and animals. Two of us focus on map and historic tour. So, we tend to further investigate the concept on animals. After conducting the world cafe method on week 6, we discussed to build a data and information website on animal protection, because we found there are many valuable animal datasets on SLQ, but few concepts or websites related to animal protection. Therefore, we focused on developing three main functions: news, game, information and education.

Team meeting being held once a week, is the chance we articulate ideas and communicate with other team members. We also discussed on how to make these ideas into real function and adjusted to be more reasonable and achievable. The idea of a game from one of team member Nook, which provides a better experience to suffer on our website. Users can learn some knowledges related to animals protection by playing games. The player can control an animal in the game scene, by using keyboard and mouse to move and collect apples. Each apple stands for one key information about the animal, when users collect them, they can get rewards. And stones represent notice and alarm (related to animal protection) that will make users lose marks when collide with them.

We also generated a great idea about wildlife rescue. When people meet some animals facing emergency or danger, he or she can use our website to call the veterinary hospital or even the local NGO for assistance. We would like to set a chatting room, which allows users and professionals communicate online, so as to improve efficiency.

2.2 Research

This section presents the investigation of the intended audience and domain research. The concept tools and design methods we applied in the idea generation section are Use-Case Scenario, Mind-mapping, and Storyboard to set target audience, main functions and website visual appearance. As our target audience are government officers, students, QLD visitors, citizens or any animal lovers, from teenagers to adults, we use storyboard and scenario to probe users experience and behaviours.

a. **Persona**

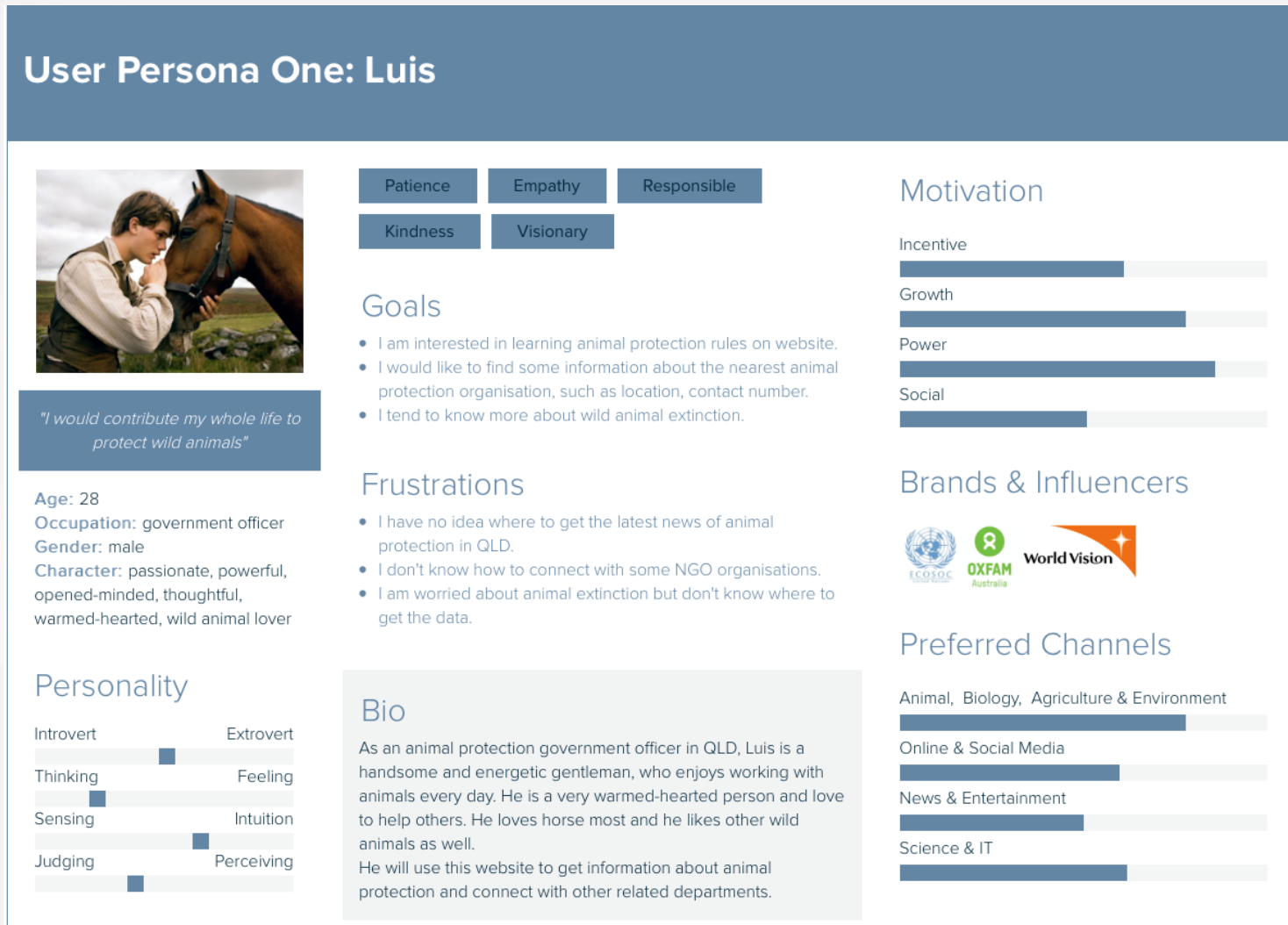


Figure 29 Persona

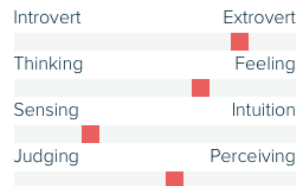
User Persona Two: Sophia



"God creates nature and life. We should protect them."

Age: 22
Occupation: journalist
Gender: female
Character: energetic, warm-hearted, vegetarian, dog lover

Personality



- Optimistic
- Empathy
- Human-centered
- Helpful
- Sensitive

Goals

- I am interested in NGO and animal protection and want to find such a kind of website that I can get information from.
- I would like to find some information about the current news related to animal protection.

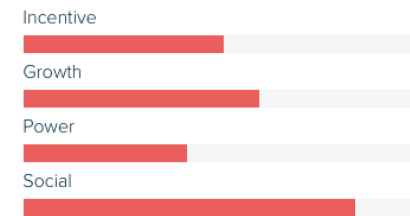
Frustrations

- I don't know how to find the latest news and regulations about animal protection in QLD.
- I don't know how to help wild animals when they face emergency.
- I have no idea about how to get data on animal extinction.

Bio

As a journalist in QLD, Sophia is an optimistic girl with great energy. She is also an excellent photographer. She started to focus on animal protection when she raised a puppy in her childhood. Her favourite animal is dog and she is the mother of five dogs. She is going to use this website to learn more information about animal protection, especially wild animals.

Motivation



Brands & Influencers



Preferred Channels

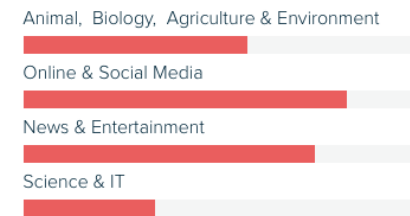


Figure 30 Persona

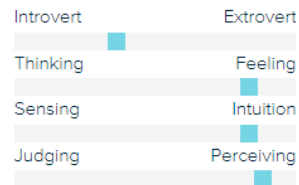
User Persona Lily



"Life is beautiful."

Age: 20
 Work : Student
 Gender: female
 Character: sensitive, emotional, travel enthusiasts, photographer

Personality



- Scrupulous
- Independent
- Creative
- Sensitive

Goals

- I love wildlife and nature, I want to know there can take many photos of wildlife.
- I want to learn biology through games.
- I want to read current news related to animal protection.

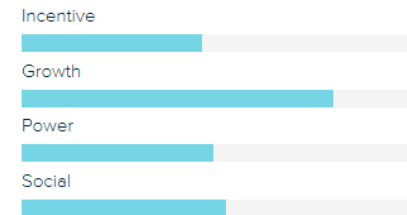
Frustrations

- I don't know where to find the places for taking photos of wild animals.
- I don't know how to do when encountering a wild animal.
- I don't know an interesting way to learn animal protection.

Bio

As a university student major in biology, Lily is a kind and energetic girl, love to take photos and be a volunteer of wildlife. She wanted to be a photographer or journalist for wild animal magazines. She will use this website to learn how to contact with animals. She would like to play games on this website as well to learn knowledge when relaxed.

Motivation



Brands & Influencers



Preferred Channels

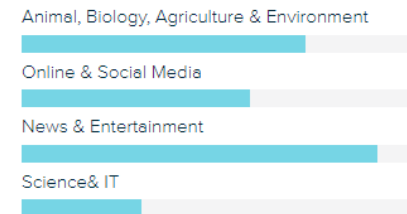


Figure 31 Persona

b. Storyboard



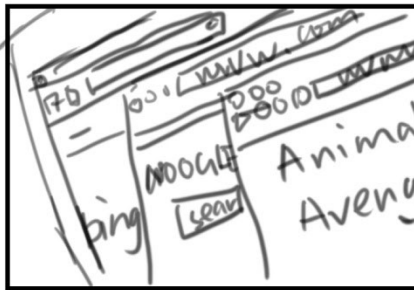
Luis is a government officer.
His work is to protect animals.



Once he saw a tiger facing
problems but can't do anything to help animals.



He searches the website,
but find useless information.



Then, he found the
website Animal Avenue.



He has learned many
tips and become animals'
best friend.

Figure 32 Storyboard



Lily is a uni-student
major in biology. She is
also a photographer.



She loves travelling
and cares about animal
protection.



Once she met a puppy
and it was sick.



she tried to find some
treatments in the book.
but there is no use.



Then, she braced on
website and connect
the animal hospital.



She thought it was a
good way to protect animal
and share her experience to
others.

Figure 33 Storyboard

c. Scenario

Scenario 1: As an animal protection government officer in QLD, Luis is a handsome and energetic gentleman, enjoys working with animals every day. He is a very warm-hearted person and love to help others. He loves horse most and he likes other wild animals as well. During the work time, he should find a lot of information related to animal protection.

Today, he arrives office at 9:00 am. After the morning meeting, he starts one wild kangaroo protection project. He opens the browser and search on the Google for any news with respect to the animal protection office but failed. He finds much of this information, including phone number, address and website are invalid. He has no idea where to get the latest news of animal protection in QLD and doesn't know how to connect with some NGO organisations. He is worried about animal extinction but doesn't know where to get the data. However, when he finds our website, he can easily get information about animal protection and connect with other related departments. He clicks the news button and finds some latest news in QLD, which are very useful in his work. He also clicks the emergency contact button and gets in touch with officers from other organization. He then writes an e-mail to his colleagues to recommend our website.

Scenario 2: Sophia works as a journalist in QLD. She is an optimistic girl with great energy. She is also an excellent photographer. She started to focus on animal protection when she raised a puppy in her childhood. Her favourite animal is dog and she is the mother of five dogs.

It is a normal sunny workday today. After feeding her five puppies, she goes to her office at 8:30 am. She has conducted the report with the topic of wild animal extinction rate in QLD. When she is suffering the internet, her team leader comes and says that her research is not related to the topic and need to review the report. She was very sad for her failure works. She has no idea on how to get data on animal extinction. Suddenly, she receives an e-mail from her best friend Luis, who is an animal protection officer. It says there is a useful animal protection website, which may help her somehow and the URL is copied as follows. She then clicks the link and browse this website. She finds some useful information and data about animal extinction, especially wild animals in QLD. And she starts to write her report again with big smile.

