UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

MASTER THESIS

Bird Go: Mapping of Songbirds Using a Citizen Science Game

by Aixi Guo

Matriculation Number.: 3057072
Date of Submission: 4. November 2019

First Examiner: Prof. Dr.-Ing. Jürgen Ziegler
Second Examiner: Prof. Dr.-Ing. Maic Masuch

A thesis in the degree program ISE-Embedded System, submitted to the Department of Computer Science and Applied Cognitive Science at the University of Duisburg-Essen to obtain the academic degree Master of Science (M. Sc.).

Declaration in Lieu of Oath

I hereby declare, under oath, that this master thesis has been my independent work and has not been aided with any prohibited means. I declare, to the best of my knowledge and belief, that all passages taken from published and unpublished sources or documents have been reproduced whether as original, slightly changed or in thought, have been mentioned as such at the corresponding places of the thesis, by citation, where the extent of the original quotes is indicated.

Duisburg, 4. November 2019, Aixi Guo

Abstract

The study of natural systems, such as the global distribution of birds, often needs a large amount of data. Collecting these data typically needs to take a lot of time and resources, especially in the project of songbird mapping. Birds are widespread creatures, and they live in almost every type of environment. However, because they have a large population and some of them need season-related migration, a long-term data collection is required over a wide area. Relying solely on bird scientists to collect these data is not enough. Some existing projects encourage citizen to help scientist with the birds finding. The people who help with scientists in the citizen science project are called citizen scientists, and they play an essential role in bird searching, bird observation and bird songs collection. However, most tools that support capturing of bird observations are cumbersome to use and do not put user interests and motivation into consideration. In this thesis, I present a citizen science game: Bird Go. The primary purpose of Bird Go is to encourage citizens to go out and record birdsongs in the nearby environment. An important requirement of this game is to combine the entertainment with the scientific task. Whenever players record a bird song, they will receives rewards which altogether establish a wholesome game experience. That means that making a little scientific contribution can bring a lot of fun to the player. At the same time, the species name of a bird is extracted from recordings through machine learning in the game. For this purpose, a convolution neural network (CNN) is built in this thesis, which has the ability to recognize species of birds from birdsong. The bird's data, which includes the species name and the recording's information (such as location and time), was sent to the server to generate a map of wild songbirds. In the end, in order to assess the ability of Bird Go to motivate players in mapping songbirds, a user study was conducted, leading to promising results of game experience and output produced. The user study included ten volunteers who played Bird Go in a week. A survey that focuses on players' experiences was done at the end of that week. The survey shows that Bird Go has the ability to encourage players to go to the places where birds live. And the reward mechanism of Bird Go is one of the main motivations of players. Players also receive a lot of fun during the birds finding. Some of them have the desire to continue to contribute to bird research and want to play Bird Go for a long time in the future.

Key Terms: Birdsong recognition, Citizen Science, Game with a Purpose, CNN

Contents

1 Introduction

In recent years, there are several areas of scientific work (especially in ecological research) where citizens help with. Most of the participants are non-scientific volunteers. They usually have a great interest in the scientific topics which they want to help. Although volunteers don't write papers or process data, they are critical to collecting information in the research field (Gura, 2013).

In this thesis, I focus on a citizen science project about the mapping of songbirds. Birds are easy to observe conspicuous creatures. They are found in all terrestrial and most aquatic environments (*Ornithology*, 2006). They also have the longest distance migration in any organism (Able, 1999). And they are environmental indicators that indicate ecosystem health (Sullivan et al., 2009).

For the public, birds are friendly, cute and often encountered creatures. Some of the citizens treat observing birds as a passion. Amateur ornithologists also study birds for a long time (Barrow, 2000). In this thesis, I have designed a citizen science project using games with a purpose. This game prototype can be used to encourage volunteers to help scientists collect bird distribution information in the environment.

1.1 Description of the Problem

The main participants in the citizen science project are citizen scientists. They are volunteers who participate in scientific research in the form of assistants. They are not real scientists, nor are they paid because of their assistants. Some studies have found that participants enjoy being citizen scientists because they want to contribute to research and regard research goals as highly valued (Holohan & Garg, 2005; Krebs, 2010).

Another study (Curtis, 2015) also shows that interest is the motivation of the participants. Some volunteers are willing to spend several hours a day on volunteer work. To achieve the long-term success of the project, it is necessary to first know how to attract and keep participants (Reed et al., 2013; Rotman et al., 2012).

There are many projects that use different methods to motivate volunteers. One of the most efficient technology is using gamification and 'Games with a Purpose' (Von Ahn, 2006). Each year, people around the world spend a lot of time playing games. If scientific research and the collection of scientific data are embedded in a game, people can solve large-scale problems when they play the game.

In this thesis, I focus on one citizen science project which generates bird cartography using games with a purpose. Although the gamification and games with a purpose have shown potential for helping to generate motivation in citizen science, still no combination of bird cartography and gamification exists so far.

Recognition of the bird's species from players' recordings is another problem while designing this game. There are nearly 10,000 species of birds in the world. And there may be several birds singing at the same time in the player's recording. Extraction of the birds' songs from the environment's noises and recognition of birds' species normally need huge computation. Implementing an algorithm suitable for bird recognition in the game is a challenge and an important part of the project.

Another problem is how to encourage players to go to special places where birds live and do the recording of birdsong. For example, the game should have mechanisms to give players rewards when they are doing correctly. Combining entertainment and scientific research needs to be considered in the process of designing the game.

1.2 Solution

To overcome these challenges, several mechanisms that encourage players to go out and collect birds' songs are used in game design. The game's idea is similar to the augmented reality mobile game Pokémon Go, which showed great potential in successfully motivate players to go outside and play the game (Althoff et al., 2016). In Pokémon Go, players need to go out of the house, walk or wander in the surrounding environment to catch the pokémon. To enter the store or participate in some important game events, players also need to go to a special location marked on the map.

The game in this thesis also requires to encourage players to go out and walk to find birds. In Bird Go, a 3D virtual world is first built based on the real map. A character model representing the player is located at the same place as the player's real position. When a player enters a forest or park, where a lot of trees exists, he/she can find treasures chests in the 3D virtual world. Players also have a high possibility of contact with birds in these places. Through this machinism, the game encourages players to go to places where birds live. When players hear the birdsong, they can use the recording function in the game to record birdsong. In return, the game will give players rewards after bird species are successfully recognized form the birdsong by the game. These rewards are also the key elements which altogether establish a wholesome game experience.

To recognize birds' species automatically from players' recording, An algorithm model is required, which can analyze and learn from exists bird's song recordings. With the help of the recent improvement of the field of Deep Learning has made this work possible (Kahl et al., 2018). Therefore, I built up my own neural network module that is especially for bird's song recognition and suitable for the mobile phone. This model is used to predict bird species based on player records. And the player can get rewards related to accuracy. This means that the higher the accuracy of the prediction, the more rewards the player can receive. This can encourage the player to record better audio with less noise or to go close to the bird's voice source and record clearer birdsong.

In order to analyze the distribution of birds, the recognition results are automatically sent to the server. The server records each recording's information, including bird species name, latitude, longitude, probability, and recording time. A server management web site is also developed, which shows the collected data and has a map to display the birds' distribution according to these data.

1.3 Contribution of Work to the State of Science

In terms of bird recognition, this thesis compares several recognition algorithms and designs a recognition prototype that can be applied to mobile clients. This model has a certain extent against noise, and it is also optimized for the mobile client. For a large number of training classes, it still maintains a relatively high accuracy of prediction. People can implement a better model based on this prototype. This model can also be used to identify bird sounds in other areas by changing the training data set.

On the other hand, this thesis provides a prototype that combines bird cartography with a citizen game. Also, several of motivation strategies have been applied to attract the interest of volunteers. Finally, a user study of 10 volunteers in a week showed that the method used by this thesis can successfully encourage players to find birdsong. And a map of songbirds in Duisburg was generated, which was related to collection of these 10 participants in a week.

1.4 Construction of the Work

Chapter 1 introduces the basic background of citizen science and also describes the problem studied in this thesis. It can give a basic outline of the whole thesis.

The second chapter shows the status of research, including algorithms for birdsong recognition and researches in citizen science.

The third chapter mainly talks about the method used in this paper. This chapter is divided into three parts.

Chapter 3.1 gives the basic concept of my solution and the basic steps of implementation. Chapter 3.2 mainly introduces implement of CNN birdsong recognition and used technology and key parameters. Chapter 3.3 mainly talks about game design and the basic implementation of game elements.

Chapter 4 is a simple evaluation. In this chapter, ten players were inviteted to play Bird Go in a week and the system was evaluated based on feedback from the players.

The fifth chapter gives simple conclusion and further work.

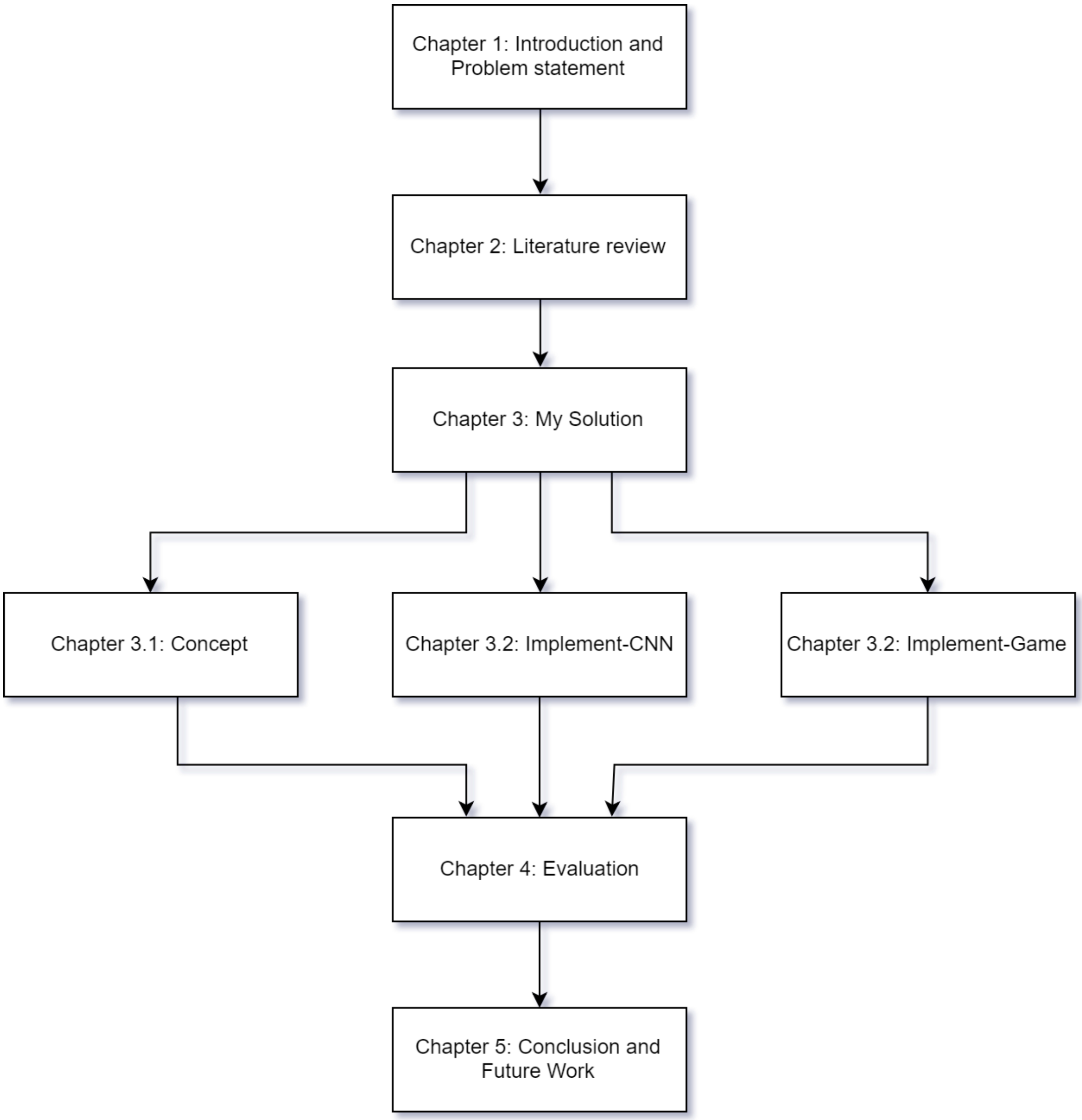

Figure 1.1: Thesis Approach

2 State of Research

2.1 Citizen science

The term "Crowdsourcing" is arising in recent years. The word crowdsourcing is a mixture contraction of Crowd and Outsourcing (Schenk & Guittard, 2009). It refers to information systems that divide works or tasks among many people. Crowdsourcing systems have a wide of applications in commercial, educational, and scientific tasks. In this topic, I am focusing on "Citizen science" which is one kind of "crowdsourcing" project. It encourages the general public to take participate in scientific inquiry (Cohn, 2008)

Over the past decade, the Internet and smartphone have greatly expanded all over the world. They increased the efficiency of most scientific research. Thereby citizen science projects have also been rapidly enhanced (Gura, 2013). At the same time, thanks to the development of information technologies, collecting and managing data are much easier than before for scientists. Scientists can also easier manage projects, communicate with volunteers, encourage new people, and propagate research findings more broadly (Newman et al., 2011).