2.3 Prototype

a. Site map



Figure 34 Site map

b. Mock-up

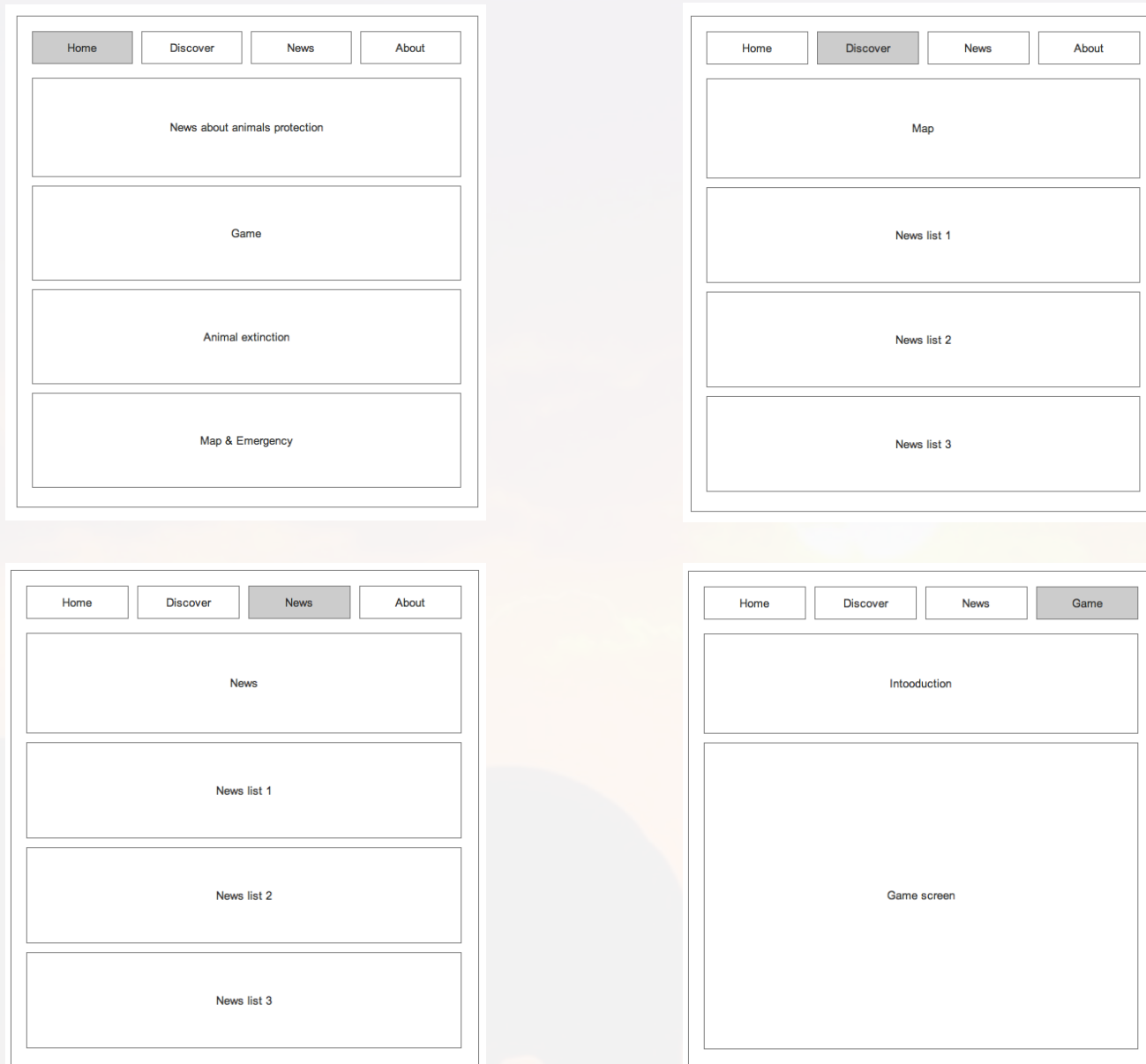


Figure 35 Mock-up

c. Low-fidelity prototype

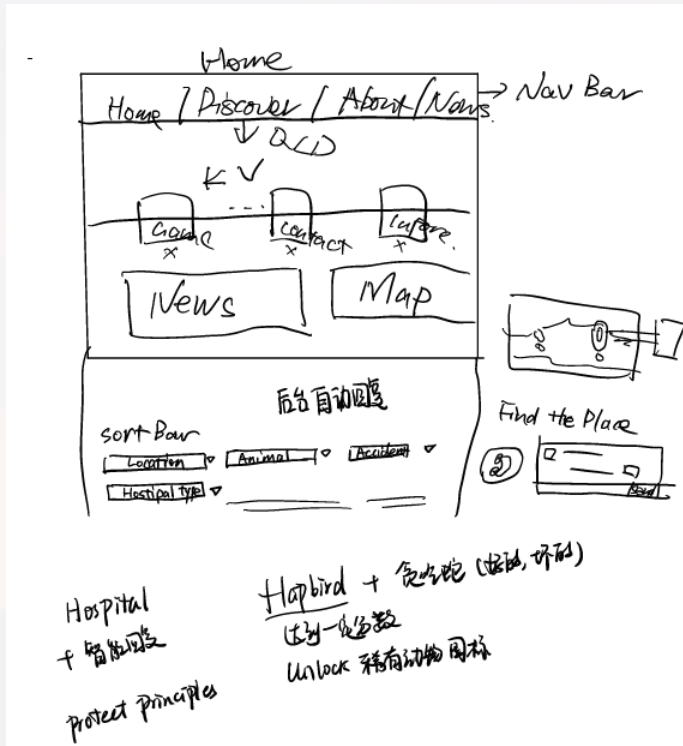


Figure 36 Concept

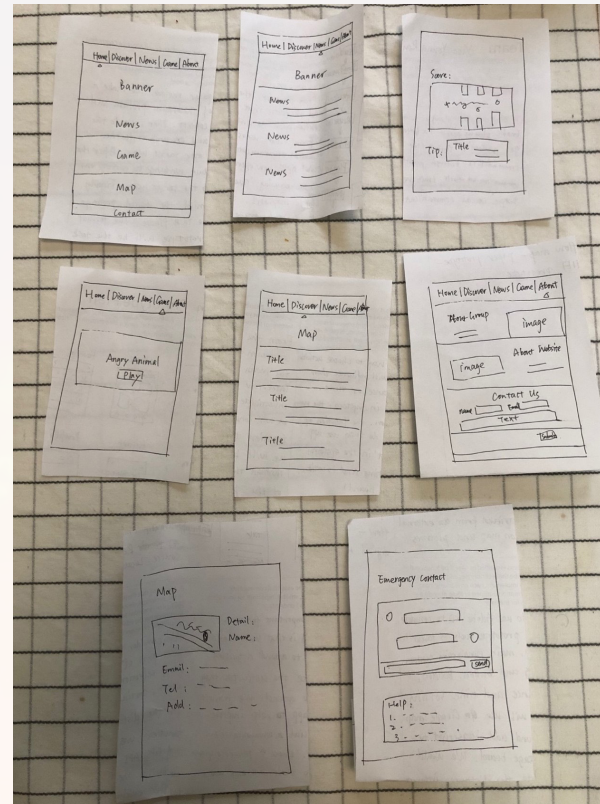


Figure 37 Paper prototype

d. High-fidelity prototype

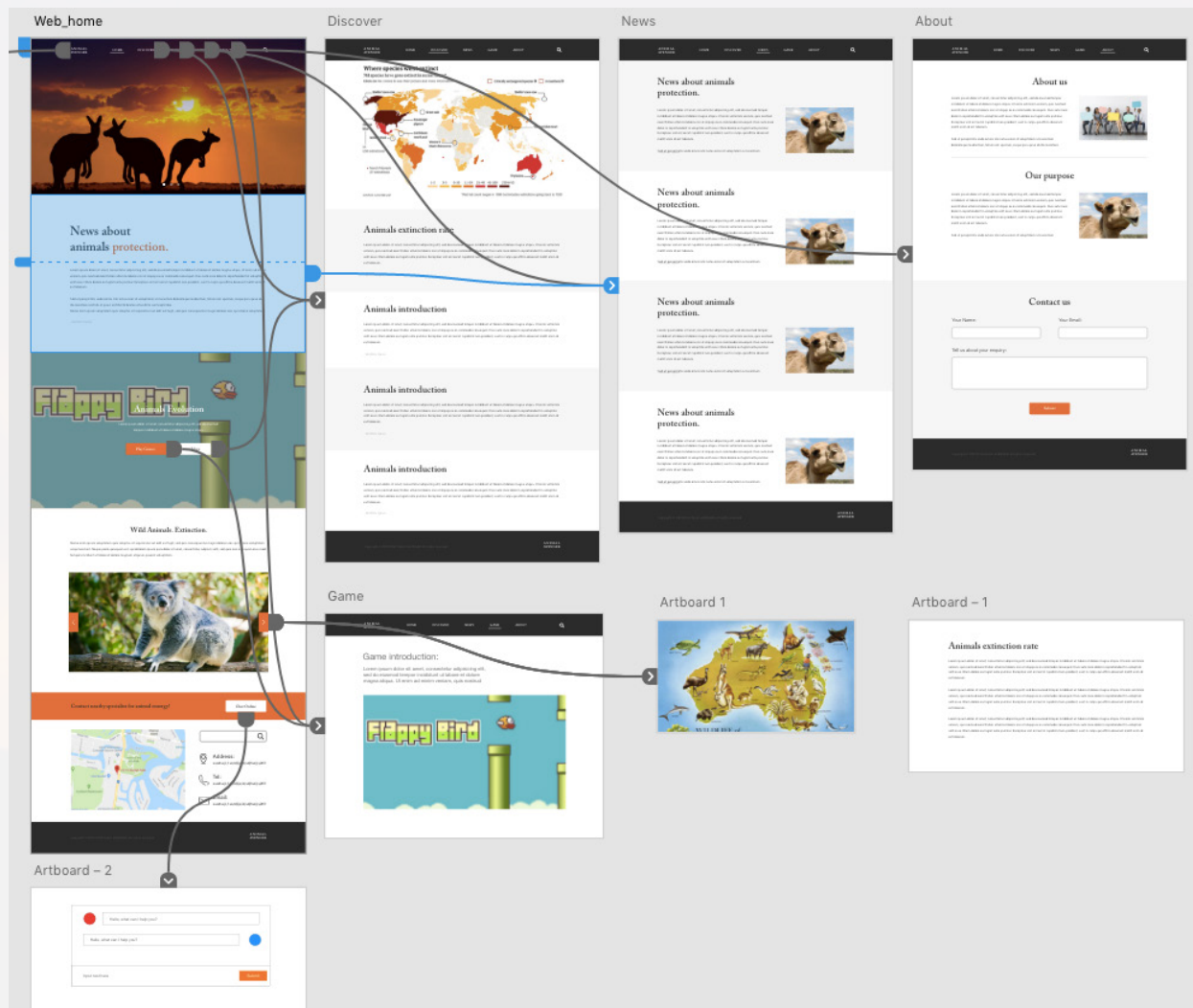


Figure 38 Homepage interaction

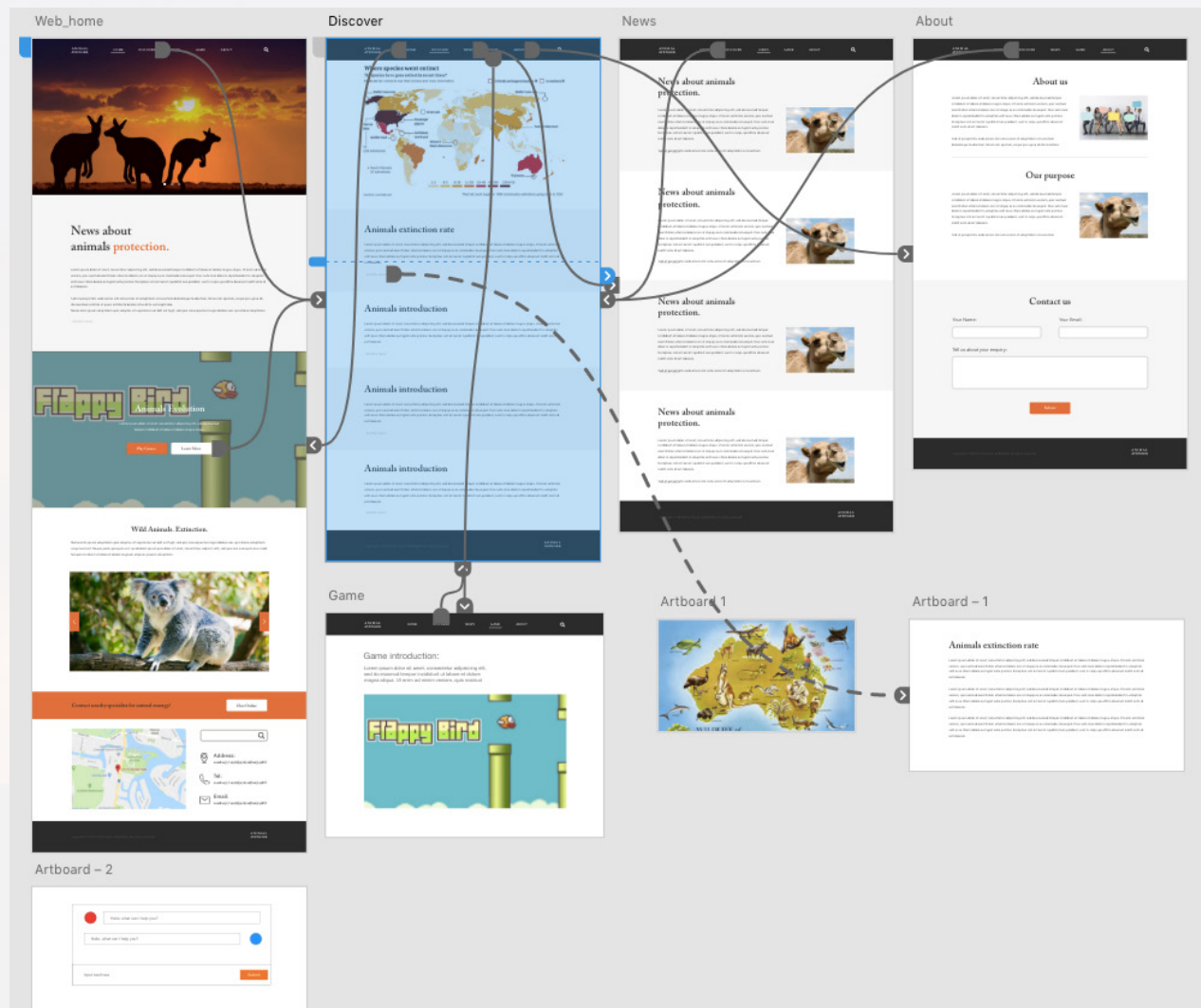


Figure 39 Discover page interaction

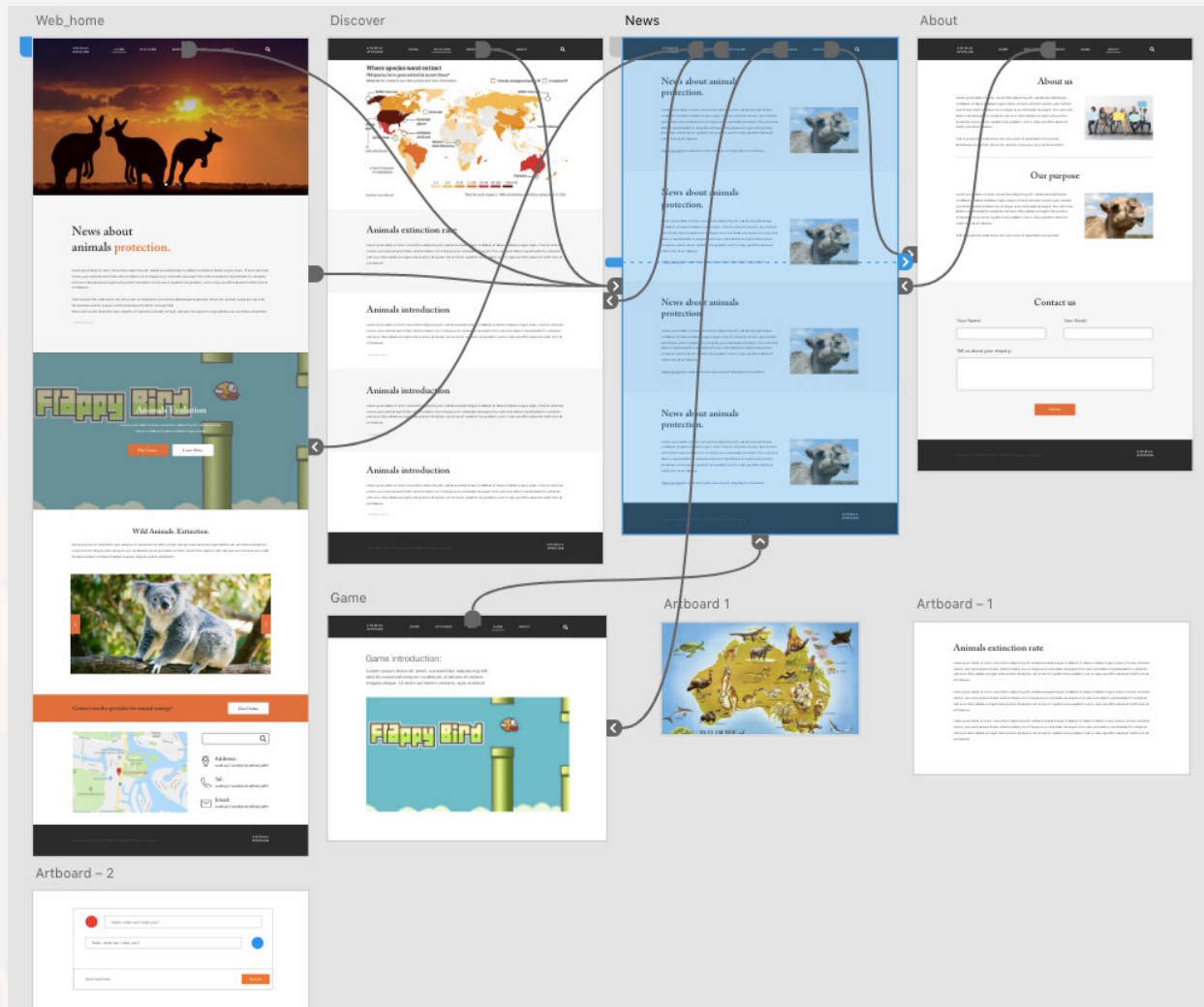


Figure 40 News page interaction

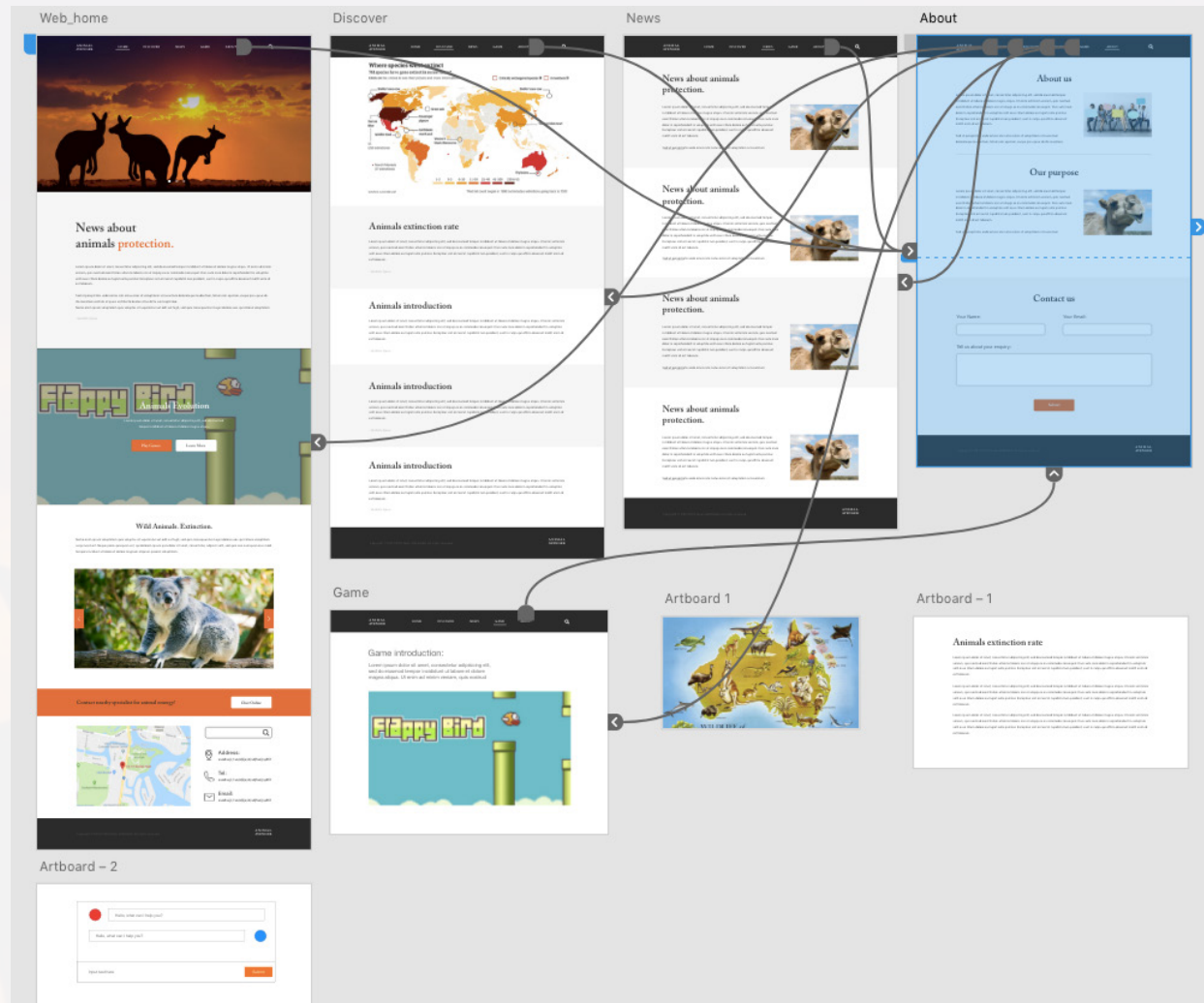


Figure 41 About page interaction

2.4 Website Development

a. Visualization

Generally, aesthetic contains five senses including: vision, hearing, touch, taste and smell. While aesthetic on website regards to visual and function, it is important to pay more attention on personality of a website and user experience.

The specific target audience zoology teachers, students, wildlife lovers and people who are going to contact with endangered wild animals. Based on their personalities, the visual aesthetic mainly focus on the two sections: visual aesthetic and functional aesthetic. Visual aesthetic is an emotional response, which is related to Australia animals and animals protection, with the mysterious dark colour to accord with the theme of wildlife. Functional aesthetic plays a vital role in how users understand educational contents in our website, which is about the communication state typically in the homepage (information and game functions) and AI emergency contact (map and chat room interface).

The images below are the two versions of the website. The first version uses orange as basic colour theme and the second one uses green. While green looks so attractive and shining that audience cannot see the texts clearly, we decide to use the first one.

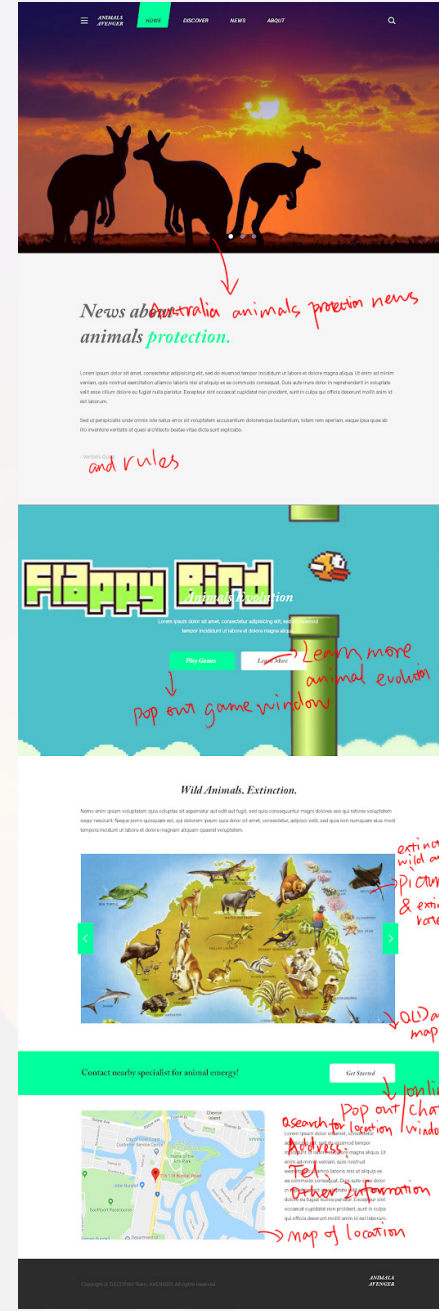
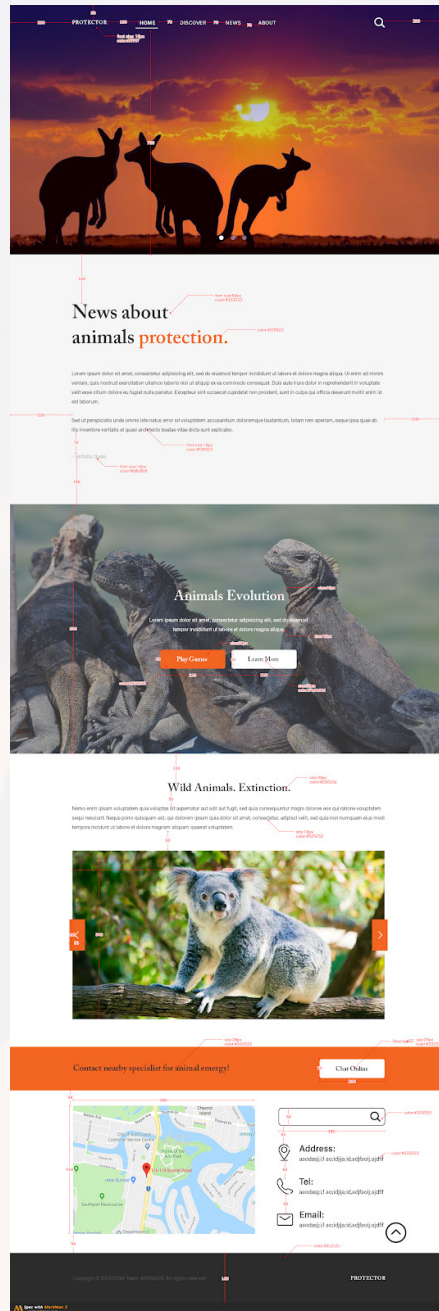


Figure 42 Webpage visualization

Colour scheme:

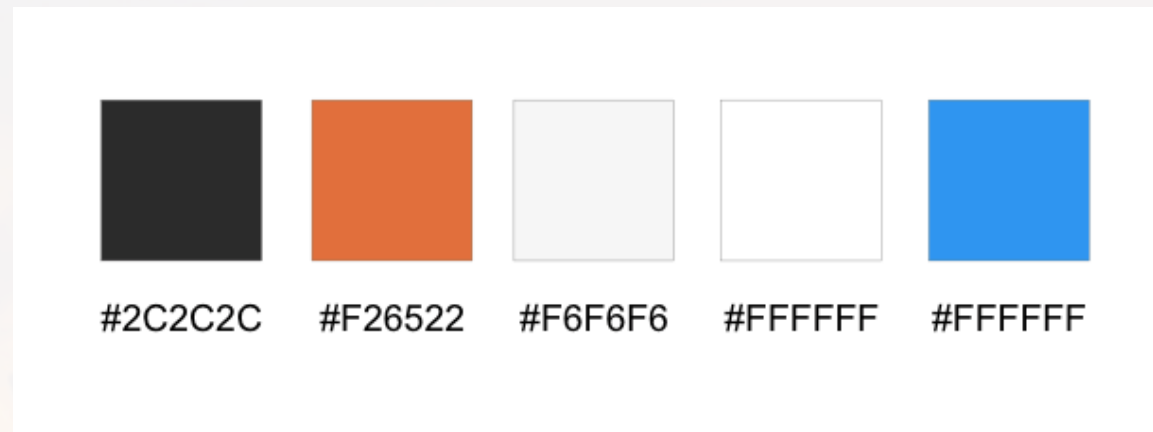


Figure 43 HEX code

Due to the background of animal protection, we choose white and orange as background and navigation colour, which seems bright and energetic. In addition, the text colour is grey and black, thus forming a sharp contrast with the background. In the primary pages (homepage), we set one banner image with orange and dark grey to fit the background colour. we choose the text to be white with the line hover in the navigation bar contrasted with the black background.

Text:

Text as one of the most important elements, it presents both visual and functional aesthetic in the website, which tends to deeply affect user's emotion and interaction. Under deliberation, we use Acumin Pro as the main text fonts in our website (navigation bar and detailed information). we also use Adobe Caslon Pro as the text font in secondary pages (discover, news and about), because it regards to the information and this kind of font seems more formal and make web pages more enchanting and professional. Talking about font size, we use 16px of navigation, 48px of heading and 16px of body in these pages to contract with each other creating a hierarchy layout, which make text more readable.

Text Location	Font	Size	Decoration
Navigation	Acumin Pro	16px	Regular
Heading	Adobe Caslon Pro	48px	Bold
Body text	Acumin Pro	16px	Regular
Tab	Adobe Caslon Pro	36px	Regular
Detailed information	Acumin Pro	16px	Regular
Landing page	Adobe Caslon Pro	24px	Bold