One of the citizen science projects is the bird's information collection. Birds are easy to observe and live in most environments creatures. They play an important role for the environment and is the indicator for the ecosystem health (Sullivan et al., 2009). However, collecting bird information requires a lot of time and human resources. Because they have a large population and some of them need season-related migration. Aims to the collection problem. This thesis proposes a solution that encourages citizens to participate in bird information collection and help scientists for bird analysis and studies.

There are already some existing citizen science projects for bird's information collection. For example, *eBird* is one of these projects. It uses the Internet and information technology to provide citizens with a convenient way to contribute their observations of birds for science and bird protection (Sullivan et al., 2009). *eBird* has built an online community for bird watchers. It supplies a globally accessible database to store and share observers' discoveries. According to the collected data, *eBird* also provides information on species distribution, migration, and various spatial and temporal scales. In addition, through the process of informal science education, *eBird* users can expand their bird knowledge when they use visual tools to explore bird data and interact with experts.

However, *eBird* does not actively encourage users to take more interest in this science activity. The motivation of *eBird* is users' own desire to find and identify birds, as well as recognition and praise for their discoveries. It is hard to motivate users who are not interested in observing birds.

2.1.1 Games with a Purpose

Although citizen science projects show great potential to encourage thousands of citizens to participate in scientific inquiry, it is also necessary to find a better way to attract more citizens' interest. Aiming to improve the acceptability of citizen science for the general public. Some of the citizen science projects are using the computer game format to encourage people (Von Ahn, 2006). In this form, scientific activities are embedded in computer games. The game is also known as "Games with a Purpose". Different from other citizen sciences, projects designed with Games with a Purpose reduce the cost of collecting data and increase the level of participation.

There are many games with a purpose which are using points, scores, and achievements techniques to motivate players (Bogost, 2011). In contrast, entertainment games show more of the game's story, beautiful game visuals and aural experience. Obviously, these games are more attractive to the player's attention and interest.

A study by Prestopnik & Tang (2015) compared two kinds of games with a purpose. One is a story-based game, Forgotten Island and another is a points-based game, Happy Match. Both games focus on the life sciences. Researchers in life sciences regularly collect photos of creatures. These photos are often labelled with time and location data. However, they are only valuable when the subject of a photo is known. This information is usually missing when a photo is taken.

Both of the games mentioned above allow players to help classify photos of plants, animals and insects. They both provide sample photos that are related to the classification. Players can zoom in or out of the photo. In addition to the photos, the two games also provide help text to explain how to make the right choice. In the "Happy Game", the player classifies ten photos in one round. And after all of the questions have been answered, a score representing the performance of the player is displayed. The score is measured based on a set of classifications provided by the expert. In the game Forgotten Island, photos of creatures are distributed in various places. Players need to recover these photos and rebuild the archive by processing classification. In this game, the classification activity is motivated by the player's fantasy and curiosity. Besides, players are rewarded with in-game elements (such as currency) instead of a score. Players can use these elements to interact with the other game elements.

Happy match is a quiz game like most of current citizen science games. The scientific tasks are on the foreground of the game. Players are motivated by getting a high score after classification activity. In contrast, Forgotten Island is a story-based game that can motivate the players who don't have enough science enthusiast motivations (Garris et al., 2002). Scientific missions have been embedded in the larger complex world of games and are waiting for players to explore. And it also offers a variety of strategies to bring fun to the player, such as humorous characters, various puzzles and so on. These features are designed to appeal to players who are not interested in scientific tasks. By doing a little science, they can earn hours of entertainment time.

This research shows that story-based games are better to recruit and retain participants in citizen science. And they have the ability to engage the players who are conflicted about scientific inquiry. On the other hand, the research (Von Ahn, 2006) shows that games with a purpose also has the ability to engage people into 'human computation'. It also shows humans, as they play, can solve problems that computers can not yet solve.

Another important aspect is that the game should not be too complicated for players since players are non-professional people. For example, Foldit is an online puzzle video game. It helps to predict protein structures (Cooper et al., 2010). Compared to other games, Foldit is a relatively complex game, and players need to complete a series of tutorials (32 in total) in order to get basic information about the structure and function of the game (Andersen et al., 2012). Thus participants will lose their patience and interest during learning how to play.

A study of Foldit (Curtis, 2015) also shows survey responses of players. Over 60% of respondents said that they play Foldit because they want to contribute to science. And over a third of players interested in the background of the science. Only a small number of players(< 4%) were motivated because it is a game that can be fun. In this case, the players are only motivated by himself. It is difficult for the game to expand its players to more people who don't have an interest in the protein structure prediction.

Curtis (2015) also mentioned that the motivation of players could be divided into internal factors and external factors. For intrinsic motivation, a study (Ryan & Deci, 2000) indicates that it exists when the user feels enjoyable of an activity or feels fulfilment and competence during the process of participation. Internal factors include entertainment, visual appeal, interest in science, learning opportunities, contributing to scientific research, and making friends. Foldit participants receive enjoyment from co-operating with others, and they have extraordinary interests in scientific tasks. Players' motivation is mainly from internal factors. Therefore, there is no necessary to use external factors to motivate participants in Foldit. Another internal motivation is that participants have the desire to collaborate and collaborate with others as part of a social group (Alexander Hars, 2002).

For external factors, there are such as important scientific goals, rankings/scores/scores/rewards in the game and positive feedback from scientists. They are in operation after an activity is done or one task is completed. Extrinsic motivation includes not only the rewards and punishments, but it also exhibits that the activity is personally endorsed in some way (Ryan & Deci, 2009). It bases on personal engagement, such as a desire for high rank or more points in the game, or others' approval, for example, the participants are named by the co-author on paper. There are also some other external factors that influence participants. For example, altruistic motivation or empathy occurs more often in citizen science projects. Participants have the desire to make a contribution to scientists or scientific tasks. And this is a more general "community-based" voluntary behaviour (Batson et al., 2002; Clary et al., 1998).

In a word, to expand participants, it is necessary to consider both internal and external factors in game development. For those players who don't have much knowledge or are not interested in scientific tasks, external factor seems to be more critical.

2.2 Birdsong Recognition

In order to automatically recognize bird songs, Machine Learning has shown great potential (Koumura & Okanoya, 2016). Although automatic speech recognition using Machine Learning has been extensively studied, they cannot be blindly used for other biological purposes. Koumura & Okanoya (2016) has proposed an automated system, which is suitable for recognizing the content of birdsong. It focuses on the three attributes of bird songs. The first is a sound element, which like a single word in the human language. The second attribute is the temporal structure of the sound element. And the last one is the probability of the sound elements. To recognize the birdsong from these three attributes, it used a hybrid model of a deep convolutional neural network (DCNN) and an HMM (a hidden Markov model by Kogan & Margoliash (1998)). This method has also been used to achieve high performance in human speech recognition (Hinton et al., 2012).

Another method is the study conducted by Kahl et al. (2018), which shows an effective method for identifying birds species. In this study, it first extracted the spectrogram from the recording of the bird. A spectrogram is an image that contains all the frequency and temporal characteristics of the audio. And each bird has its own special spectrogram. The spectrogram of a bird is different from another bird. So it can be used to distinguish birds. Therefore, the audio classification problem can be treated as an image recognition task. If the spectrogram image of birdsong can be recognized by Machine Learning, then it can identify the corresponding bird species.

In details, Kahl et al. (2018) trains a deep convolutional neural network(DCNN) based on the classical convolutional neural network (CNN) architecture, which has seven weighting layers with no bottlenecks or shortcuts. The model was then trained on 521,873 bird sound spectrograms. At the same time, a cosine dynamic learning rate schedule mentioned in Huang et al. (n.d.) is used and starting at 0.001. Also the ADAM optimizer (Kingma & Ba, 2014) is applied. Finally, The trained DCNN can be used to predict the birds' species in the recordings.

They also mentioned some improvements solutions, such as DenseNet (Huang, Liu, van der Maaten, & Weinberger, 2017) or WideResNet (Zagoruyko & Komodakis, 2016a), which can improve the accuracy of the prediction. They all improve the accuracy of the classic DCNN structure. However, some of them require huge computation and long-term training with a powerful computer or a supercomputer. Also, the number of parameters for different methods is different. The more parameters, the more storage space are needed when the model is saved. There is a comparison of different methods supplied by Keras[1]. The comparison is between the error rate and model size of the 2012 ILSVRC ImageNet verification set (Russakovsky et al., 2015). Table 2.1 shows the parts of the results. For *Top-5*, the model predicts five same categories at the same time. When any of the predictions are correct, the results are correct. When five times are all wrong, the prediction error is calculated. The classification error rate is called the top-5 error rate. *Top-1* means for classification error rate for only one category predictions.

[1]https://github.com/keras-team/keras-applications

Methods	Top-1	Top-5	Size
VGG16	28.732	9.950	138.4M
VGG19	28.744	10.012	143.7M
ResNet50	25.072	7.940	25.6M
ResNet101	23.580	7.214	44.7M
ResNet50V2	24.040	6.966	25.6M
ResNet101V2	22.766	6.184	44.7M
ResNeXt50	22.260	6.190	25.1M
InceptionV3	22.102	6.280	23.9M
Xception	20.994	5.548	22.9M
MobileNet(alpha=0.25)	48.418	24.208	0.5M
MobileNetV2(alpha=1.4)	24.770	7.578	6.2M
DenseNet121	25.028	7.742	8.1M
NASNetLarge	17.502	3.996	93.5M

Table 2.1: Comparison of different methods

Due to limitations in equipment and experimental environments, ResNet(He et al., 2016a) is mainly used in this thesis. In the traditional CNN structure, one layer represents one function of data processing. For example, a convolutional layer is used to extract features of an image. The pooling layer is used to reduce the dimensionality of the data. Usually, one layer can be connected to another layer to complete the full prediction function. If we want to add more function to the model, we can add some new layers. One of the most common examples is that if we want to improve the accuracy of the prediction, we can add more convolutional layers so that the model can consider more image details and make better predictions.

However, in fact, He et al. (2016a) found that when the model layer is increased above a certain number, the accuracy cannot be improved. So they proposed a short-connection method to improve this. Short connections make the signal easier to pass through the network, so that it can reduce gradient explosions and gradients disappearing problems. With the help of short-connection, the model's accuracy can be improved again when the number of layers is increased. Compare with tradition CNN model, which has 16 or 19 layers, ResNet can even achieve 101 layers or more layers.

There are also some other models which have a similar short-connection design. For example, Yolo (Redmon et al., 2016) is a real-time object detection CNN algorithm. It uses its own network called Darknet (Redmon, 2013–2016) for prediction. It not only predicts the classification of objects but also calculates the boundaries of objects in the image.

In general, although there has been a lot of researches on bird recognition and citizen science using games with a purpose, it is still a vacancy to combine them. In this thesis, birdsong recognition will be applied to citizen science using games with a purpose.

3 Bird Go: Concept and Implementation

This thesis implements a bird song collection game. The game uses several methods to encourage players to leave home to places where birds usually exist. The game is implemented on a mobile platform and players can play it in forests, gardens or parks. When the player hears the surrounding birdsong, he can use the in-game recording function to record the birdsong. Bird species present in the birdsong are automatically recognized by machine learning in the game.