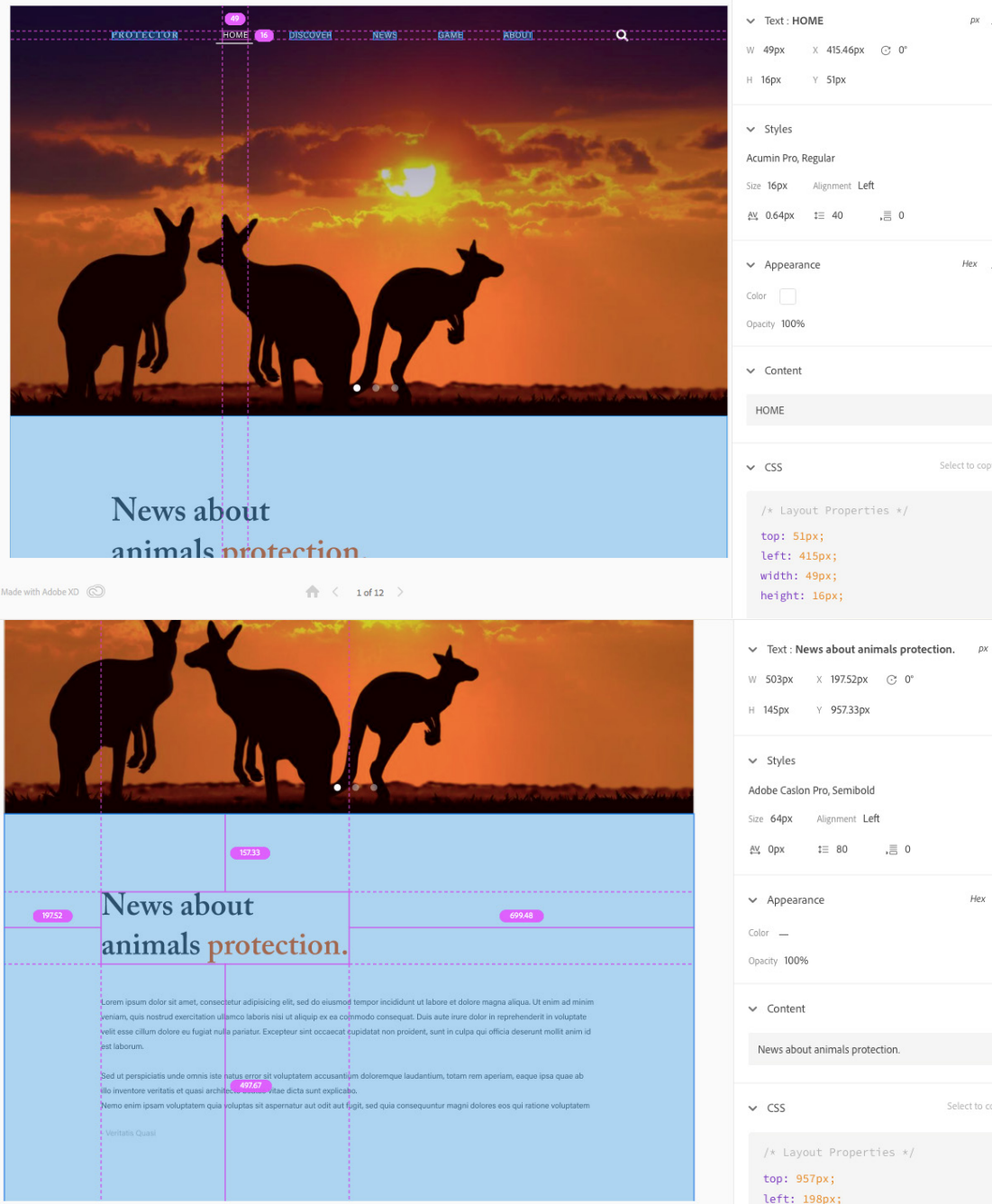
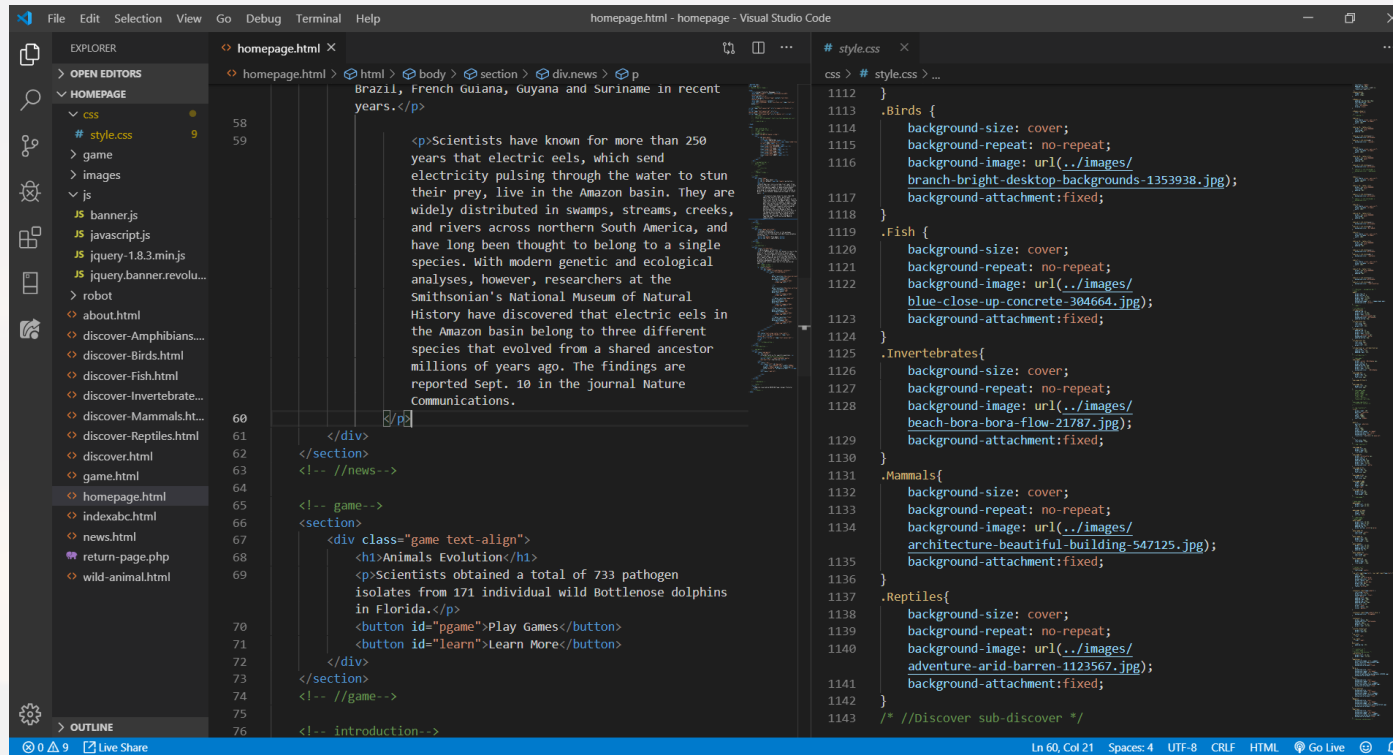


Figure 44 Aesthetic style

b. Front-end



The screenshot displays the Visual Studio Code editor with two files open: `homepage.html` and `style.css`. The `homepage.html` file shows a navigation menu with links to various discovery pages (e.g., `discover-Amphibians...`, `discover-Birds.html`, `discover-Fish.html`, `discover-Invertebrate...`, `discover-Mammals...`, `discover-Reptiles.html`, `discover.html`, `game.html`, `homepage.html`, `indexabc.html`, `news.html`, `return-page.php`, `wild-animal.html`). The main content area contains a paragraph about electric eels and a section titled "Animals Evolution" with a paragraph about pathogen isolates and two buttons: "Play Games" and "Learn More". The `style.css` file shows CSS rules for various categories: `.Birds`, `.Fish`, `.Invertebrates`, `.Mammals`, and `.Reptiles`. Each rule sets `background-size: cover`, `background-repeat: no-repeat`, `background-image: url(...)`, and `background-attachment: fixed`.

Figure 45 Homepage Html & CSS

The image above shows that all the code except game, maps and chatting robot. The challenge in this part is application of CSS layout and how to apply and connect other functions into these pages. For CSS layout, position, float, margin, padding methods are combined to use.

The image shows a screenshot of the Visual Studio Code editor interface. The Explorer sidebar on the left shows a project structure with folders for 'css', 'game', 'images', and 'js', and various HTML files. The main editor area displays the code for 'homepage.html'. The code includes a section for 'five species per year' with a list of slides. Each slide is defined with a data-transition property (e.g., '3dcurtain-horizontal', '3dcurtain-vertical', 'papercut', 'turnoff', 'flyin') and a data-masterspeed property set to '300'. The slides are contained within a 'fullwidthbanner-container' and a 'fullwidthbanner' div. The code also includes a script tag for 'pg_bd_ad_720x90.js' and a 'follow.js' script. The status bar at the bottom indicates 'Ln 65, Col 17' and 'Spaces: 4'.

```
five species per year.</p>
<!-- Images Slides-->
<!-- Slides-Online-->
<b class="slides">
  <div id="wrapper">
    <div class="fullwidthbanner-container">
      <div class="fullwidthbanner">
        <ul>
          <li data-transition="3dcurtain-horizontal" data-slotamount="15" data-masterspeed="300">
            
          </li>
          <li data-transition="3dcurtain-vertical" data-slotamount="15" data-masterspeed="300" data-link="#">
            
          </li>
          <li data-transition="papercut" data-slotamount="15" data-masterspeed="300" data-link="#">
            
          </li>
          <li data-transition="turnoff" data-slotamount="15" data-masterspeed="300">
            
          </li>
          <li data-transition="flyin" data-slotamount="15" data-masterspeed="300">
            
          </li>
        </ul>
      </div>
    </div>
  </div>
</b>
<div style="text-align:center;clear:both;">
  <script src="pg_bd_ad_720x90.js" type="text/javascript"></script>
  <script src="/follow.js" type="text/javascript"></script>
</div>
<!-- //slides-Online-->
</div>
</section>
<!-- //introduction-->
```

Figure 46 Html & CSS

It is a challenge to make a smooth slide function. JavaScript and many data properties in CSS are combined to achieve this function.

```

<script>
  $('.tile')
  // tile mouse actions
  .on('mouseover', function(){
    $(this).children('.photo').css({'transform': 'scale('+ $(this).attr('data-scale') +')'});
  })
  .on('mouseout', function(){
    $(this).children('.photo').css({'transform': 'scale(1)'});
  })
  .on('mousemove', function(e){
    $(this).children('.photo').css({'transform-origin': ((e.pageX - $(this).offset().left) / $(this).width()) * 100 + '%' + ((e.pageY - $(this).offset().top) / $(this).height()) * 100 + '%'});
  })
  // tiles set up
  .each(function(){
    $(this)
    .append('<div class="photo"></div>')
    .append('<div class="x">'+ $(this).attr('data-scale') +'x</div>ZOOM ON<br>HOVER</div>')
    .children('.photo').css({'background-image': 'url('+ $(this).attr('data-image') +')'});
  })
</script>

```

Figure 47 JavaScript

Zooming pictures, another challenge, used JavaScript in this part for improving user interaction. First, a photo container is added. Then some text which shows the zoom level on current item on the picture. Lastly, set up a background image for each tile based on data-image attribute.

c. Emergency map

```
function initMap() {  
  
    infoWindow = new google.maps.InfoWindow({  
        content: document.getElementById('informationshwoing')  
    });  
  
    map = new google.maps.Map(document.getElementById('map'), {  
        zoom: 4,  
        center: {lat: -25.3, lng: 133.8},  
    });  
  
    places = new google.maps.places.PlacesService(map);  
  
    autocomplete = new google.maps.places.Autocomplete((  
        document.getElementById('autocomplete')), {  
        componentRestrictions: {'country': 'au'},  
        types: ['(regions)']  
    });  
  
    autocomplete.addListener('place_changed', DisplayPlaceChanged);  
}
```

Figure 48 Setup map

In the emergency part, the map was built based on google map function. Initially, we use init-Map() function to initialize the map through setting its center position, the number of view-zoom, and set the search place within the Australia.

```

function DisplayPlaceChanged() {
    var place = autocomplete.getPlace();
    map.panTo(place.geometry.location);
    map.setZoom(15);
    hopitalSearch();
}

function hopitalSearch() {
    var hopitalSearch = {
        bounds: map.getBounds(),
        types: ['hospital']
    };

    places.nearbySearch(hopitalSearch, function(results, status) {
        if (status === google.maps.places.PlacesServiceStatus.OK) {
            clearOriginalMarkers();
            clearOriginalResults();

            for (var a = 0; a < results.length; a++) {
                markerIcon = 'https://img.icons8.com/officexs/30/000000/marker.png';

                markers[a] = new google.maps.Marker({
                    position: results[a].geometry.location,
                    icon: markerIcon,
                    animation: google.maps.Animation.DROP,
                });

                markers[a].placeResult = results[a];
                google.maps.event.addListener(markers[a], 'click', showInfoWindowInformation);
                setTimeout(Markerdropping(a), a * 130);
                Resultadding(results[a], a);
            }
        }
    });
}

```

Figure 49 Help search method

DisplayPlaceChanged() method is a type of help method to show the newest place and do the hospital search for the type of hospital. And then nearbysearch will clear all previous markers and results then for each new result, add a marker for them.

```

function showInfoWindowInformation() {
    var marker = this;
    places.getDetails({placeId: this.placeResult.place_id},
        function(place, status) {
            if (status === google.maps.places.PlacesServiceStatus.OK) {
                InfoWindContentbuilder(place);
                targetname.innerHTML = place.name;
                addressrow.innerHTML = place.vicinity;
                tellphoneid.innerHTML = place.formatted_phone_number;
            }
            else{
                return;
            }
        });
}

function Resultadding(result, i) {
    var results = document.getElementById('results');
    var tr = document.createElement('tr');
    tr.onclick = function() {
        google.maps.event.trigger(markers[i], 'click');
    };
}

function InfoWindContentbuilder(place) {
    document.getElementById('address-td').textContent = place.vicinity;

    if (place.formatted_phone_number) {
        document.getElementById('phonenum-row').style.display = '';
        document.getElementById('phonenum').textContent =
            place.formatted_phone_number;
    } else {
        place.formatted_phone_number = '';
        document.getElementById('phonenum-row').style.display = 'none';
    }
}

```

Figure 50 Information showing

For the information showing, we used `document.getElementById()` to read information and set these variable information through `innerHTML`. And if the information for the specific variable is none, then in the information showing window, it will be empty.

Besides, the maps built in discovery page was implemented similarly. However, because the data from SQL is very little, thus, we just record each data one by one. And then when user click the marker, the information will be shown in the below form.

```

var distributionLocation = [
  ['The Lace-eyed Tree Frog occurred throughout the Wet Tropics Bioregion from Paluma to Cooktown, northern Queensland, Australia. The frog is found in rainforest and wet sclerophyll forest. It is a small to medium-sized frog growing to 50 mm in length. The dorsal surface is richly patterned with dark brown and black spots and blotches. The ventral surface is white with dark brown spots. It is a common frog in its range. Chytridiomycosis is an infectious disease affecting amphibians worldwide. The disease has been recorded in four regions of Queensland. Read More</p></p> <a href ="http://www.environment.gov.au/cgi-bin/sprat/public/publicspecies.pl?taxon_id=86707">Read More</a></p>'],
  ['The Common Mistfrog historically occurred from Broadwater Creek National Park to Amos Bay, northern Queensland, Australia. The frog is found in rainforest and wet sclerophyll forest. It is a small to medium-sized frog growing to 50 mm in length. The dorsal surface is dull grey or brown coloured frog, with irregular darker markings. The ventral surface is white with dark brown spots. It is a common frog in its range. Chytridiomycosis is an infectious disease affecting amphibians worldwide. The disease has been recorded in four regions of Queensland. Read More</p></p> <a href ="http://www.environment.gov.au/cgi-bin/sprat/public/publicspecies.pl?taxon_id=1802">Read More</a></p>'],
  ['The Eungella Day Frog is restricted to the ranges west of Mackay, mid-eastern Queensland, from Clarke Range in the north to the Eungella Range in the south. The frog is found in rainforest and wet sclerophyll forest. It is a small frog growing 25 to 28 mm (males) or 27.6 to 30 mm (females). The dorsal surface is brown with dark brown spots and blotches. The ventral surface is white with dark brown spots. It is a common frog in its range. Chytridiomycosis is an infectious disease affecting amphibians worldwide. The disease has been recorded in four regions of Queensland. Read More</p></p> <a href ="http://www.environment.gov.au/cgi-bin/sprat/public/publicspecies.pl?taxon_id=1887">Read More</a></p>'],
  ['Fleay Frog is narrowly and disjunctly distributed in wet forests from the Conondale Range in south-east Queensland to the Eungella Range in the north. The frog is found in rainforest and wet sclerophyll forest. It is a large fossorial frog growing to 90 mm (snout-vent length 63 to 89 mm). The dorsal surface is brown with dark brown spots and blotches. The ventral surface is white with dark brown spots. It is a common frog in its range. Chytridiomycosis, caused by the chytrid fungus, is an infectious disease affecting amphibians worldwide. The disease has been recorded in four regions of Queensland. Read More</p></p> <a href ="http://www.environment.gov.au/cgi-bin/sprat/public/publicspecies.pl?taxon_id=25960">Read More</a></p>']
]

```

Figure 51 Record information

Initially, we create a 2-D list called `distributionLocation` to store all animals information. For each animal, it contains distribution information, latitude and longitude, animal description and potential threats.

```

var Tempmap = new google.maps.Map(document.getElementById('mapshow'), {
  center: new google.maps.LatLng(-20.92, 142.70),
  zoom: 5,
  // mapTypeId: google.maps.MapTypeId.ROADMAP
});

var Tempinfowindow = new google.maps.InfoWindow();

var marker, i;

for (i = 0; i < distributionLocation.length; i++) {
  marker = new google.maps.Marker({
    position: new google.maps.LatLng(distributionLocation[i][1], distributionLocation[i][2]),
    map: Tempmap
  });

  google.maps.event.addListener(marker, 'click', (function(marker, i) {
    return function() {
      Tempinfowindow.setContent(distributionLocation[i][0]);
      AnimalsDistribution.innerHTML = distributionLocation[i][0];
      AnimalsDescription.innerHTML = distributionLocation[i][4];
      threat.innerHTML = distributionLocation[i][5];
    }
  })(marker, i));
}

```

Figure 52 Read information and show in form

Through adding listener when click each mark, we read information from this 2-D list through its index. And then display it with innerHTML (i.e innerHTML is used to set the value of variable).

d. Game

Game interface

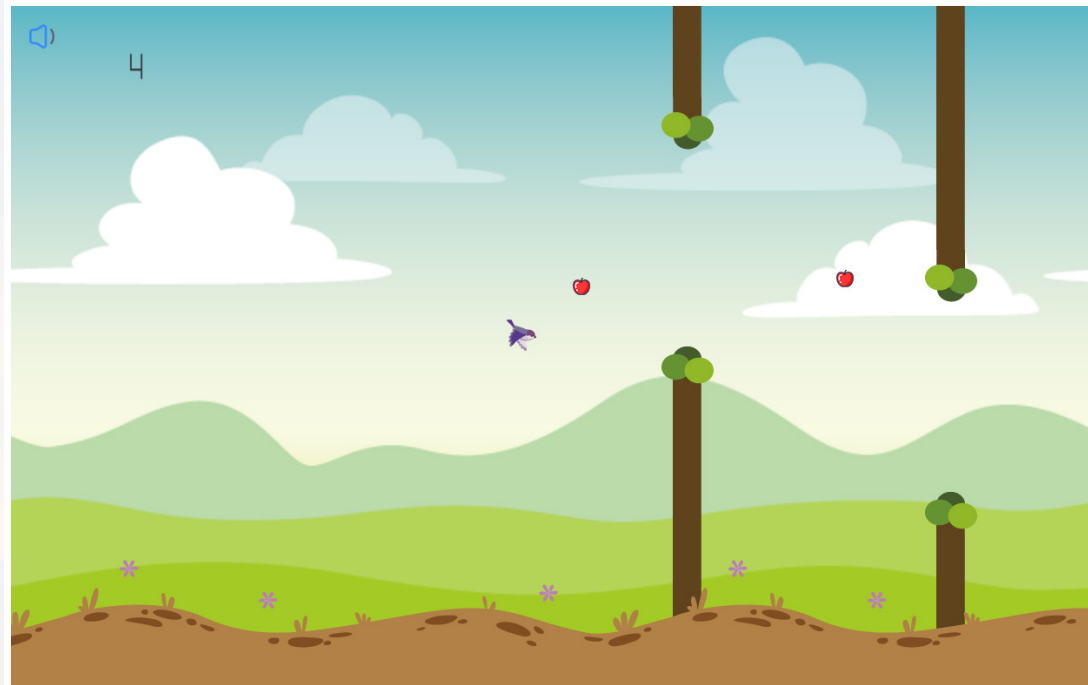


Figure 53 Game GUI

We implemented a game called "Super Animal" to mobilize the enthusiasm of website users. In this game, the user can manipulate the game character to avoid the trees and fly forward. The farther the flight is, the higher the score is. At the same time, the character can also score points when eating the bonus apple during the flight. Scores can unlock new game characters. During the game, when the game character eats the bonus apple, a tip pops up to show some knowledge about the endangered animals. Users can click "more" to learn more, or click "close" to continue the game. After inserting the knowledge of endangered animals during the game, the longer the user plays the game, the more knowledge they will learn.

In order to implement this game, we divided the game into several parts to write code.

```

/**
 * Game character
 */
var character = {
  flyTimer:null,
  animationTimer:null,
  characterPause:false,
  div:document.createElement("div"),

  /**
   * Show the game character
   * @param {div} parentObj Parent Div
   */
  displayCharacter:function(parentObj) {
    this.div.style.width = "40px";
    this.div.style.height = "40px";
    this.div.style.backgroundImage = "url(img/"+playerCharacter+"0.png)";
    this.div.style.backgroundRepeat = "no-repeat";
    this.div.style.position = "absolute";
    this.div.style.left = browserWidth * 0.45 + "px";
    this.div.style.top = "200px";
    this.div.style.zIndex = "1";
    parentObj.appendChild(this.div);
  },

  fallSpeed: 0,
  /**
   * Game character fly status. It will have a 2px fall speed and when it hits
   * the ground or game pauses, it will stop flying.
   */
  flyCharacter: function(){
    character.characterPause = false;
    character.flyTimer = setInterval(fly,40);
    function fly() {
      character.div.style.top = character.div.offsetTop + character.fallSpeed++
+ "px";
      if (character.div.offsetTop < 0) {
        character.fallSpeed = 2;
      }
      if (character.div.offsetTop >= browserHeight*0.88 || character.
characterPause == true) {
        character.fallSpeed = 0;
        clearInterval(character.flyTimer);
        clearInterval(character.animationTimer);
      }
      if (character.fallSpeed > 15) {
        character.fallSpeed = 15;
      }
    }
  },

  /**
   * Show the animation when the character flying
   */
  animation: function() {
    var up = ["url(img/up_"+playerCharacter+"0.png)", "url(img/up_"+
playerCharacter+"1.png)"];
    var down = ["url(img/down_"+playerCharacter+"0.png)", "url(img/down_"+
playerCharacter+"1.png)"];
    var i = 0, j = 0;
    character.animationTimer = setInterval(picChange,120);
    function picChange() {
      if (character.fallSpeed > 0) {
        character.div.style.backgroundImage = down[i++];
        if (i==2) {i = 0}
      }if (character.fallSpeed < 0) {
        character.div.style.backgroundImage = up[j++];
        if (j==2) {j = 0}
      }
    }
  }
};

```

Character.js

This is the game character.

In the display function, we created a div for the character and added it as a child of the target div.

In the fly function, we wrote a new function to drop the character by two pixels and call this function every 40 milliseconds through setInterval to achieve the character's flight in the vertical direction. When the game is paused or the character drops to the ground, the flight stops. The code for the character to fly up is in Main.js, which we will cover later. The flight in the horizontal direction is achieved by setting the ground, the trees and the bonus apple moving in the opposite direction.

In the animation function, we created an array for the two background images of the character, and wrote a function to cycle through the background image in the array. This function is called every 120 milliseconds by setInterval. Animation effect.

Figure 54 Character

```

/**
 * Game Tree
 */
function Tree() {
  this.upTree = null;
  this.downTree = null;
  this.downHeight = object.randomNum(browserHeight*0.1,browserHeight*0.5);
  this.gapHeight = object.randomNum(browserHeight*0.5,browserHeight*0.5+10);
  this.upHeight = browserHeight - this.downHeight - this.gapHeight;

  /**
   * Help method to create the div
   */
  this.createFrame = function(url, height, positionType, left, top) {
    var frame = document.createElement("div");
    frame.style.width = "60px";
    frame.style.height = height;
    frame.style.position = positionType;
    frame.style.left = left;
    frame.style.top = top;
    frame.style.backgroundImage = url;
    return frame;
  };

  /**
   * Create the up tree and down tree then add them to the parent div
   */
  this.createTree = function() {
    var upTreeTrunk = this.createFrame("url(img/up_trunk.png)", this.upHeight +
    "px");
    var upTreeCrown = this.createFrame("url(img/up_crown.png)", "60px");
    this.upTree = this.createFrame(null, null, "absolute", "100%");
    this.upTree.appendChild(upTreeTrunk);
    this.upTree.appendChild(upTreeCrown);

    var downTreeCrown = this.createFrame("url(img/down_crown.png)", "59px");
    var downTreeTrunk = this.createFrame("url(img/down_trunk.png)", this.
    downHeight + "px");
    this.downTree = this.createFrame(null, null, "absolute", "100%", browserHeight
    *0.85 - this.downHeight + "px");
    this.downTree.appendChild(downTreeCrown);
    this.downTree.appendChild(downTreeTrunk);

    jsGameFrame.appendChild(this.upTree);
    jsGameFrame.appendChild(this.downTree);
  };

  /**
   * Move the tree
   */
  this.moveTree = function() {
    this.upTree.style.left = this.upTree.offsetLeft - 3 + "px";
    this.downTree.style.left = this.downTree.offsetLeft - 3 + "px";
  };
}

```

Tree.js

This is game's tree.

In the createFrame function we implemented a helper method for creating divs.

In the createTree function, we stack the trunk and add the crown at the end to achieve a random tree height.

In the moveTree function, we move the tree to the left by setting the left style of the tree div to indicate that the character is flying to the right.

Figure 55 Tree

```

/**
 * Game bonus apple
 */
function Bonus() {
    this.bonusFrame = null;
    this.top = object.randomNum(30,370);

    /**
     * Create frame(div) function
     */
    this.createBonusFrame = function(url, positionType, left, top) {
        var frame = document.createElement("div");
        frame.style.width = "20px";
        frame.style.height = "20px";
        frame.style.position = positionType;
        frame.style.left = left;
        frame.style.top = top;
        frame.style.backgroundColor = url;
        return frame;
    };

    /**
     * Create bonus apple
     */
    this.createBonus = function() {
        var bonus = this.createBonusFrame("url(img/bonus.png)");
        this.bonusFrame = this.createBonusFrame(null, "absolute", "100%", this.top +
"px");
        this.bonusFrame.appendChild(bonus);
        jsGameFrame.appendChild(this.bonusFrame);
    };

    /**
     * Move bonus apple
     */
    this.moveBonus = function() {
        this.bonusFrame.style.left = this.bonusFrame.offsetLeft - 3 + "px";
    };
}

```

Bonus.js

This is the game's bonus apple.

The createBonusFrame function is similar to the createFrame function in Tree.js.

The createBonus function is similar to the createTree function in Tree.js.

The moveBonus function is similar to the moveTree function in Tree.js

Figure 56 Bonus

```

/**
 * Basic object function
 */
var object = {
  /**
   * Get a random number between min and max
   * @param {int} min min number
   * @param {int} max max number
   */
  randomNum: function(min, max) {
    return parseInt(Math.random() * (max - min + 1) + min);
  },

  /**
   * Detecting whether object1 collides with the object2
   * @param {div} object1 Object1
   * @param {div} object2 Object2
   */
  collisionDetection: function (object1, object2) {
    var object1Left = object1.offsetLeft;
    var object1Width = object1.offsetLeft + object1.offsetWidth;
    var object1Top = object1.offsetTop;
    var object1Height = object1.offsetTop + object1.offsetHeight;

    var object2Left = object2.offsetLeft;
    var object2Width = object2.offsetLeft + object2.offsetWidth;
    var object2Top = object2.offsetTop;
    var object2Height = object2.offsetTop + object2.offsetHeight;

    if (!(object1Left > object2Width || object1Width < object2Left ||
object1Top > object2Height || object1Height < object2Top)) {
      return true;
    }
    return false;
  },
};

```

Crash.js

This is the basic functions of the game.

The randomNum function can get a random number between the maximum and minimum values of an input.

In the collisionDetection function, the position of two objects is used to determine whether or not to collide. If the collision returns true, otherwise it returns false.

Figure 57 Basic functions

```

var distributionLocation = [
  [
    'The Lace-eyed Tree Frog occurred throughout the Wet Tropics Bioregion from Paluma to Cooktown, northern Queensland, at altitudes between 0 and 1200 m (McDonald 1992).',
    -19.173428, 145.963135, 4
  ],
  '<h>Litoria dayi</h>'+
  '<p>This is a small to medium-sized frog growing to 50 mm in length. The dorsal surface is rich-brown to orange-brown with or without scattered cream or lichen like spots and blotches covering it, the head or the limbs.</p>'
  ,
  '<p>Chytridiomycosis is an infectious disease affecting amphibians worldwide. The disease has been recorded in four regions of Australia, namely the east coast, south-west Western Australia, Adelaide, and more recently Tasmania. This highly virulent fungal pathogen of amphibians is capable at the minimum of causing sporadic deaths in some populations, and 100 per cent mortality in other populations (AGDEH 2006). Feral pigs are a potential cause of riparian habitat damage and adult frog mortality. The activity of feral pigs has increased over the period 1989-1992 in an area previously inhabited by the Lace-eyed Tree Frog. However, there is very little research into the impact of feral pigs on native frog populations (Richards et al. 1993).</p>'
  +
  '<p> <a href = "http://www.environment.gov.au/cgi-bin/sprat/public/publicspecies.pl?taxon_id=86707">Read More</a></p>'
  ],

```

Tips.js

In Tips, we use arrays to store information about endangered animals.

Figure 58 Tips

```

var jsGameFrame = document.getElementById("gameFrame");
var jsHeadTitle = document.getElementById("headTitle");
var jsHeadCharacter = document.getElementById("headCharacter");
var jsVoiceButton = document.getElementById("voice");
var treeArr = [];
var bonusArr = [];
var usedtreeArr = [];
var treeDistance = 200;
var browserWidth =document.body.clientWidth;
var browserHeight = document.body.clientHeight;
var voiceBool = false;

var jsScore = document.getElementById("score");
var jsNum1 = document.getElementById("num1");
var jsNum2 = document.getElementById("num2");
var jsNum3 = document.getElementById("num3");
var score = 0;

var jsBottomLand1 = document.getElementById("bottomLand1");
var jsBottomLand2 = document.getElementById("bottomLand2");

var jsTips = document.getElementById("tips");
var jsTipsContent = document.getElementById("tipText");
var jsClose = document.getElementById("close");
var jsMore = document.getElementById("more");
var jsGameOver = document.getElementById("gameOver");
var jsReplay = document.getElementById("replay");
var jsThisScore = document.getElementById("thisScore")
var jsHistoryScore = document.getElementById("historyScore")

var playerCharacter = $('input:radio:checked').val();

var Y = 3;
var index = 0;
var imgArr = ["img/"+playerCharacter+"0.png", "img/"+playerCharacter+"1.png"]

var voiceIndex = 0;
var voiceArr = ["img/close.png", "img/open.png"]

var loop= new Audio("radio/Loop.wav");
var loopPlay;
var skill= new Audio("radio/Skill.mp3");
var over= new Audio("radio/GameOver.wav");

```

Main.js

In the main function of the game, we first define the various variables required by getElementById

Figure 59 Main variables

```
/**
 * Set the head character animation
 */
var headWaveTimer = setInterval(headWave,200);
function headWave() {
    Y *= -1;
    jsHeadTitle.style.top = jsHeadTitle.offsetTop + Y + "px"
;   jsHeadCharacter.src = imgArr[index++];
    if (index == 2) {
        index = 0;
    }
}
```

Figure 60 Head animation

We wrote the headWave function, changed the position of the HeadTitle, and looped through the array to change the background image of the headCharacter.