The birdsong recognition algorithm is implemented using Convolutional Neural Network (ConvNets or CNN). CNN is one of the deep neural network algorithms. It is mostly used in image recognition, object detection. Here the CNN is used to recognize the species of birds from recordings of birdsongs. Different from the common CNN model, which is run on the PC with strong computing power, this model is optimized for the mobile platform which has limited power and lower computing ability.

After recognition, players can get rewards which are the key elements of the game. Prestopnik & Tang (2015) mention that story-based games are better to recruit and retain participants in citizen science. Therefore, exploring the game story of Bird Go is the most central part of the game. It will bring fun to the players and attract them to repeat the action of the birds finding.

At the same time, the identification information will be sent to the server, which will collect information about the birds (location, time and species name, etc.). These data can help with bird research.

3.1 Concept

This section is divided into 2 parts (Figure 3.1). The first part is a description of the identification of bird species from birdsong, including the CNN algorithms and workflows. The second part is about game design, including game logic and game elements. In addition, this chapter also shows the combination of these two parts.

3.1.1 Birdsong Recognition

The main step of birdsong recognition part is described in Figure 3.2. There are two main phases. First is the data preparation phase. Second is the CNN training phase.

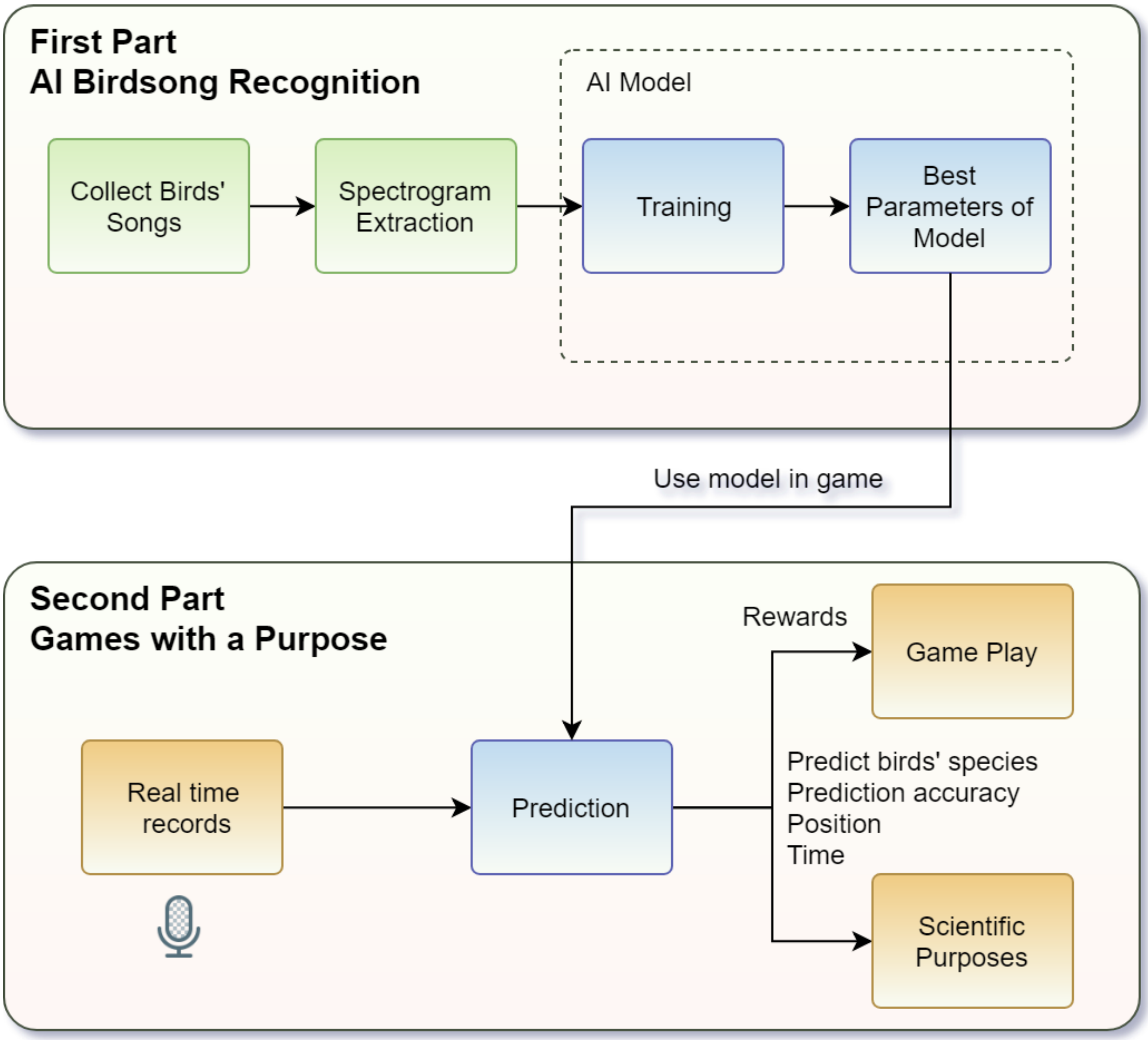

Figure 3.1: 2 parts of the system. Top: Part 1, CNN training workflow; Bottom: Part 2, the game uses the CNN model to predict from the player's recordings. Players will be rewarded based on the accuracy of the prediction. Bird information will be used for scientific purposes.

1. Data Preparation Phase

For recognizing the bird species, the CNN model needs to be trained. Therefore it is necessary to collect data-set of birdsong. The data-set includes bird songs audio files and information about the corresponding bird, like the species names, the location and possibilities. These data are retrieved from a public data-set that was collected by the aforementioned eBird application (Sullivan et al., 2009) and can be accessed under `https://ebird.org/explore`.

The bird song's audio files are recorded by different people in different places. And their length and quality are also different. Therefore, it is necessary to first divide them into the same format and the same length before training. And put them in a hierarchy according to their species name.

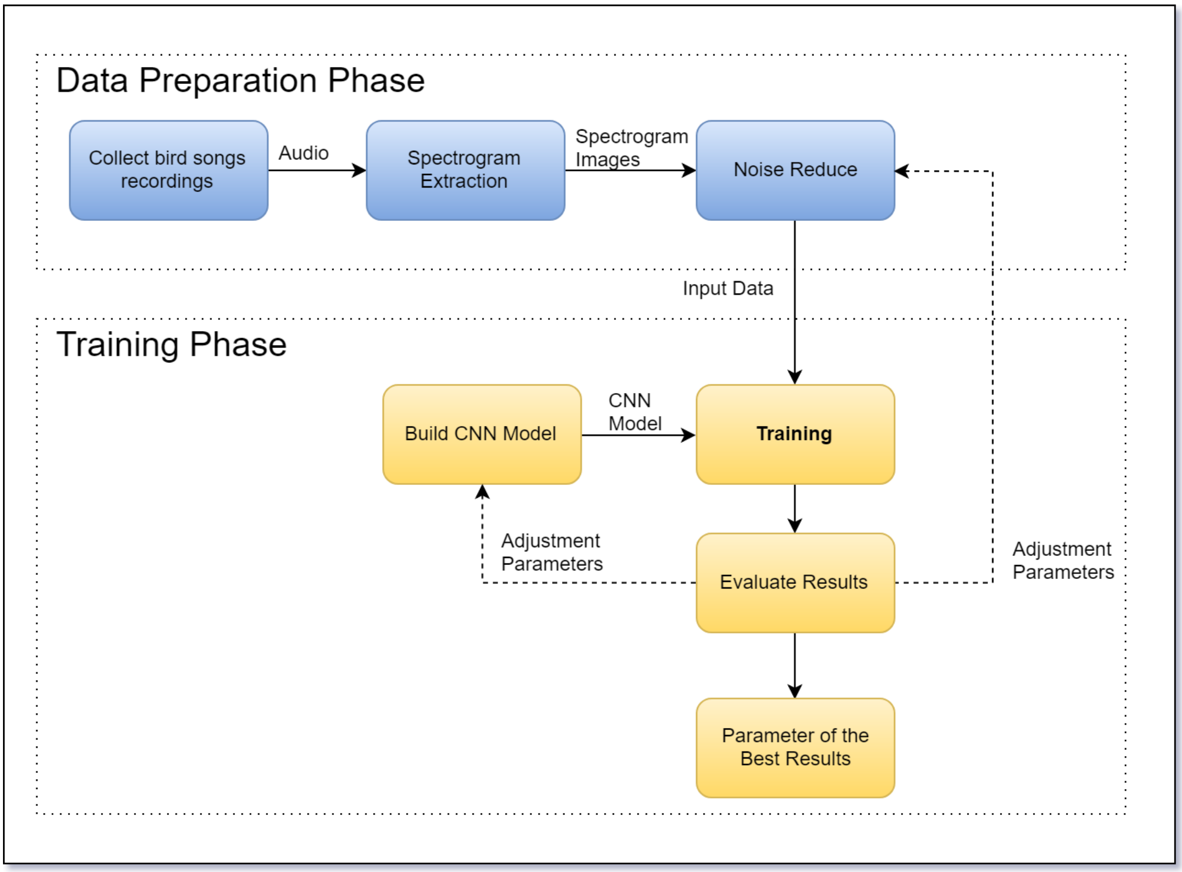

Figure 3.2: Top: During the data preparation phase, the collected bird sounds will be converted to spectrogram and process noise reduction. Bottom: training phase, training and optimizing CNN model at this phase

The next step is to generate a spectrogram of each song. In this step, a spectrogram generation algorithm (Choi et al., 2017) is implemented using CNN. Unlike the general spectrogram extraction algorithm, this method trains a special CNN model with audio as input. Its output is a spectrogram image of the input audio. There are two reasons for using this method. First, It uses GPU to accelerate spectrogram extraction. The method is a real-time on-GPU algorithm with higher speed and better compatibility than usual CPU-implemented spectrogram extraction. Second, because the trained model can be shared on the game side, this spectrogram generation algorithm model can be imported into the game design without any additional modifications. Therefore in the game, the player's recordings will be converted to spectrogram using the same algorithms and parameters, which are also used in training.

After the spectrogram is extracted, an image of the spectrogram is generated to represent the sound. This image can be used as an identification for a special bird. It includes the features of this category of bird. The bird recognition problem is then converted to an image recognition problem. On the other hand, there are many mature technologies that use CNN for image recognition and are rarely used directly for audio recognition. Typically, an audio file contains a waveform that contains only two-dimensional information about the sound (time and amplitude). Spectrogram extraction extends the information to three dimensions (time, frequency, and intensity) by Fourier transform of the waveform. Therefore, the spectrogram image has more sound characteristics than the waveform, and it has higher prediction accuracy when it is used as the input of the CNN.

Before starting to recognize, the noise of sound needs to be reduced. However, since the birdsong recordings do not have the same noise level, it is difficult to find a fixed noise threshold for noise reduction. Therefore, a random noise threshold is used to reduce noise. Using a random threshold allows the model to have the ability to identify birds under different noise conditions.

2. CNN Training Phase

At this phase, the recognition model is the core part. It receives the input of the spectrogram image and will give a bird label as the output.

There are many recognition algorithms available. Kahl et al. (2018) gave a CNN prototype of birdsong recognition. They have also mentioned that this prototype can be improved by DenseNet(Huang, Liu, Maaten, & Weinberger, 2017) or by WideResNet (Zagoruyko & Komodakis, 2016b). In this thesis, a modified Resnet (He et al., 2016b) is used as the main algorithm. The main modification is to minimize the size of the model parameters so that the model can run on mobile devices with relatively high accuracy.

Once the training results are obtained, it will be used to predict the test data-set. The test dataset is new for the model, and CNN has not learned from it before. According to the results, the parameters of the noise reduction part and the model part were modified and optimized to obtain better results.

Finally, the parameters of the best CNN results are saved as a bytes-file. This best CNN result has a minimum size and a relatively high prediction accuracy.

3.1.2 Citizen Game

In this part, the citizen science game Bird Go will be introduced. In order to collect the available game elements, the game elements in Pokémon Go are analyzed. The augmented reality mobile game Pokémon Go is very successful in encouraging players to do outdoor activities. It also has the potential to motivate players to go to special places. These features are similar to Bird Go, which encourages players to go to the forest or park to find birds. Therefore, Bird Go's design can take example by the game elements of Pokémon GO.

In order to collect the available game elements, the game elements in Pokémon Go are analyzed. There is a list (Table 3.1) of comparison for game elements between Bird Go and Pokémon Go.