We use the setInterval to call the headWave function every 200 milliseconds to achieve the animation of the title.


```
/**
 * Main function of the game
 */
var mainTimer = setInterval(mainRun,30);
function mainRun() {
    if (jsBottomLand1.offsetLeft <= -2100) {
        jsBottomLand1.style.left = "2100px";
    }
    if (jsBottomLand2.offsetLeft <= -2100) {
        jsBottomLand2.style.left = "2100px";
    }
    // Move the land
    jsBottomLand1.style.left = jsBottomLand1.offsetLeft - 3 + "px";
    jsBottomLand2.style.left = jsBottomLand2.offsetLeft - 3 + "px";
}
```

Figure 61 Main_land

mainRun is the main function of the game, we use setInterval to call this function every 30 milliseconds to make the game run continuously.

The first part of mainRun is to move the land to the left, like moveTree, which is used to indicate that the character flies to the right.

```

if (treeArr.length) {
  for (var i = 0; i < treeArr.length; i++) {
    // Move tree
    treeArr[i].moveTree();

    // Detect if the game is over
    var x = object.collisionDetection(treeArr[i].downTree, character.div);
    var y = object.collisionDetection(treeArr[i].upTree, character.div);
    var z = character.div.offsetTop >= browserHeight*0.88 || character.div.offsetTop <
    if (x || y || z) {
      // End the game
      window.clearInterval(mainTimer);
      window.clearInterval(loopPlay);
      loop.pause();
      character.fallSpeed = 0;
      jsGameFrame.onclick = null;
      document.onkeydown = null;
      if (voiceIndex != 1) {
        over.play();
      }
      jsScore.style.display = "none";

      // Show the Game Over div
      jsGameOver.style.display = "block";
      var historyscore = localStorage.getItem('maxScore');
      if(historyscore == null || historyscore < score) {
        localStorage.setItem('maxScore', score);
        historyscore = score;
      }
      jsHistoryScore.innerHTML = historyscore;
      jsThisScore.innerHTML = score;
    }
  };

  // Create a new tree
  if (treeArr[treeArr.length - 1].downTree.offsetLeft < (browserWidth - treeDistance)) {
    var newTree = new Tree();
    newTree.createTree();
    treeArr.push(newTree);
  };

  // Display the score
  if (treeArr[0].downTree.offsetLeft == Math.ceil(browserWidth * 0.45) - 62
  || treeArr[0].downTree.offsetLeft == Math.ceil(browserWidth * 0.45) - 61
  || treeArr[0].downTree.offsetLeft == Math.ceil(browserWidth * 0.45) - 63) {
    score++;
    if (score < 10) {
      jsNum1.style.backgroundImage = "url(img/" + score + ".png)";
    } else if (score < 100) {
      jsNum2.style.display = "block";
      jsNum1.style.backgroundImage = "url(img/" + parseInt(score/10) + ".png)";
      jsNum2.style.backgroundImage = "url(img/" + score%10 + ".png)";
    } else if (score < 1000) {
      jsNum3.style.display = "block";
      jsNum1.style.backgroundImage = "url(img/" + parseInt(score/100) + ".png)";
      jsNum2.style.backgroundImage = "url(img/" + parseInt(score/10)%10 + ".png)";
      jsNum3.style.backgroundImage = "url(img/" + score%10 + ".png)";
    }
  };
  console.log(score);
};

```

This part is about the tree in the game.

First, for each tree we call the movetree function to move it to the left.

Secondly, there is the part that detects if the game is over. We call the collisionDetection function to determine whether the character collides with the tree and determine whether it collides with the ground or top through the position of the character. If the character collides with the tree or collides with the ground, the game ends, stop mainRun, stop playing background music, stop mouse and keyboard control. Play game ending music, display end screen and score.

Thirdly, there is the part that creates the new tree. When the previous tree is more than the treeDistance on the right side of the screen, we create a new tree.

Fourthly, there is the part that shows the score. When the character passes through the tree, that is, the character's offsetLeft is equal to the tree, one point is added. Since the tree moves three pixels at a time, there are three cases to consider when detecting equality. Calculate the ones, tens and hundred digits for the current score and display the corresponding picture.

The last part is to delete the tree that the character crossed through.

Figure 62 Main_tree


```

/**
 * Create the first bonus apple
 */
function createBonusDIV() {
    var bonus = new Bonus();
    bonus.createBonus();
    bonusArr.push(bonus);
}
var jsStartBtn = document.getElementById("startBtn");
/**
 * Start the game
 */
jsStartBtn.onclick = function() {
    jsHeadTitle.style.display = "none";
    clearInterval(headWaveTimer);
    jsStartBtn.style.display = "none";
    character.displayCharacter(jsGameFrame);
    character.flyCharacter();
    character.animation();
    // When click the frame, the character flies
    jsGameFrame.onclick = function(){
        character.fallSpeed = -8;
        if (voiceIndex != 1) {
            skill.play();
        }
    };
};

var pausFlag = false;
document.onkeydown = function(event){
    // When the space is pressed, the character flies
    if(event.keyCode == 32){
        character.fallSpeed = -8;
        if (voiceIndex != 1) {
            skill.play();
        }
    };
    // When the "P" is pressed, the game paused
    if(event.keyCode == 80 && !pausFlag) {
        window.clearInterval(mainTimer);
        character.characterPause = true;
        pausFlag = true;
    } else if (event.keyCode == 80 && pausFlag) {
        jsTips.style.display = "none";
        mainTimer = setInterval(mainRun,30);
        character.flyCharacter();
        character.animation();
        pausFlag = false;
    };
};
var tree = new Tree();
tree.createTree();
treeArr.push(tree);
jsNum1.style.display = "block";
setTimeout("createBonusDIV()", 2000);
}

```

This is the start function. The game starts when the start button is pressed. When the user clicks on the game interface or presses the spacebar, the character moves up 8 pixels and plays the flying music. What's more, the game will be paused when user presses "P". We use setTimeout to delay the creation of the first bonus apple for 2 seconds to ensure that the bonus apple is always in the interval of the tree.

Figure 64 Start function

```

/**
 * Replay the game
 */
jsReplay.onclick = function() {
    window.location.href = "game.html";
}

/**
 * Close and open music
 */
jsVoiceButton.onclick = function() {
    jsVoiceButton.src = voiceArr[voiceIndex++];
    if(voiceIndex == 1) {
        loop.pause();
        window.clearInterval(loopPlay);
        voiceBool = true;
    } else if (voiceBool & voiceIndex == 2){
        loopPlay = setInterval(loop.play(),84000);
    }
    if (voiceIndex == 2) {
        voiceIndex = 0;
    }
}
}

```

In the replay function, we click on the replay button to refresh the page and replay the game.

In the Voice function, we loop through the background image of the volume icon and determine if the volume is off.

Figure 65 Replay and Voice

```

/**
 * Select the game character
 */
$("#selectorSubmit").click(function() {
    $("#selector").css("display", "none");
    loopPlay = setInterval(loop.play(), 84000);
    playerCharacter = $('input:radio:checked').val();
    console.log(playerCharacter);
});

/**
 * Determine whether user can unlock the game character by bestScore
 */
$(document).ready(function(){
    var bestScore = localStorage.getItem("maxScore");
    console.log("Best Score: " + bestScore);
    if(bestScore > 26) {
        $("#animal3").next().html("koala");
        $("#animal3").css("visibility", "visible");
        $("#animal2").next().html("kangaroo");
        $("#animal2").css("visibility", "visible");
    } else if(bestScore > 6) {
        $("#animal2").next().html("kangaroo");
        $("#animal2").css("visibility", "visible");
    }
});

```

When the user clicks the submit button, the character information selected by the user is obtained and the background image of the game character is changed.

When the page is loaded, it will automatically get the bestScore through maxScore stored in localStorage to determine whether the user has unlocked other characters.

Figure 66 Select game character

Feedback and Database

The screenshot displays the phpMyAdmin interface for a MySQL database named 'Feedback'. The current view is the 'Table structure' for a table named 'feedback'. The table structure is as follows:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	name	varchar(255)	utf8_general_ci		No	None			Change Drop More
2	email	varchar(255)	utf8_general_ci		No	None			Change Drop More
3	text	text	utf8_general_ci		No	None			Change Drop More

The 'name' column is marked as the primary key. Below the table structure, there are options to add a new column, with a dropdown menu showing '1 column(s) after text'. The 'Indexes' section shows a primary index on the 'name' and 'email' columns. The 'Partitions' section is currently empty.

Figure 67 Table

First, we created a Feedback database using phpmyadmin, and created a new feedback table to store the user's feedback.

Server: localhost

Databases SQL Status User accounts Export Import Settings Replication

Global Database Change password Login Information

Edit privileges: User account 'feeduser'@'localhost'

Database-specific privileges

Database	Privileges	Grant	Table-specific privileges	Action
Feedback	INSERT	No	No	Edit privileges Revoke

Add privileges on the following database(s):

mysql
phpmyadmin
sys
test

Figure 68 User

Then we created a new user as feeduser, limiting its privileges to only execute insert operations in the Feedback database.


```

<!-- contact us -->
<div class="container1">
  <form method="POST">
    <h1>Contact us</h1>
    <br>
    <div class="justify-contact-us">
      <div class="row-contact-us">
        <div class="row-left">
          <label for="name">Your Name:</label>
          <input type="text" id="name" name="name" placeholder="">
        </div>
        <div class="row-right">
          <label for="emailY">Your Email:</label>
          <input type="text" id="emailY" name="emailY" placeholder="">
        </div>
      </div>
    </div>
    <div id="tell-us">
      <label for="subject">Tell us about your enquiry:</label>
      <textarea id="subject" name="subject" placeholder="" style="height:200px"></textarea>
    </div>

    <input type="submit" value="submit" onclick="
    Popupbox()">
    <script>
      function Popupbox(){
        alert("Thanks for submitting!");
      }
    </script>
  </div>
</form>
</div>
<!-- //contact us -->
<?php
  if($_POST) {
    $name = $_POST["name"];
    $email = $_POST["emailY"];
    $text = $_POST["subject"];
  }

  $servername = "localhost";
  $username = "feeduser";
  $password = "feedback";
  $dbname = "Feedback";

  $conn = new mysqli($servername, $username, $password, $dbname);
  if ($conn->connect_error) {
    die("Failed: " . $conn->connect_error);
  }
  $sql = "INSERT INTO feedback(name, email, text) VALUES (\\"$name\\", \\"$email\\", \\"$text\\")";
  $conn->query($sql);
  $conn->close();
?>

```

We added a feedback form on the About page, the method is post, then use the php code to log in to the database as feeduser, insert the “name”, “emailY” and “subject” contents of the page post into the feedback table.

Figure 69 Form

e. Chat robot

The chat robot is designed in two parts one part is the tag shown in the homepage, the other is the webpage hosting the dialogue frame and type in frame.

The tag is the gateway of the chat robot, it is basically saying “I am here, How can I help you?”

Because we need to insert the tag in the main page for easier accessibility. I choose to implement this part by inserting a `<div>` in the homepage, in order to makes it more accessible and not to interfere with other parts of the main page. I used `style="position:fixed"`; for positioning. I chose `target="_black"` to show the dialogue frame in the new windows: to avoid trouble for a separate styling file, I chose to use inline text for the tag:

```
<div id="onlinexm" style="position: fixed; box-sizing: content-  
ox; height: 43px; bottom: 0px;␣  
padding: 0px 20px; border-  
radius: 4px 4px 0px 0px; right: 100px; transition: all 0.3s ease-in-  
out 0.1s; background:blue;">␣  
  <a hreffocus="" id="popupbtn" href="../chatrobot-  
windows.html" target="_blank"␣  
    style="text-decoration: none; outline: none; font-  
family: Microsoft Yahei, Arial, Helvetica;␣  
    color: #fff; font-size: 16px; display: inline-  
block; margin: 0; padding: 0; border: none; line-  
height:43px; float:none;"><img␣  
    style="vertical-  
align: middle; width: 31px; border:none; margin: -  
4px 10px 0 0; float: none; position: static; height: 27px;">␣  
  <span style="font-  
family: Microsoft Yahei, Arial, Helvetica; color: #fff; font-  
size:16px; margin: 0; padding: 0; ">get help from robot Winnie</span></  
a>␣  
</div>
```

Figure 70 Tag

The second part of the chat-robot is the chat windows, the hard part is to create the desired response in JavaScript. In order to do this two function has been used:

```
function send(text){
    var curmsg={
        text:text,
        from:uuid,
        to:"robot"
    }
    socket.emit('newmessage',curmsg);
    console.log('send:' + curmsg);
    curmsg.html = generateChatBubble(curmsg);
    $scope.msg.text="";
    $scope.msgghistory.push(curmsg);
    $scope.showSugg = false;
    }
},0);
```

Figure 71 send function

This function is created to take in what used input and send them back in the users' speech bubble: The effects as following:

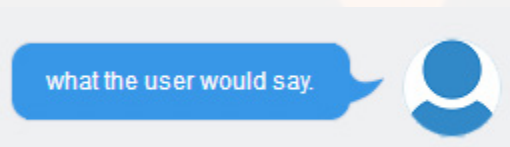


Figure 72 Icon

The other function is the send1 function, it actually works as a send back function, but instead I used send1 to name it for convenience:

```
function send1(text){
  var curmsg={
    text:text,
    from:"robot",
    to:"user123"
  }
  socket.emit('newmessage',curmsg);
  console.log('send:' + curmsg);
  curmsg.html = generateChatBubble(curmsg);

  $scope.msghistory.push(curmsg);
  $scope.showSugg = false;
}
```

Figure 73 Send back function

It is used to reply what the user just said, like:

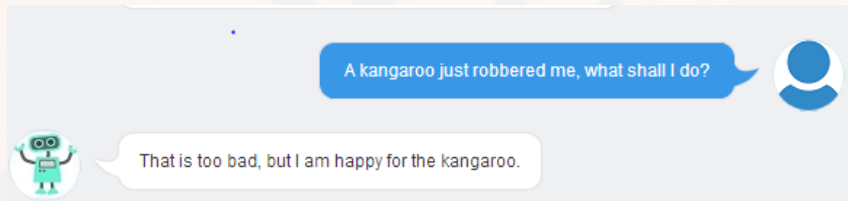


Figure 74 Answer and reply

Finally the logic of what the robot needs to say is actually bounded in the enter function:

```
function send1(text){
  var curmsg={
    text:text,
    from:"robot",
    to:"user123"
  }
  socket.emit('newmessage',curmsg);
  console.log('send:' + curmsg);
  curmsg.html = generateChatBubble(curmsg);

  $scope.msghistory.push(curmsg);
  $scope.showSugg = false;
}
```

Figure 75 Logic of response

Because the user can use two ways to send the message, the same logic is done twice in both:

```
$scope.Enter = function(event){
  if(event.keyCode == 13){
```

Figure 76 Enter function

And:

```
$scope.sendMessage = function(){
```

Figure 77 Button function

They have the same function, but respond to different calls: one respond to a click on **SEND** button the other respond to Enter on the key board.

The logic of respond is created with 3 layers, using a flag variable:

This is an animal emergency center.
if you have an emergency,
please tell me what type of animal you have encountered?
As in bird, mammal, reptile or fish.

Figure 78 First level response

After seeing these lines the user enter the first layer, where any irrelevant input will be ignore. Only input concerning the key words will be responded with appropriate answers. If user did not answer with a related answer, the chat bot will continue to loop in this phrase, constantly remind the user to answer the question.

At this stage, another function worth mentioning is the suggestion. Whenever more than 2 character is typed into the texting, a list of suggestions will show:

I found a wounded bird, what shall I do?
I found a wounded reptile, what shall I do?
I found a wounded fish, what shall I do?
I got bitten by a snake, what shall I do?
A kangaroo just robbed me, what shall I do?
hel

Figure 79 Match logic

This is done using the following code:

```
if($scope.msg.text.includes("mammal")){
```

Figure 80 Level 1 logic

The suggestions actually offers the sum of all the questions Winnie will respond to.

Once the user has given the answer for stage 1 they arrive at stage 2:

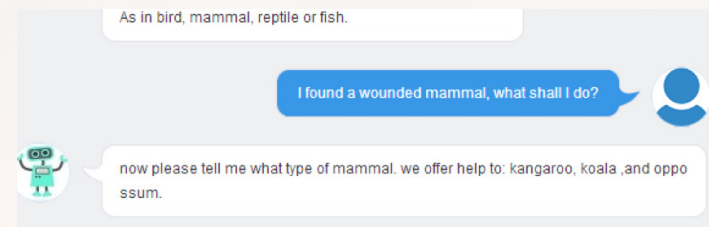


Figure 81 Level 2 effect

This could happen because the var value has been changed:

```
if(stage== '0'){  
  if($scope.msg.text.includes("mammal")){  
    send($scope.msg.text);  
    sendI("now please tell me what type of mammal. we offer help to: kangaroo, koala ,and opposum.");  
    stage= '1';  
  }  
}
```

Figure 82 Level 2 logic

The stage is just a marking variable indicates which conversation we are talking about.

This stage works similarly to the previous stage , it also only takes in the suggested input and if not will loop, only when the requirements are met, will the robot enter next stage.

In the final stage, the specific genus of the animal is given and the chat robot will give corresponding veterinary hospital to save the animal;

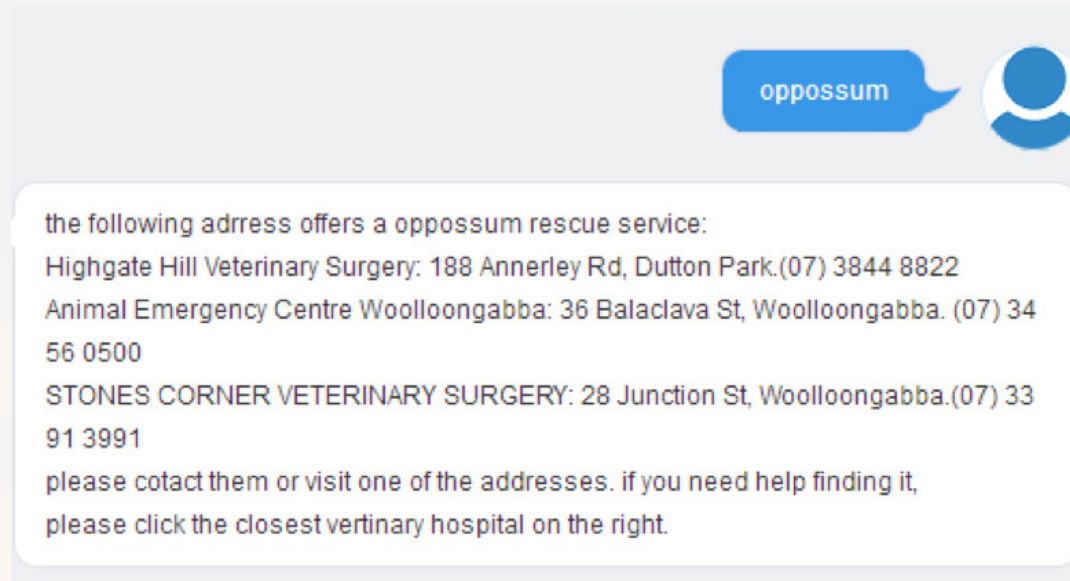


Figure 83 Level 3 output

And after then it will bounce back to stage 1:

```
");  
<br> please click the closest vertinary hospital on the right.");  
stage = '0';
```

Figure 84 Set back to level 1

This is how we implement the chat robot function.

2.5 Discuss of process

a. Aesthetic issue



Figure 85



Figure 86



Figure 87

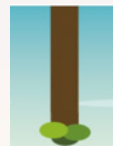


Figure 88



Figure 89

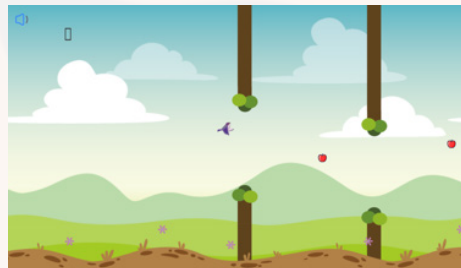


Figure 90

When we designed background and elements in game part, at the beginning, we use some pictures comes from google or other photography websites to be icons in game, whereas, it is hard to achieve a good results as it will cause the problems such as low pixel, color unmatched, flying animals size unmatched. Thus, we decided to draw all icon of game by ourselves. The results and contrast is shown above. In addition, we adjustment the color of each element, which makes game screen looks better. And this is also one part of our design. This need lots of time on making change and do users' feedback.

Tree position: 655	Main.js:115
Chracter position: 641	Main.js:116
Tree position: 652	Main.js:115
Chracter position: 641	Main.js:116
Tree position: 649	Main.js:115
Chracter position: 641	Main.js:116
Tree position: 646	Main.js:115
Chracter position: 641	Main.js:116
Tree position: 643	Main.js:115
Chracter position: 641	Main.js:116
Tree position: 640	Main.js:115
Chracter position: 641	Main.js:116
Tree position: 637	Main.js:115
Chracter position: 641	Main.js:116
Tree position: 634	Main.js:115
Chracter position: 641	Main.js:116
Tree position: 631	Main.js:115
Chracter position: 641	Main.js:116
Tree position: 628	Main.js:115
Chracter position: 641	Main.js:116
Tree position: 625	Main.js:115
Chracter position: 641	Main.js:116
Tree position: 622	Main.js:115
Chracter position: 641	Main.js:116
Tree position: 619	Main.js:115
Chracter position: 641	Main.js:116
Tree position: 616	Main.js:115

b. Game development issue

From Nook: At the beginning, I was very confused when writing code to make the length of the obstacles random. The idea I originally adopted was to implement a random length by randomly setting the length of the div using a particularly long trunk image. But doing so makes it difficult to adapt to the new page size when the page is resized. After querying other code, I found that a small segment of the trunk image repeat can be used to achieve random length.

Figure 91 Console log

Another problem is that when I calculate the player's score, the character always goes through the tree but there is no extra points. I found through the console debugging that the tree is 3 pixels per movement, so the `offsetLeft` of the tree sometimes does not equal with the character's `offsetLeft`. Then I added two other conditions to the condition, which solved the problem.

When implementing the animation function, I found that using the while loop will report an error, so re-use "if" nesting to complete the infinite loop inside the array.

We also encountered problems when implementing `characterSelector`. Initially we wanted to set the `#animalSelector` form's method to post, submit the user's selected radio value to the server, then use php to judge and change the file name in `img` to change the game character. Soon we found out that if user submit the form using submit, the page will be refreshed, and if user use `on submit = return false`, the data in the form will not be submitted to the server. When the page is refreshed, the character selection interface is reloaded, so we give up this method and use the front end to judge. Use `$('input:radio:checked').val()` in javascript to determine the character the user chooses.

When adding a bonus apple to the game, we originally hoped that the bonus apple would be 1/6. We generated a random number in the `mainRun` and judged it. After that, we created the bonus apple, but soon we found out because we using `if (bonusArr.length)` in the `mainRun`, so this part of the code will not be executed after the current bonus apple is deleted. Next we changed the way, we still use the random number to judge, when the condition is that we create a bonus apple with `display =`

"block", when the conditions are not met, we create a bonus apple with `display = "none"`. This method works fine, but it seems that due to more judgments, the spacing of the bonus apples will change, causing some bonus apples to appear in the tree, and the characters cannot eat these bonus apples. So we gave up on this method, and instead created a bonus apple behind each tree and lowered the bonus score of the bonus apple.

After we got user feedback, I made some changes to the game features. First part is to change the state of the game when the character eats the bonus apple. In this state the game will display the tips window and pause. But since I only stopped the function of the character object, the click and keydown functions in the start function still take effect, and the sound can be played. I solved this problem by modifying the code. Now the click and keydown functions will be set to null when the game is paused, and these functions will be reset when the game continues.

The second part is that the character will fly up once when clicking "close", which may cause the game to end without user's mistake. I added a count function `count` and `count++` to the close function, and the character will only fly when `count>0`. This way the character will not fly when the first click (i.e., click on the "close" button).

The third part is that the character may fly out of the interface from above, and the character does not need to avoid the tree outside the interface (because the `offtop` of the `uptree` is 0). After modification, colliding with the top of the game interface will also end the game.

c. Database issue

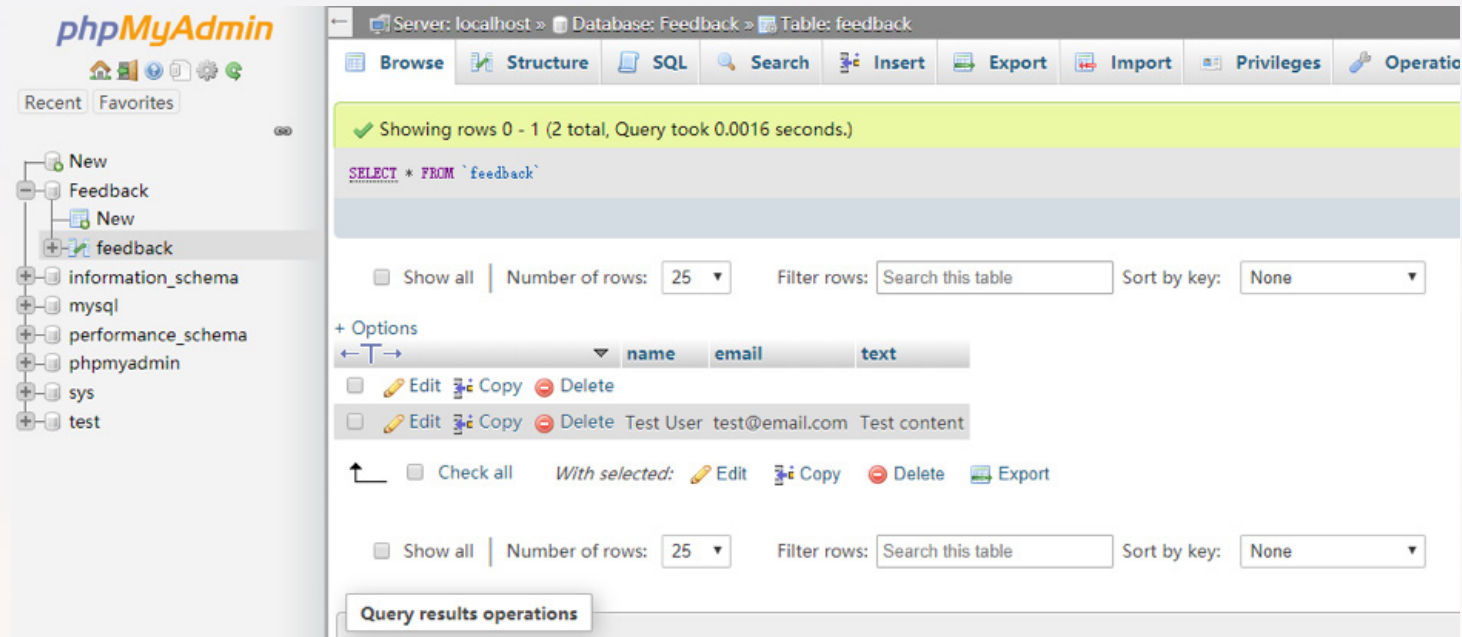


Figure 92 Test tuple

We wanted to collect feedback for the site on the “About” page, but since we were exposed to PHP for a short time, we didn’t initially think about how to store the feedback to the server. Later, we learned that by setting the form’s method to post, we can use php to get the data submitted by the form. Next we found that there is a folder in Team Zones that is phpmyadmin. We tried to log in and found that it works fine. So we decided to use phpmyadmin to create the database, and use php to get user input feedback and insert data, and now we have successfully completed it.

d. Chat robot development issue

During the implementation, I once construct the java script logic in the enter function:

```
$scope.sendMessage = function(){  
    if($scope.msg.text.includes("bitten by a snake")){  
        send($scope.msg.text);  
        send1("Apparently,You are dead, This is Australia.");  
    }  
}
```

Figure 93 problematic code

But this cause an issue, that the respond cannot relate to the previous question, if I want to make the robot to respond to a complicated stream of questions that is coherent with each, this design won't do.