Not all of the elements in Pokemon Go are implemented in Bird Go. Because not all of them are necessary for Bird Go's purpose. Some of them may even lead users to distract from the

Elements	Pokémon Go	Bird Go
Map	A 3D Map	3D world
Hot Region	Flag on Map	Flag of Treasure Chest
Collections Showcase	Pokédex	Collections
Experience	Yes	-
Egg	Egg Incubator/Lucky Egg	Bird Nest/Bird Egg
Growth Mechanism	Power Up/Evolve	Growth Value
Interaction with Virtual Creatures	Fight	Feed/Place Toy for Bird Pet
Achievement	Yes	Yes
Items	Yes	Yes
High Scores	Yes	-
Daily Rewards	Yes	-
Virtual Cash	Yes	-

Table 3.1: Comparison for Game Elements between Bird Go and Pokémon Go

original purpose. To choose the suitable elements for Bird Go, it is necessary first to define the requirements of the Bird Go.

Bird Go has four main requirements. All game elements and designs should surround and support these four goals. Figure 3.3 describes the relationship between these goals and game designs.

Record Bird Song

The first requirement is to record the song of the bird. The game should have the microphone function enabled and allow the players to record the birdsong around them.

In order to have a good user experience, several visualizations of sound are implemented. Figure 3.4 shows a screenshot that the user is recording the birdsong. The centre of the image is the 3D visualization of the spectrogram. The colour of each point will gradually increase from blue to red related to the value of the spectrogram. There is also a white circular spectrogram around the left pause button. Its shape will vary depending on the volume of the sound. In addition, the cartoon bird on the upper left will fly up and down when there is sound. There is also animation around the pause button. The player can see it when he starts recording the sound. Below the button is a progress bar that represents the recording time. The current maximum recording time is 15 seconds. With these visualizations, the player can see the sound changes directly from the graph. Different birds have different sounds, so players can see a variety of visual images.

Recognize Species of Bird

The next requirement is to identify birds based on the user's recording. In this section, the trained models are imported into the game, and they are used to do the spectrogram extraction and prediction. The prediction results will be shown for the players as Figure 3.5

Since the input of the CNN model is a spectrogram image having a certain height and width, the width of the spectrogram image is proportional to the length of the audio. Therefore, it

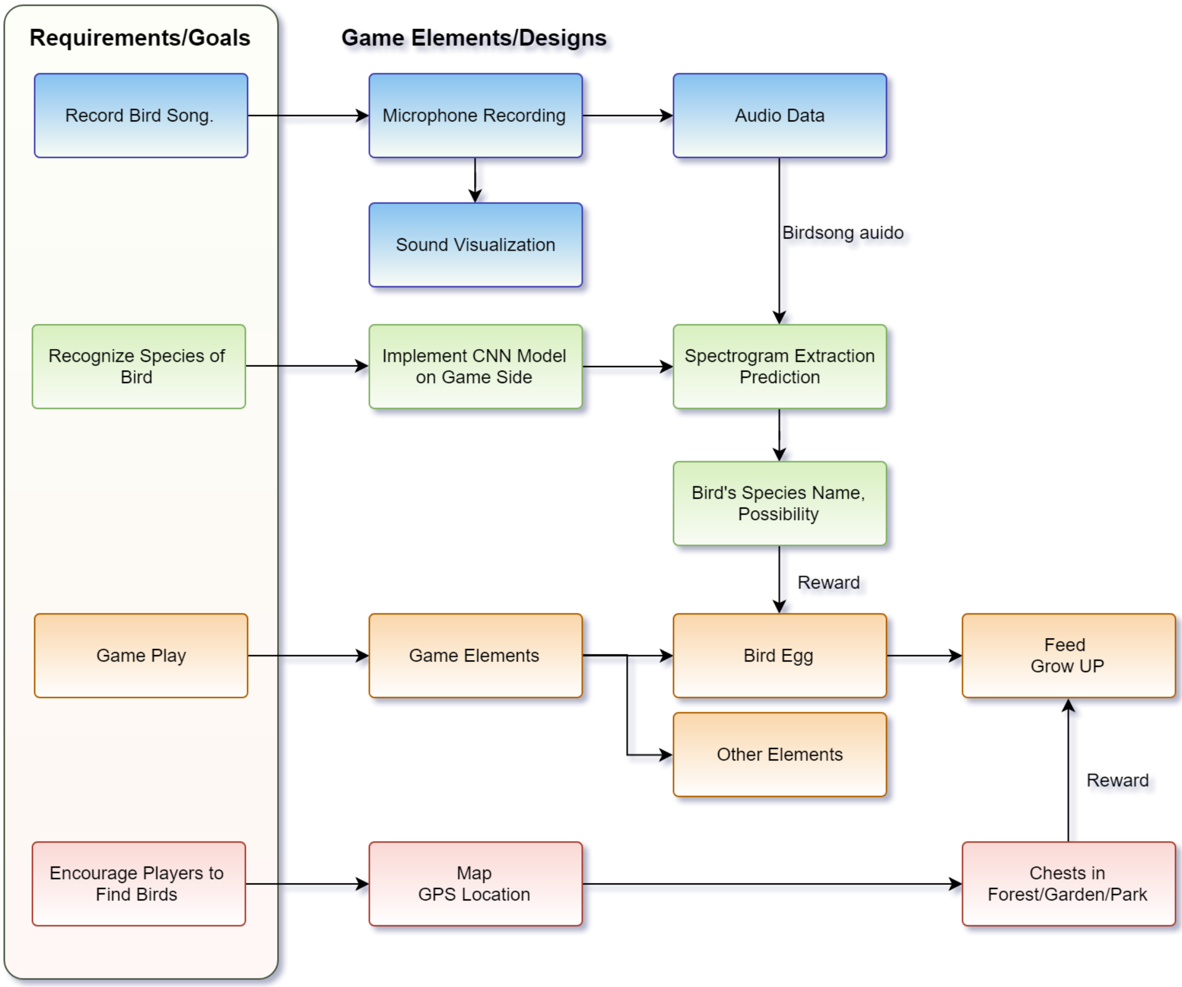

Figure 3.3: Game Requirements(left) and related game designs(right)

can only represent a certain length of audio. For recordings of bird songs, they usually have different lengths and have different spectrogram image sizes. Therefore, it is necessary to split the spectrogram image of recording into small time slots with the same size as the CNN model input. Recordings with less than one slot length and slots that do not contain bird songs will be ignored.

In Figure 3.5, the spectrogram is divided into several time slots. Each slot represents one second of sound. If there is a bird call in the slot, a label will appear at the top of this slot. The bird's name and prediction accuracy level (0-9) are displayed on the label. The prediction accuracy level (0-9) indicates the probability of prediction of each slot from the output of CNN. For example, level 9 indicates the prediction accuracy is big than 90%. Level 8 means 80% - 90% probability.

Because the CNN model will provide output for any input, this means that noise or non-bird sound will also get the most similar bird probability. So a threshold (75% is used in this thesis, and most predictions for non-bird sound have probability lower than this value) is set

Figure 3.4: Record and Recognition UI. The centre is the 3D visualization of sound. Below the pause button on the left is the 2D visualization of the sound. The cartoon bird at the top left can fly up or down according to the amplitude of the sound.

Figure 3.5: Results of Recognition. The spectrogram is divided into time slots according to the position of the birdsong. There is a medal at the top of each slot. The red medal represents a higher probability and blue means a lower probability. Results with a probability lower than 75% will not be taken and show with a transparent label. Right is a rewards list showing recognized birds.

to distinguish the bird and non-bird prediction. When the probability of identification in the slot is higher than 75%, the system will regard this recognition as a successful identification for a bird. The text on the label will be opaque simultaneously. Otherwise, the text on the label will be transparent. There is also a medal under the possibility text. The green medal is used to indicate lower possibility and red medal to indicate a higher possibility. Players can directly see the quantity and accuracy of the identification through these medals.

Under this graph is a draggable scroll view, which contains a list of possible birds. Since there are 388 bird species are trained, and not all of these birds will appear around the players, predicted birds only appearing around the player can be considered as correct predictions. Results that are not in the task will be filtered out.

On the right part of this graph, there is a “Reward“ area and a “Collect“ button. As mentioned earlier, it is necessary to have a mechanism to encourage players to make more records and recognition. In this part, when the player successfully recognizes a bird from a bird song, he will get a certain number of game items based on the probabilities of identified birds.

The number of game items is equal to the sum of all prediction accuracy level. For example, in Figure 3.5, the current player has successfully identified a bird named “Great Tit“. There are four successfully identified sound slots (only one is shown in the graph), and the prediction accuracy level of each slot is approximately 9. Therefore, the player gets 36 (4 * 9) "Great Tit" puzzles (Puzzles are shown as bird icons in rewards area on the right). After the player clicks the “Collection” button, the recognition result will be sent to the server, and the puzzles in the reward area will also be placed into player’s backpack.

This is how to combine bird recognition with game elements. The next part is about how to play the game and how the game encourages players to go out and look for birds.

Game Play

The basic game logic is represented by Figure 3.6. Once the player has successfully identified the bird, the player will receive bird puzzles representing the species of bird (In the Figure 3.5 the player got 36 bird puzzles). The number of puzzles can be stacked up when the player recognizes the same bird species again. Players can click on the puzzle item in the backpack and view the picture unlock progress (Figure 3.7). The unlock percentage is calculated by

$$\text{Unlock Percentage} = \begin{cases} 100 & \text{if } Q \geq T \\ Q/T * 100 & \text{if } Q < T \end{cases}$$

In the formula, Q is the amount of puzzles, and T is the amount of puzzles for unlocking 100% picture (In this thesis, T = 70). A higher T means that unlocking requires more puzzles, and the game is more difficult. It can be used to adjust the difficulty of the game. The calculated unlock percentage is used to determine how many white covers are removed on the image (Figure 3.7). And the order of removed white covers is random.

When the player unlocks the bird’s picture at 100%, the picture of bird can be converted into a big reward by clicking the "Use" button below the picture. 100% unlocked images provide the player with the highest rewards, including bird eggs. The Bird egg is the key elements of the game.

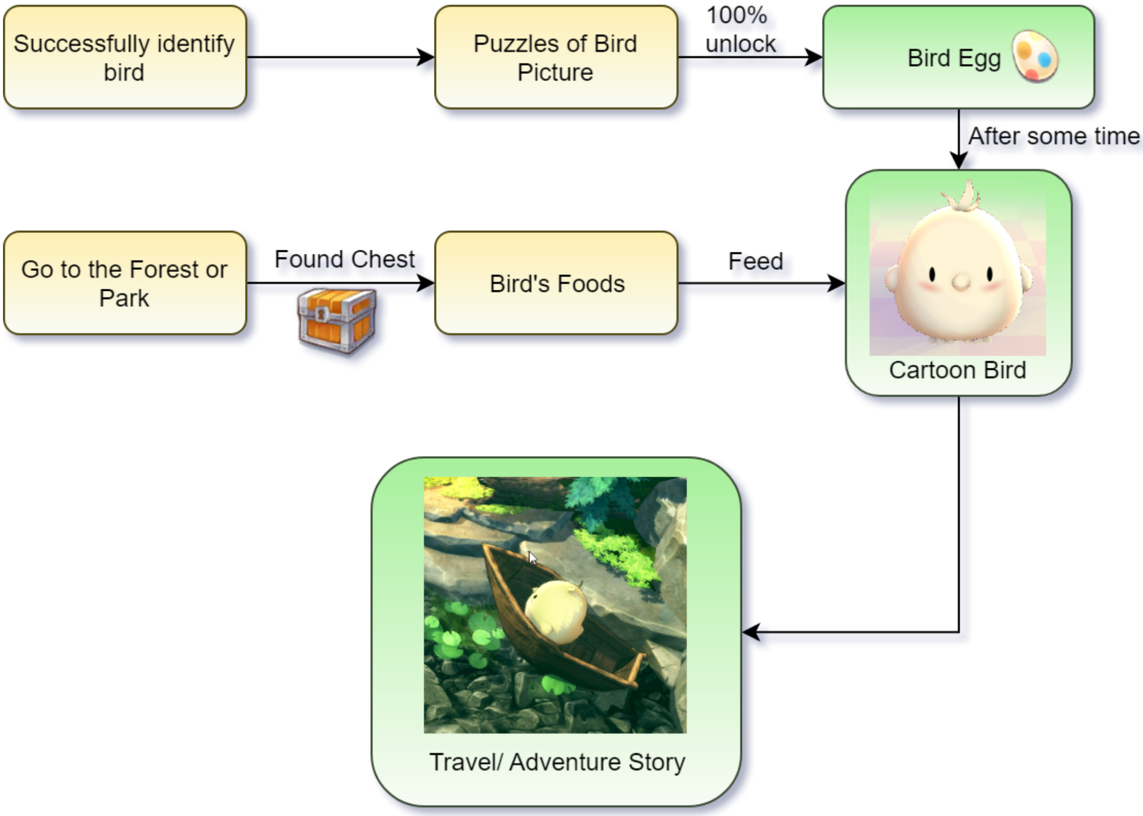

Figure 3.6: Basic Game Logic

Figure 3.7: Unlock Bird's Picture, Left: first time recognize the bird; medium: unlock 50% of the picture; right: completely unlock the bird picture

This bird egg is a virtual 3D item. It can be placed anywhere in the virtual world. The placed bird eggs will hatch into a cartoon 3D bird after a period of time. Virtual 3D birds need food to grow up. However, bird food needs to be collected by the player. The bird food can be found in the treasure chests. To encourage players to go more to the birds' favourite places, such as forests and parks. The treasure chests are only randomly generated near the tree.