I solved this problem by the following technique:

```
var stage="0";  
else if(stage=="0"){  
else if(stage=="1"){
```

This design solves the problem so that I can make the robot answer questions in layers.

3. Evaluation & Feedback

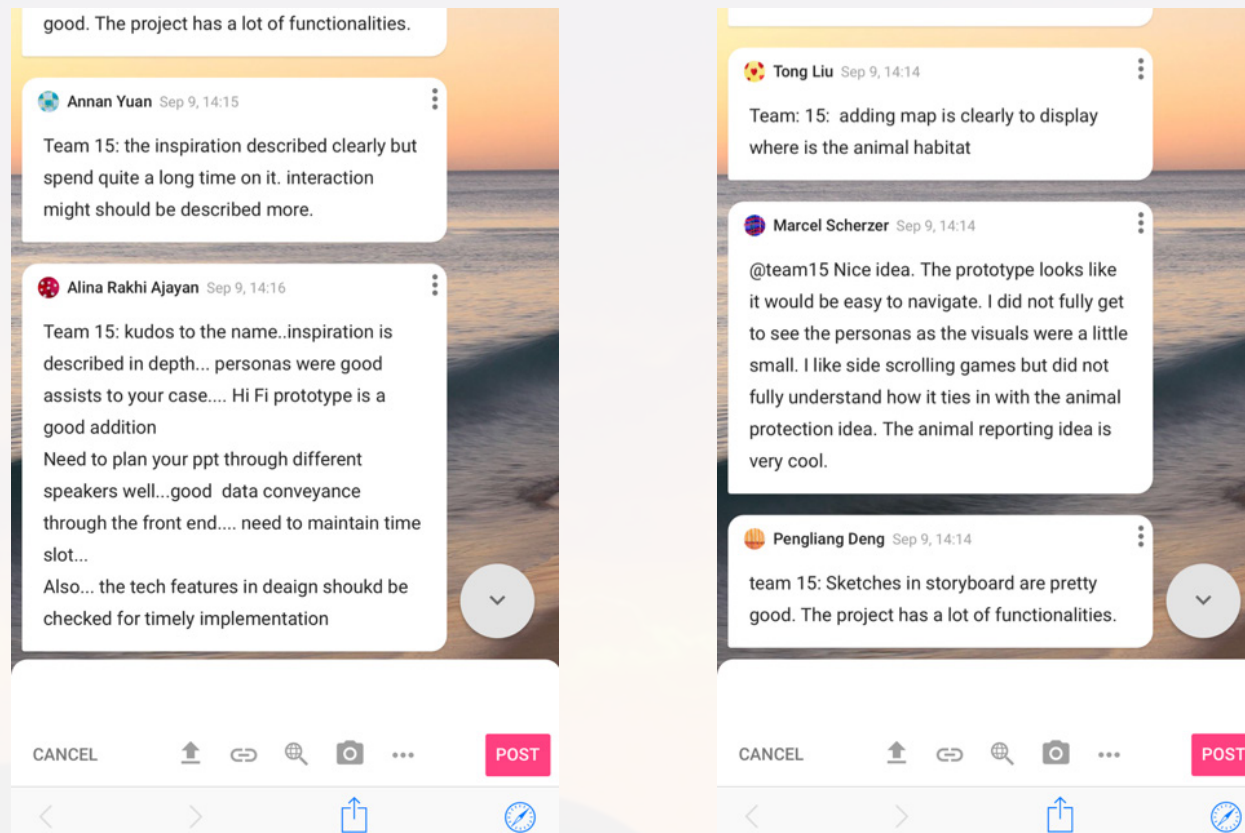


Figure 94 Slack

The images above reflection on Part A, we have got confirmed feedback on the HIFI prototype and continued to our functions development.

Reflection on Part B, we have completed the visualization by using HTML, CSS and JavaScript and part of functions demo including game, map, animal hospital API and AI contact service. In the stage of Part C, we further improved these functions and added more datasets in news and detailed information.

Then, In the testing stage, we conduct several evaluation activities. The evaluation methods we used are Think Aloud, TAM and Heuristic Evaluation. Think Aloud allows users to speak out their feeling about using our website. TAM and Heuristic are two users evaluation methods. The first one is to test the direct user and second is to specialists. Test different group of user ensure us to get comprehensive feedback from users and make a more user friendly product. Firstly, we use questionnaire method to communicate with the users and improve our products, which is important to test users experience and behaviour. Then, we interview 5 target users and transcript to record feedback. Besides, we focus on the participants who are similar with the appearance of storyboard aged from 20-40. They give the feedback more related to our topic animal protection.

Here are the feedbacks from testing and evaluation:

1. Name of the website: avenger is a killer name not suitable for animal protection
2. Purpose: create awareness address kids or young adults
3. Game interface: may suitable for all generations from children to adults

According to the feedbacks, we changed our name to animal protector and some main functions to meet the requirements of our target user in Part C. For example, we changed our image in the game, because in the first stage, the image was so cute, which are suitable for children but not for adults. We thought our target audience are mostly young adults, so we made decision to change some images and creation some animals illustrations. Besides, we keep our website colour and style as a formal information platform, which is suitable for people from all generations, not just children or teenagers.

With respect to the game, when the user eats the apple, pops up the Tips window and the game is paused, clicking and pressing the space cannot play the character, but the flight sound will still play. We fixed this problem. Then, when the user clicks "close" to close the Tips window, this click will cause the character to fly up (8 pixels), which may cause the user to collide with the tree without user's mistake. In order to improve the user experience, we quickly fixed this problem, and now the character will not fly when you click close. In addition, some users found that holding the space bar at the beginning of the game will make the character fly out of the screen. In order to fix this bug, colliding with the top of the game interface will also end the game.

4. Reflection of success

During the 10 weeks cooperation, our team have successfully completed the main function of the project including the visual organisation, digital prototype, interaction wireframe, style guide and testing. We almost finished the whole website (home, information, discover, game, about us).

Like and dislike about final outcome:

Our website is built for animal lovers, students or researchers and public generations. It has three sections, the highlight part is news and introduction of the basic knowledge of wildlife in Queensland; A gaming page tests the users' knowledge about animal protection, and by doing so, it could also train people to acquire more practical skill which they might use in case of emergency. A guiding map is also added for direction and users can connect with AI robot in chatroom. These are the highlight sections in our project, which we prefer most and try our best to build the functions.

However, I consider time limitation is the major challenge that restrict the functions of our website. However, we tried our best to contribute into the project and we should pay more attentions on the detailed information, such as the fonts, colour, background consistence and use more datasets to create PHP documents, which may reduce our pressure and create a more user friendly website.

Successful aspects:

As for design aspect, having inner connections, images are not only play a role of visual reflection of content, but also carefully selected for colour schema for aesthetics and style of web site. The main colour that website use is orange and black plus a large area of white space makes the webpage less crowded and highlight the key information for users. Moreover, in order to make the user experience and interactions more fluid and appealing, some delicate and constant dynamic elements are sited in web pages, for example, zooming pictures and other hover effects. Lastly, it is particularly designed to adapt the theme and content of every part in web pages. The design style of the website is simple and straightforward, which is benefit for users to focus on the content itself, this is because the educational part and emergency part are focused on zoology teachers and students such professionals who required for details and information of wild animals, namely content. Cartoon images are used in game part which is designed for the public, and the colour and font style are well-designed, especially the coherent background and tree-like pillars. And in discover page, the backgrounds have intrinsic continuous and concealed connection with species represented, which improves user perception of animals. For example, the background of 'Fish' is blue with bubbles, which implicates the habit of sea and river and fish life style.

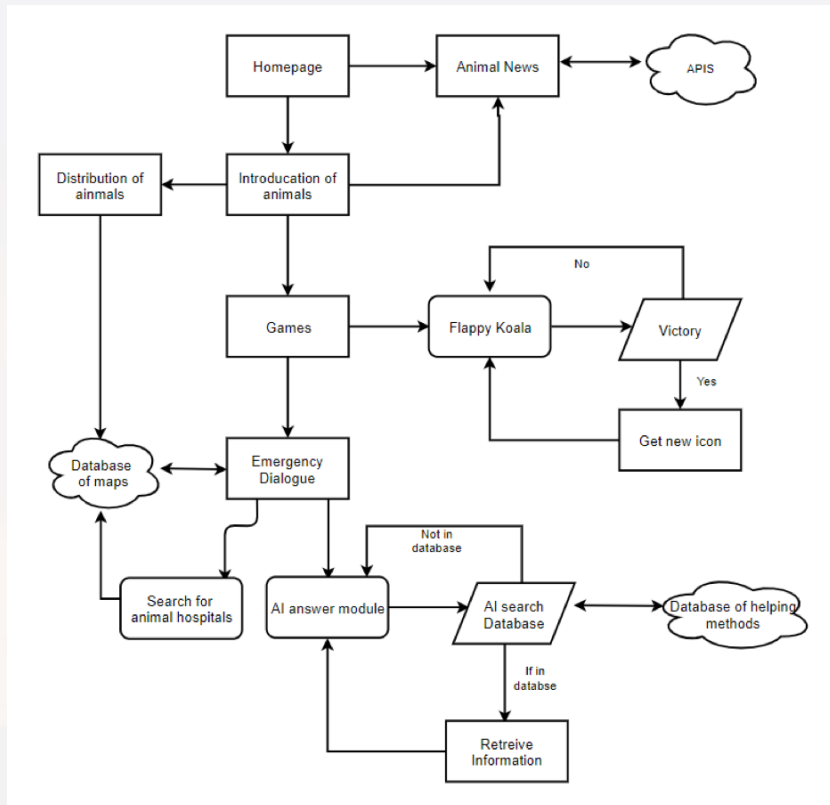
As for function aspect, the educational part is the main content, and the emergency part deepens the vertical depth of the subject while the game part is the horizontal expansion of the subject. Firstly, Chat robot, an important function of web site, is fixed on the right corner of homepage, to avoid ignorance of users successfully. Emergency maps which is made for searching animal hospitals can provide information of address and contact number for the convenience of users. Furthermore, the game named 'Super Animal' implanted educational questions to remind users of protecting animals, conditional unlocked characters are designed for attracting and extend the time users spend on our website pages. Moreover, 'Contact us' function, connecting with a SQL database, located on about page is used to collect users' suggestions quickly and efficiently, which contributes to the connection of users and us. Finally, sidebar on the discover page, also shows in a series pages for educational part, makes it convenient for users to query endangered species information faster and easier. Diverse methods for navigation on web site are supplied because different users may have different search and browse habits.

Unsuccessful aspects:

A certain number of static elements makes the webpage a little dull and lifeless, which makes the design and dynamics of the webpage are not very smooth. Simultaneously, the content in discover pages is less and insufficient so it appears a little bare and less informative, as a certain number of contents is reference links for others website. In addition, the functions still have some space to complete and improve, which is stated detailly in the following part. For example, 'contact us' function cannot filter useless information like empty name and email address. Chatting robot needs to provide more information.

We will make improvements in unsuccessful aspects we mentioned above. Firstly, more dynamic elements will be added for improving user's interactions as well as implicating users about the functions and content what they are going to look through. Secondly, webpage lacks layering so we will make some adjustments in layout and colour. Thirdly, we will enrich the content by adding brief introductions in educational part and think about

some interesting tips for every endangered wild animal in game part since text without images makes users feel bored and dull easily and it is hard to keep a balance on entertaining and learning. Moreover, some functions need to be enhanced. Chatting robot can have response with users about what should be done when meeting injured animals according to information provided by users and offer key points for users if they do not know how to rise questions. However, what we expected is far from this, we want to make a connection for chatting robot and maps in emergency part, which can make the information obtained by the user more continuous and find the corresponding treatment point conveniently. Moreover, the discover pages are full of text, and the background is simple and plain, so we will put more images on it to increase the visual attraction and make it easy to understand. Finally, for game part, we will add more unlock characters and missions to enrich the depth and luminosity of gameplay. In general, the page is still relatively monotonous and empty at the current stage, which will be the direction of our efforts.



As the task flow in design proposal picture showed, the functions what we expect for the web sites are mainly completed. However, APIs should be used in News for educational part, which is abandoned as a result of not knowing how to add APIs appropriately and not formalism. As for detailed part, we wanted to make more than one classic games, but it became only one game because of the heavy workload, instead of having two games that are not refined, it is better to make a full-featured game.

Figure 95 Design Proposal Task Flow

5. Reference

GOAHEADTOURS. (2019, September 3). 6 Unique Animals of Australia. Retrieved from <https://travelblog.goaheadtours.com/six-unique-animals-of-australia/>

Cox, L.(2018). 'A national disgrace': Australia's extinction crisis is unfolding in plain sight. Support The Guardian. Retrieved from <https://www.theguardian.com/environment/2018/feb/13/a-national-disgrace-australias-extinction-crisis-is-unfolding-in-plain-sight>

Figure 1. Retrieved from

https://www.google.com/search?rlz=1C5CHFA_enAU733AU742&biw=1440&bih=697&tbm=isch&sxsrf=ACYBGNQBp5yTHbJHf7Cpo9eVvVEEErIDO-Q%3A1571217254492&sa=1&ei=Zt-mXZ_VHe7bz7sPs_W5sAI&q=student&oq=student&gs_l=img.3..0i67I2j0j0i131j0I6.55973.56699..56876...0.0..0.226.0j6j1.....0....1..gws-wiz-img.....35i39.hXP2t4DLtVs&ved=0ahUKEwifoMayuKDIahXu7XMBHbN6DiYQ4dUD-CAc&uact=5#imgrc=ld6xnUpG-nqVfM:

Game Background Music. Retrieved from

Loop.wav <https://opengameart.org/content/questing-loop>

GameOver.wav <https://opengameart.org/content/game-over-soundold-school>

Bonus.mp3 <https://www.youtube.com/watch?v=2JjmKMP7uA>

Skill.mp3 <https://www.youtube.com/watch?v=AzEzUofreU0>

Implement Section: Persona, Storyboard, Scenario, Prototype. Retrieved from Part A Project proposal of Team C01T15

Home page, Discover page background image. Retrieved from <https://www.shutterstock.com>

Discover and News and About images.
Retrieved <https://www.pexels.com>

All Chat-Robot images.
Retrieved From <http://136pic.com/sucai/list/cd70bcba.html>

discovery.html.markerIcon1. Retrieved from
<https://img.icons8.com>

discovery.html.markerIcon.
Retrieved from <https://img.icons8.com>

discovery.html map code. Retrieved from
<https://developers.google.com/maps/documentation/javascript/places-autocomplete> and <https://developers-dot-devsite-v2-prod.appspot.com/maps/documentation/javascript/examples/full/places-autocomplete-hotelsearch?hl=zh-CN>

homepage.html.map Retrieved from <https://developers.google.com/maps/documentation/javascript/markers>