In order to increase the fun of the game, bird toys are added to the game, which can also be found in treasure chests. The bird toy can make birds grow up (increase growth value of bird) when the bird play with it. Players need to feed the birds regularly to ensure their growth. This also means that players need to keep collecting bird food and need to go to new places where more birds appear.

Grown-up birds will leave the player to take an adventure. This mechanic was implemented in order to create some sort of story the user can follow. Such stories in Prestopnik & Tang (2015) have shown great potential making games more attractive to their players. When a bird is travelling, it will send a postcard with a story back to the player. Players can unlock more stories by letting the birds do more adventures.

In order to combine the story-based elements of grown-up birds taking adventures with the rest of the game and thus create further motivation for players to collect recordings, birds hunger value will drop to zero when they return from an adventure. Players need to re-feed the birds to let the birds take another adventure.

Map

A study by Buyuksalih et al. (2017) shows that 3D virtual world built from maps is a popular and valuable tool. It has better visualization and functional effects.

In the game design, it is necessary to know the player's location and surrounding environment information. Therefore a 3D virtual world was built based on real maps (Figure 3.8). At the same time, some game elements are added to the 3D virtual world, such as houses, roads, forests and parks. The location of these game elements is related to the actual object position. That means when the player walks into the woods, he or she will see that there are also virtual 3D forests around him in the virtual world.

Figure 3.8 shows that the comparison of the virtual 3D world with the actual map. The map information comes from the website OpenStreetMap[1]. The height of the virtual house in the picture is random. Because the flat map does not contain the height information of the house. Besides, trees will be filled inside the gardens or forest areas.

When a treasure chest appears around the player, the player can see the sign of the treasure chest(Figure 3.9). And the treasure chest can be opened by clicking on the game object.

There is a star in the upper right corner of Figure 3.9. The number next the star will increase when the player moves a certain distance. When the number of stars reaches a certain number, the player can unlock new characters (Figure 3.10).

[1] `https://www.openstreetmap.org/#map=19/51.41505/6.76406`

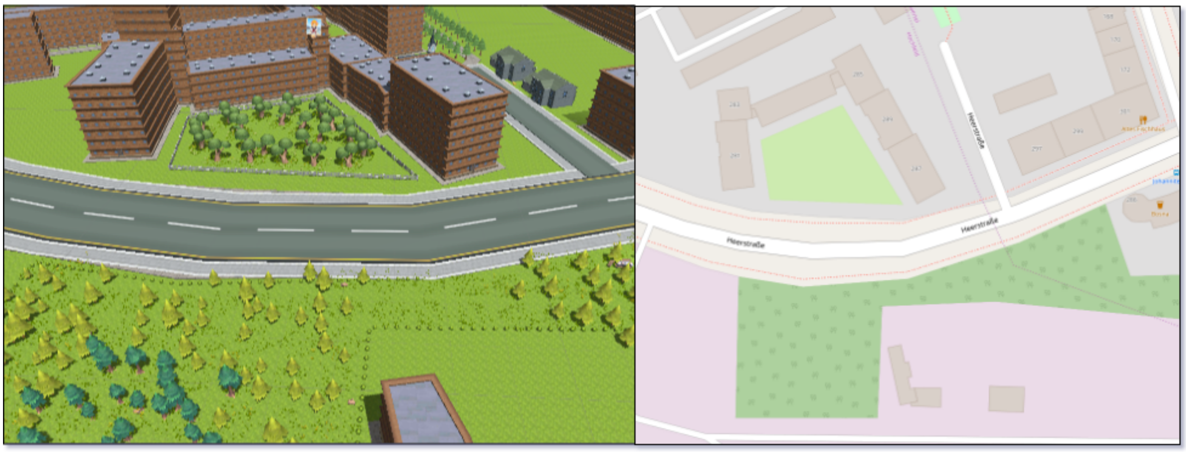

Figure 3.8: Virtual 3D World(left) Real World Map(right)

Figure 3.9: Chest Sign, A label with a treasure chest logo is displayed in the position of the treasure chest.

The star mechanism is to encourage players to go out and look for birds. The farther the players move, the more stars they will get.

So far, This is basic idea about how to identify Birds and how to combine Birdsong recognition with games. This section also describes the mechanism how to encourage players to look for bird. In the next chapter, it will describe in detail how to use Unity, Python and other techniques to design and implement elements mentioned in this chapter.

Figure 3.10: Unlock Character. Players can collect stars by walking. When the number of stars reaches a certain number, players can use it to unlock new characters. Characters that can be unlocked will display a star symbol on the icon.

3.2 Implementation of CNN

To identify birds, a basic model using Python and Keras [2] is built, and then extract the spectra using the methods mentioned in article (Choi et al., 2017). The spectroscopic image of the sound can be used as an input to the model. Finally, the model is trained, and the parameters are optimized based on the results. The whole process is shown in Figure 3.2.

3.2.1 Audios Processing

Collect Birds Information

In order to let the CNN model learn how to recognize bird species form birdsong, sound files of different birds need to be collected. First of all, ebird[3] provides a list of birds. The list includes the species of birds that people have observed in Germany in the past year. Besides, ebird also provides a database[4] for scientific research. From these data, the following bird information (Table 3.2) is adopted.

From these data, I collected two data sets. One data set contains 388 birds, which are from Germany and have relatively high observations (at least ten recordings in the past year). Another data set contains the most common or most observed birds in Duisburg and only includes 41 species of birds. Next, recordings of the bird are collected according to the name of the bird

[2] https://keras.io
[3] https://ebird.org/region/DE?yr=all
[4] https://ebird.org/data/download

Parameters	Example 1	Example 2
Common Name	Great Tit	Barn Swallow
Scientific Name	Ciconia ciconia	Hirundo rustica
Observation Count	1	5
Country	Germany	Germany
Locality	Beelitz	Gülper See
Latitude	52.2484297	52.7329551
Longitude	12.9767632	12.2501421
Observation Date	2019-05-08	2013-08-25
Time Observations Started	14:30:00	11:25:00

Table 3.2: Birds Information

species. To avoid the restriction of collecting birds at a single site, these birds are gathered from different websites. The list below shows websites for collecting bird sounds.

- `https://ebird.org/home`
- `https://www.hbw.com/ibc`
- `https://www.xeno-canto.org`

Collect Sound Files

Because these collected bird sound records have different lengths, different quality and different noise levels, it is necessary to do some pre-processing on these sound files.

First, since some sound files are too long, only the first 10 seconds of each song is intercept. Then the bird's song audios of the same bird species are put in the same folder. Sound files less than 10 seconds will be dropped. For each type of bird up to 1000 sound files are collected. Each sound file is up to 10 seconds long. That means, up to 10,000 seconds of sounds for each species of bird.

Not all bird can have so many sound recordings. Some birds are not common, and their corresponding sound files are relatively rare. In this case, I repeat the collected recordings so that these rare birds have the same amount of recordings as other birds. And they have a fair chance compare with other species for CNN. Otherwise, unequal numbers will make rare birds difficult to predict.

Finally, through the above method, I have collected 219,145 sound files from 388 species of birds - around 75.4 GB Data-set.

Spectrum Expand

The article (Choi et al., 2017) provides two ways to perform spectrum expansion with the help of CNN. One is to perform linear spectrogram expansion. The other is to perform spectrogram expansion of the Mel cepstrum. The Mel cepstrum[5] evenly distributes the entire frequency

[5] `https://en.wikipedia.org/wiki/Mel-frequency_cepstrum`

band on the Mel scale. In other words. The spectrum image produced by Mel cepstrum is closer to the human nonlinear audio system.

In general, Mel cepstrum is more widely used in speech recognition and is more suitable for human language. It expends the frequency band in the 0-10k Hz range (the human language frequency is typically in the field of 0-10k Hz) and compresses the frequency band in the 10k-20k range. However, the sound spectrum brand range of birds is very different from the spectrum brand range of humans. Their spectrum range is more than 10K Hz in many cases. In this case, the linear cepstrum method for spectrum spreading is more suitable.

Based on the introduction of the article(Choi et al. (2017)), I first build and train a CNN model for spectrogram extension (Listing 3.1) In order to implement this code, the following environment is used:

- Python 3.6.8
- tensorflow-gpu 1.14.0
- CUDA 10.0
- cuDNN v7.6.3
- Keras 2.3.1
- FFmpeg [6]

Listing 3.1: Build and Train Model for Spectrogram Extension

```
#use kera's sequential
model = Sequential()

#Add Linear Spectrogram Extraction Layer
model.add(Spectrogram(n_dft=800, n_hop=256, input_shape=src.shape,
          return_decibel_spectrogram=True, power_spectrogram=2.0,
          trainable_kernel=True, name='static_stft'))

#Training
batch_input_shape = (2,) + model.input_shape[1:]
batch_output_shape = (2,) + model.output_shape[1:]
model.COMPILE('sgd', 'mse')
model.fit(np.random.uniform(size=batch_input_shape), np.random.uniform(size=
    batch_output_shape), epochs=1)
```

Then librosa[7] is used to load a sound file and use the sound data as input to the model. The output of the model will be a 2-dimensional spectrogram image, as shown in Figure 3.11.

Noise Reduce

[6] https://www.ffmpeg.org
[7] https://librosa.github.io/librosa/

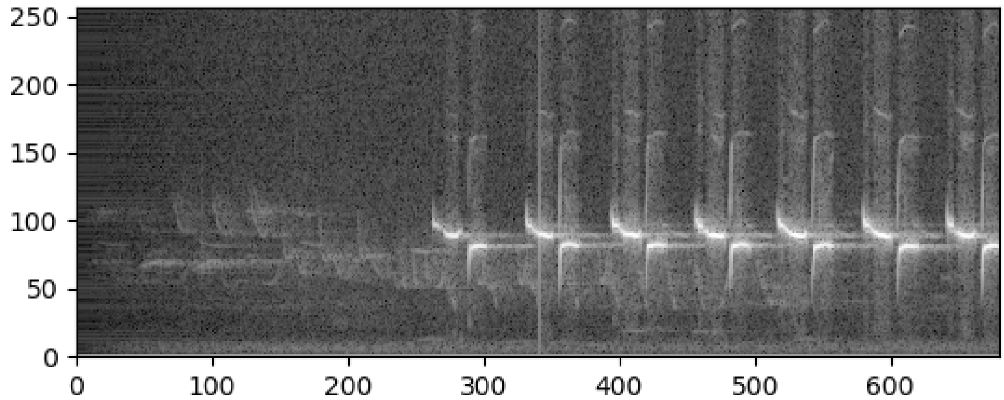

Figure 3.11: Spectrogram of Birdsong(Great Tit)

Figure 3.11 shows that the spectrogram image contains a lot of noise. The noise in the spectrogram image is shown as black and white spots. The more spots, the more the noise. These noises are distributed not only in non-bird-sound areas but also around bird sound. To reduce this noise, there are two ways that can be used. One is to set a sound decibel threshold. Sounds above the threshold will be retained, and the parts below the threshold will be zero. Another method is to separate the sound of the non-bird part. Use this part as a sample of noise. Then calculate the threshold of noise in each frequency segment and make the noise reduction for whole audio according to the calculated thresholds.

The first method is relatively simple, but the threshold determination is difficult. If the threshold setting is low, the noise removal is not obvious. However, if the threshold is set too high, the bird will be partially removed. The comparison is shown in Figure 3.12

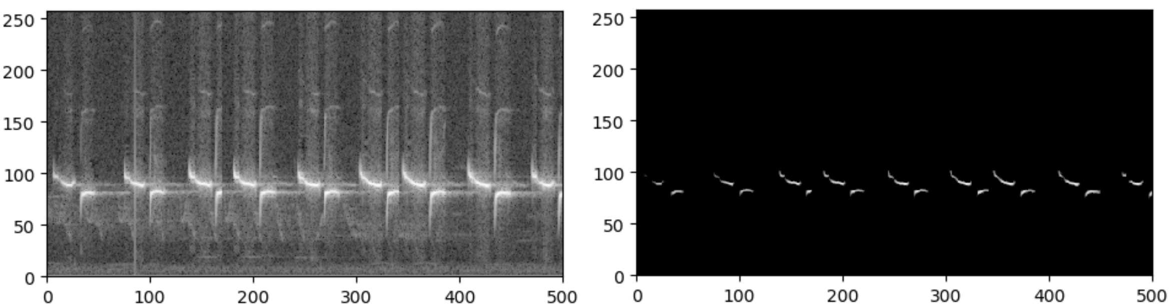

Figure 3.12: Noise Threshold Setting (Left: too low; Right: too high)

In practice, I used a random threshold so that I would produce various results under different thresholds. The advantage of this is that I can let the prediction model learn the results under different noise reduction thresholds. Therefore, the model will have the ability to make the correct predictions for sound under different noise reduction thresholds.

Another method is more complicated. First, I need to determine the non-bird sound area. I used a small window (3-pixel width) to sample the original spectrogram. If the average of the samples in the small window is less than the average of the entire spectrogram, then add it to the noise stack. The figure 3.13 shows the process. The Figure 3.14 shows the separation of the noise part.

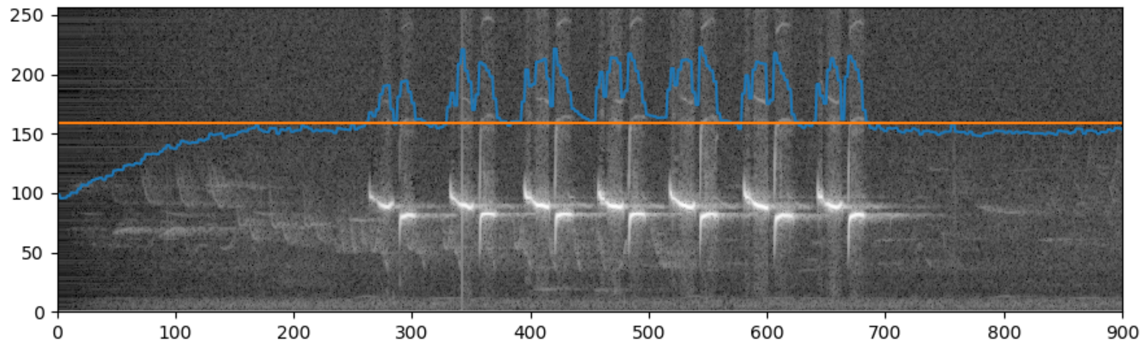

Figure 3.13: Noise Detect (Blue line: Window mean. Yellow line: Global mean. When blue line is under yellow line, the part of sound is noise.)

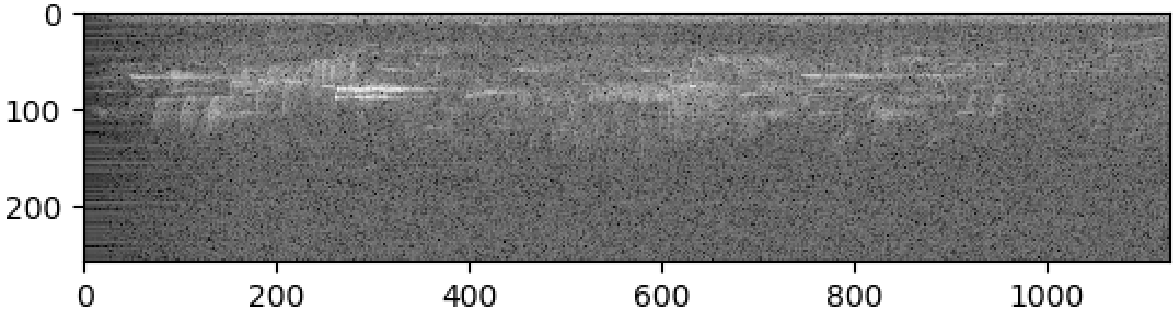

Figure 3.14: Separated Noise Part

The next step is to calculate the average value of each frequency of the noise part. Figure 3.15(Left) shows the results. This result shows that the noise has a larger mean value in the low-frequency part and a smaller value in the high-frequency region. Use this results as the thresholds for sound noise reduction, a better noise reduction results can be generated as Figure 3.15(Right).

This result shows that the second noise reduction method will remove noise dynamically. It has better outcomes for the distributed noise.

Finally, I split the results into the same shape image. Image size is 257 * 171 which represents 1-second spectrogram of birdsong. There is 25% overlap between the two slots. For each bird species, I randomly take 2400 of spectrogram images as input to the next CNN. These 2400 images are divided into 35 batches and 75 slots per batch.

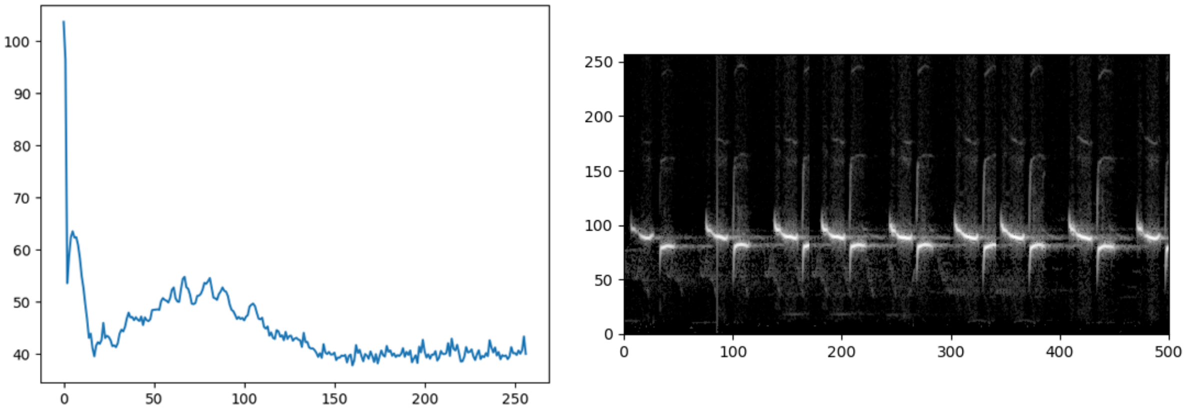

Figure 3.15: Average Values(Left); Final Noise Reduction(Right)

3.2.2 CNN model

The last chapter talked about the spectrogram expansion and noise reduction of the sound. This chapter will describe how to train the model to identify these spectrograms.

The spectrogram of the sound can be regarded as a picture. The recognition of sound is actually the recognition of these pictures. The model I have used is also based on image recognition. There are many methods for image recognition. The most common methods like VGG, Resnet, Inception, Xception DenseNet etc. But they have different model sizes and application platform. For my paper, I mainly study algorithms on mobile platforms. The mobile platform has lower computation and smaller memory than the PC platform. Therefore the model must be able to guarantee a small volume, a small amount of computation but with the highest possible accuracy.

This paper focuses on two possible methods, VGG and ResNet. Their advantages and disadvantages and the accuracy of prediction will be compared in this section.

VGG - Very deep convolutional networks

Simonyan & Zisserman (2014) proposes a CNN architecture with multiple weighting layers, which is more widely used in image recognition. Its structure is shown in Table 3.3. From its structure, the shape of the input image is (256, 512, 1). After the multi-layer convolution kernels. The final output size is (999). This means that the input image size is (256*512*1). The final output is the probability of 999 items. The item with the biggest probability is the most likely items.

Similar to this method, I built my CNN architecture to recognize bird songs. First of all, the size of the spectrogram image is 257*171. And the number of species of bird is 41(For birds in Duisburg). In addition, it is necessary to reduce the numbers of parameters to reduce the size of the model. So I built a simple model for bird recognition. The structure is shown in Table 3.4.

Layer (type)	Configuration	Output Shape
InputLayer	(256, 512, 1)	
BatchNormalization	(256, 512, 1)	
Convolution2D	64 5x5 kernels, 1x2 stride	(256, 256, 64)
MaxPooling2D	2x2 kernel, 2x2 stride	(128, 128, 64)
BatchNormalization	(128, 128, 64)	
Convolution2D	64 5x5 kernels, 1x1 stride	(128, 128, 64)
MaxPooling2D	2x2 kernel, 2x2 stride	(64, 64, 64)
BatchNormalization	(64, 64, 64)	
Convolution2D	128 5x5 kernels, 1x1 stride	(64, 64, 64)
MaxPooling2D	2x2 kernel, 2x2 stride	(32, 32, 128)
BatchNormalization	(32, 32, 128)	
Convolution2D	256 5x5 kernels, 1x1 stride	(32, 32, 128)
MaxPooling2D	2x2 kernel, 2x2 stride	(16, 16, 256)
BatchNormalization	(16, 16, 256)	
Convolution2D	256 5x5 kernels, 1x1 stride	(16, 16, 256)
MaxPooling2D	2x2 kernel, 2x2 stride	(8, 8, 256)
BatchNormalization	(8, 8, 256)	
Flatten	(16384)	
Dropout	dropout 0.4	(16384)
Dense		(1024)
Dropout	dropout 0.4	(1024)
Dense		(999)
Total Params	19,523,883	

Table 3.3: VGG 16 Architecture

Layer (type)	Configuration	Output Shape
InputLayer	(257, 171, 1)	
Convolution2D	32 3x3 kernels, 1x1 stride	(257, 171, 32)
MaxPooling2D	2x2 kernel, 2x2 stride	(128, 85, 32)
Convolution2D	64 3x3 kernels, 1x1 stride	(128, 85, 64)
MaxPooling2D	2x2 kernel, 2x2 stride	(64, 42, 64)
Convolution2D	128 3x3 kernels, 1x1 stride	(64, 42, 64)
MaxPooling2D	2x2 kernel, 2x2 stride	(32, 21, 128)
Convolution2D	256 3x3 kernels, 1x1 stride	(32, 21, 128)
MaxPooling2D	2x2 kernel, 2x2 stride	(16, 10, 256)
Convolution2D	512 3x3 kernels, 1x1 stride	(16, 10, 256)
MaxPooling2D	2x2 kernel, 2x2 stride	(8, 5, 1024)
Convolution2D	1024 1x1 kernels, 1x1 stride	(16, 10, 256)
Average Pooling2D,Flatten,Dense		(41)
Total Params	2,429,225	

Table 3.4: Birdsong Recognition Architecture

The biggest difference between my model and VGG is that I reduced the size of the image and the number of image classes. And I use only one Dense layer at the end. Total parameter amount of the model is only 2,429,225 (Compare with VGG: 19,523,883). The reason for reducing the parameters is to allow the model to run smoothly on the mobile platform.

However, I can only train 41 species of birds in Duisburg by this method because the model is not suitable for a large number of classifications. When the number of classifications is too large, the training of the model will appear over-fitting problem. And the accuracy of the model does not rise with training time (Figure 3.16).

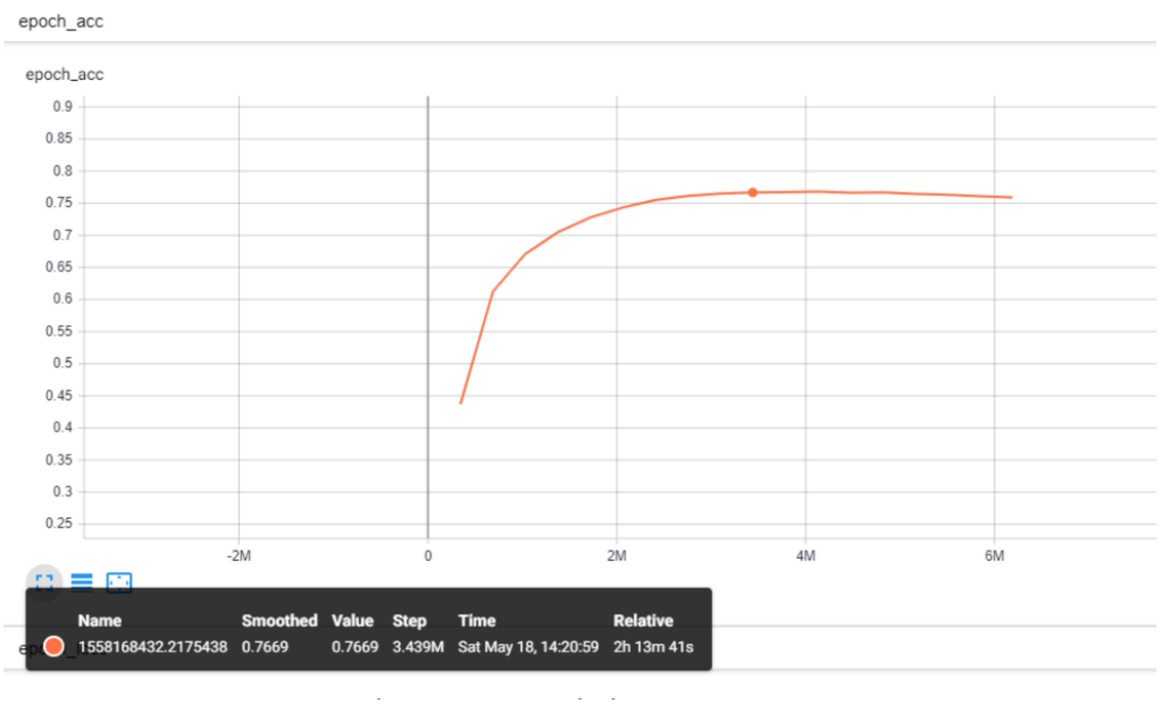

Figure 3.16: Over-Fitting, Accuracy does not improve over time

ResNet

He et al. (2016b) has mentioned that increasing the number of layers in the convolution model sometimes does not improve accuracy very well. Indeed, as the number of layers increases, the computational efficiency of the model decreases. And the accuracy of the final result will be worse as before.

The deep residual neural network(ResNet) by He et al. (2016b) is using “shortcut connection“ to improve the convergence speed and classification accuracy of very deep CNN. Short connections make the signal easier to pass through the network, reducing gradient explosions and gradients disappearing problems. And it makes the model easier to train.

The residual network consists of residual units. In a residual uint, the input of the convolutional layer and output of the normalizing layers are added together as the unit output. The residual unit is shown in Figure 3.17.

With the help of ResNet, I improved the original training model. The input shape in the new model does not change, but the number of types of output can be increased to 388. The new model is made up of blocks. He et al. (2016c) provides several different forms of blocks. Using different blocks for different size of the model will improve accuracy. For my model, I used the basic block form, which is defined as shown in Figure 3.17.

To implement ResNet, I used the existing Keras project[8] to define the model. The configuration of the model is shown in the Listing 3.2

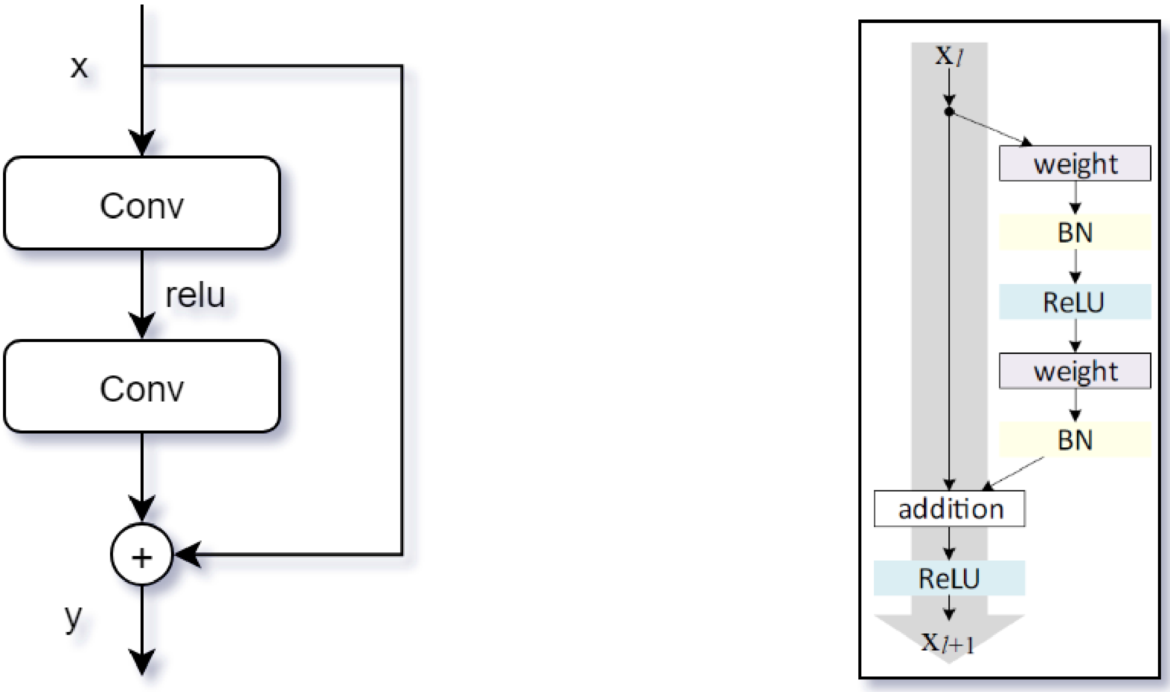

Figure 3.17: Left:A simplified residual unit; Right: Basic block of ResNet

Listing 3.2: Build ResNet Model for Spectrogram Extension

```
# Generate ResNet Model

# config.Spec_Shape = (257,171,1)
# len(self.labels) = 388
# The starting number of filters is 32
# Repetitions means times of basic block.

model=resnet.ResNet(config.Spec_Shape, LEN(self.labels), resnet.basic_block,
                    repetitions=[1,2,3,2],initial_filters=32,initial_kernel_size
                        =(7,7))
```

[8] https://github.com/raghakot/keras-resnet

Data Shuffle

The meaning of shuffle in machine learning and deep learning is to disrupt the order of data set for the training model.

The original data may be arranged in a certain order related to the data-set. For example, the first 1000 data inputs belong to one same group and next 1000 are in another category. When a large amount of continuous data (the first category) is input for training, it will cause over-fitting of the model on the first type of data. After the first type of data learning is over, the model begins to learn a large amount of another class, which will make the model try to approach the second category. This will result in a new over-fitting phenomenon. Such a repetitive training model will result in model jitter, which is detrimental to the rapid convergence of the training.

In order to avoid this problem, the data set needs to be shuffled to have some randomness. And the probability that the next sample obtained in the sequential reading is the same for any categories. In practice, I use sklearn.utils.shuffle[9] to process data shuffle.

Data Generator

When training models with kears, all training data is generally loaded into memory and fed to the network in one time. However, when the memory is limited, and the amount of data is too large, this method is no longer available. A large number of data will cause out of memory problem (OOM).

At the same time, I should also perform data enhancement for each data unit to avoid over-fitting and increase the generalization ability of the model. In these cases, I took advantage of Keras's generator function[10].

The generator can run on the CPU and perform real-time data enhancement, while the training of the model can be run in parallel on the GPU. Such an operation can improve the actual efficiency. On the same time, the generator can control the size of the data input to the memory. It will release the memory after completing one batch and load the data of the next batch. It can solve the problem of OOM.

For data enhancement, I use left and right random jitter for the data. And add random noise according to a certain probability. Combined with data shuffle, the entire algorithm is shown in Listing 3.3.

Listing 3.3: Generator for Training

```
# Generate for Generator for Training
DEF GeneratorTrain(self):
    npyes=self.allNpyes,count=0,x = [],y = []
    WHILE 1:

        #Shuffle Data
```

[9] https://scikit-learn.org/stable/modules/generated/sklearn.utils.shuffle.html
[10] https://keras.io/models/sequential/

```
        npyes=shuffle(npyes)

        FOR i IN RANGE(0,LEN(npyes),config.BatchSize):
            x=[],y=[]
            FOR k IN RANGE(0,config.BatchSize):
                npy=npyes[i+k]
                data=ifile.LoadArray(npy)
                dirs = ifile.GetDirsFromPath(npy)
                species = dirs[-2]

                #Position jitter
                pos=np.random.randint(128)-64
                data=np.roll(data, pos, axis=1)
                IF pos < 0:
                    data[:,pos:] = 0
                ELIF pos > 0:
                    data[:,:pos] = 0

                #Add noise
                IF(np.random.rand()<0.75):
                    noise=np.random.rand(data.shape[0],data.shape[1],data.shape
                        [2])*0.6
                    data=data+noise
                IF(np.random.rand()<0.75):
                    data=data*(0.6+np.random.rand()*0.8)

                x.append(data)
                y.append([species])

            y = self.LabelToOnehot(y)
            x = np.array(x)
            y = np.array(y)

            yield(x,y)
        CONTINUE
```

After the data is ready, the training can be started. 10% of the trained data set will be used for validation. Training will also use EarlyStopping[11] to stop training when accuracy stops increasing. After the training for each batch, I saved the highest accuracy parameters to the file. In the end, I got the best model parameters.

3.2.3 Prediction and Results

This section shows that the training curve and prediction method. Training is under in the following environments: GTX 970, Intel Core i7-6700K, 16GB Memory.

The figure 3.18 shows 3 training sessions. These three sessions are performed sequentially. Each training lasts approximately 7-9 hours. After each training, the new training first loads

[11] https://keras.io/callbacks/

the best parameters saved in the last and then start a new session. The final training accuracy is about 77%

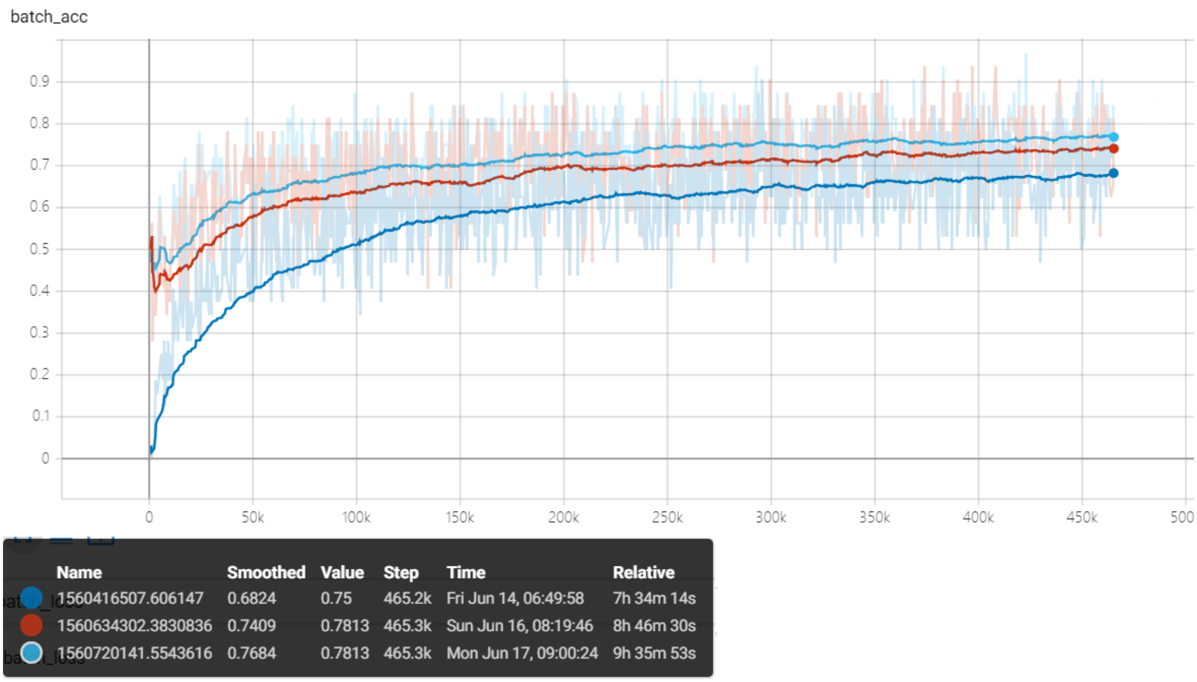

Figure 3.18: 3 times Training Curve

With a well-trained model, I can start the prediction for birdsong. Same as training. When I get a new sound file, I need to expand the spectrogram first. The spectrogram extraction algorithm is the same as the algorithm used in training. Then divide the expanded spectrogram into slots of one second. And I use the trained model to make predictions separately for each cell. The predicted result for each cell will be 388 possibilities for each species. Then I calculated the sum of the possibilities for each species. The species with the largest sum of possibilities will be the most likely species.

3.3 Implementation of Game

This chapter describes how to use Unity to implement the game design. The game will run on the mobile platform. In addition to the mobile platform, I also built a simple HTML server for receiving bird recognition records sent by users.

3.3.1 System Structure

Figure 3.19 shows the basic framework of the game. The game client is implemented using Unity. I also used some third-party unity plugins to speed up the development process, and some 3D models are also from the unity store[12]. On the server-side, I mainly use node.js to build the HTML server. The client sends the predicted result to the server via the PUSH method of HTML.

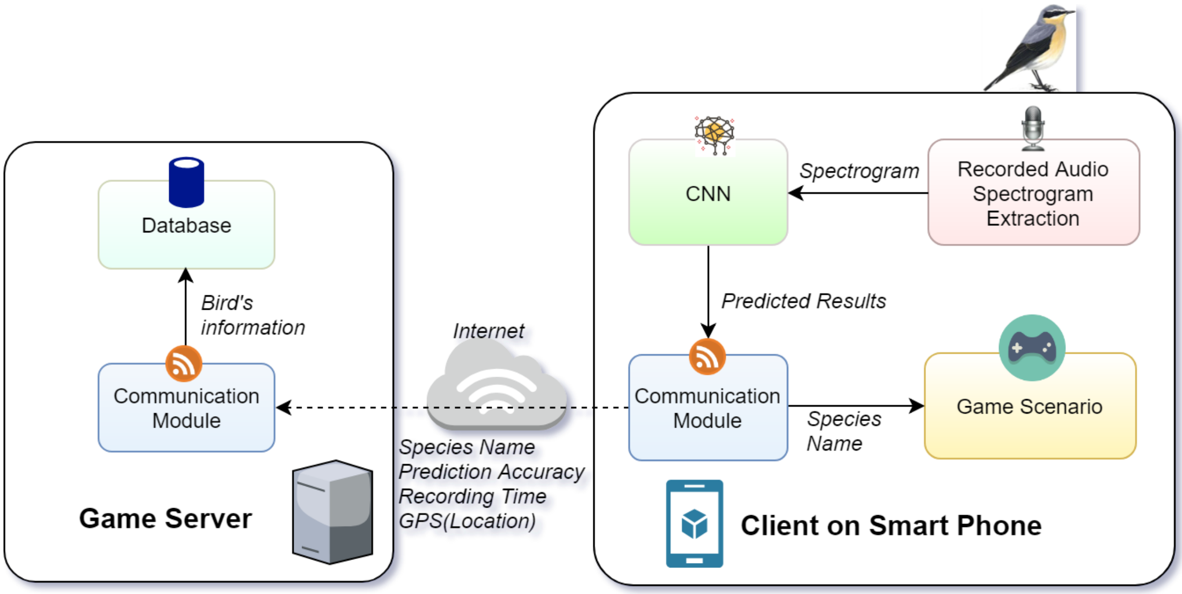

Figure 3.19: Game Structure

3.3.2 Bird Song Recognition

This chapter shows how to use previously trained models in Unity. Because the Python language, which is used to train the model before, cannot be used directly in Unity. I need to use TensorFlow under the c sharp version to load the previously trained model.

Machine Learning under Unity

Unity Machine Learning Agents (ML-Agents) [13] is an open-source Unity plugin that allows users to train smart agents in the game and simulation environments. It supports reinforcement learning, imitation learning, neural evolution or other machine learning methods, and controls the agent through an easy-to-use Python API.

For my system, I use the ML-Agents version 0.6.0a because this version uses Tensorflow csharp, which could support my CNN model.

To use the well-trained model in Unity. It is necessary to save the parameters of model in a Unity recognisable format. In practical, I first freeze the model graph and then store the parameters as a binary file. The freeze program is given by TensorFlow Github[14]. The basic

[12] `https://assetstore.unity.com`
[13] `https://unity3d.com/machine-learning`
[14] `https://github.com/tensorflow/tensorflow/blob/v1.12.0/tensorflow/python/tools/freeze_graph.py`

save code is shown in Listing 3.4. On the Unity side, I can use the Listing 3.5 to recreate the session with the saved parameters.

Listing 3.4: Save Model Graph to Unity

```
# Save Model Graph to Unity
DEF Save2Unity(self):
    K.set_learning_phase(0)

    # Load Model Parameters
    self.LoadModel(config.Model_ModelPath)

    # Freeze Model Graph
    frozen_graph = self.freeze_session(K.get_session(),
        output_names=[out.op.name FOR out IN self.model.outputs],
        input_names =[inp.op.name FOR inp IN self.model.inputs])

    # Save Path
    path=config.Model_ModelPath[:-3]+'.bytes'

    # Write Byte File
    tf.train.write_graph(frozen_graph, "",path, as_text=False)
```

Listing 3.5: Unity Recreate CNN Session

```
//  Unity Recreate CNN Session
// Initiate CNN Session
public VOID InitCNN()
{
    // Close if it is opened
    IF (cnnSession != null)
    {
        cnnSession.CloseSession();
    }

    // Create new Graph Object
    cnnGraph = new TFGraph();
    // Load Paramters from bytes file
    cnnGraph.Import(CNNModel[CNNSelect].bytes);
    // Create new CNN Session from Graph
    cnnSession = new TFSession(cnnGraph);

}
```

There are two models I need to import. One is for the spectrogram extraction, and the other is for the birdsong recognition. Although unity provides APIs[15] for spectrogram expansion, I still used my own spectrogram expansion model. First, since the training also uses this

[15] https://docs.unity3d.com/ScriptReference/AudioSource.GetSpectrumData.html

method to generate a spectrogram, the result of the spectrogram expansion will be the same as the training. Second, Unity's API is often used as a visual visualization of audio. It does not have enough accuracy for my approach.

After the spectrum is expanded, the system makes the noise reduction and split the results into one-second slots as during the training. In addition, the prediction method is the same as the method in the last chapter. Execution of predictions is very fast, but on mobile platforms, it still takes a few seconds to complete. So in order not to affect Unity's main thread, I put these tasks on the child thread to execute.

Sound Visualization

To have a good user experience, a visualization of the spectrogram is made (Figure 3.4). The main part of visualization is 3D visualization of spectrogram of sound(Figure 3.20).

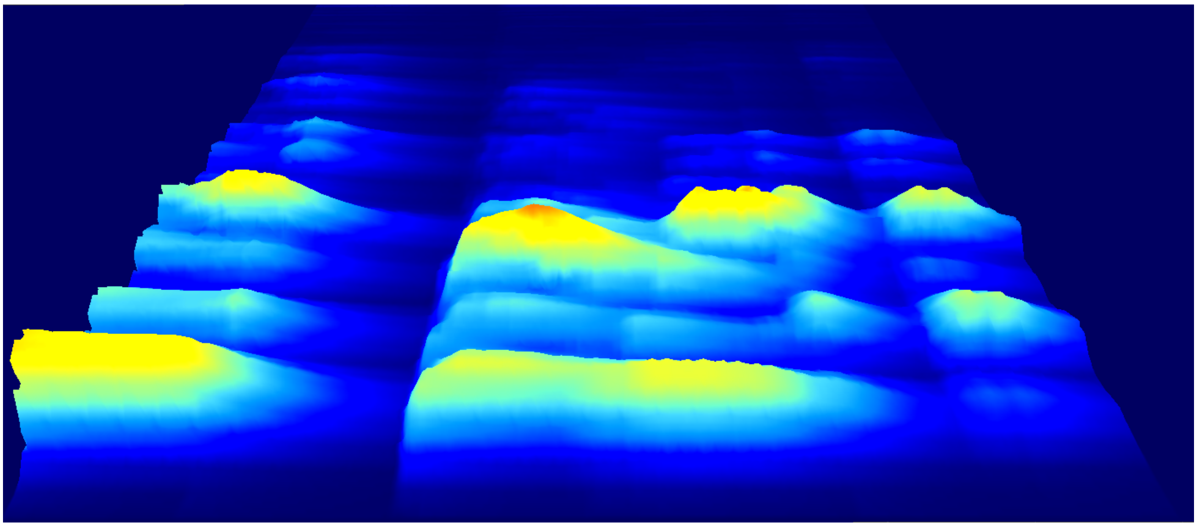

Figure 3.20: 3D Visualization of Spectrogram of Sound

To achieve this effect, I first converted the spectrogram into a 2D image and set it as a plane's texture. Then used a custom shader to give the plane new colours and calculated height position for each vertex. The custom shader is made by Amplify Shader Editor[16] which is a plugin from Unity Store. The configuration of this custom shader is shown in part in Figure 3.21.

In this figure, the green boxes on the left are the input parameters. It consists of three sets of variables that are dedicated to 3 different colours. Each set of variables contains a range value for one colour and a centre value of the range. Next to these parameters are 3 GetRange function boxes. Base on the range and centre value, GetRange box function could select special pixels from the input, which is a picture representing spectrogram. The output of the GetRange box function will be joined to the multiplication function with a fixed colour. Finally, these three colour parts will be added together.

[16] https://assetstore.unity.com/packages/tools/visual-scripting/amplify-shader-editor-68570

At the same time, the height of each vertex will be proportional to the colour value of the input image (2D spectrogram). That means that the higher the value on the spectrogram, the higher the position of the corresponding pixel. Figure 3.22 shows the configuration of height.

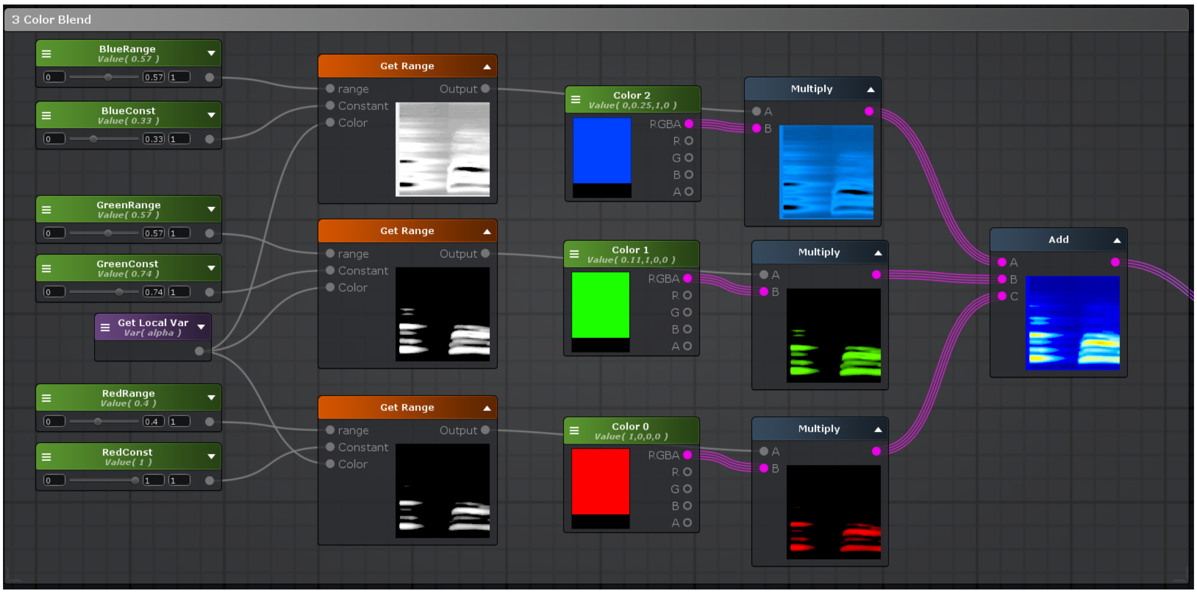

Figure 3.21: 3 Color Blend of Custom Shader

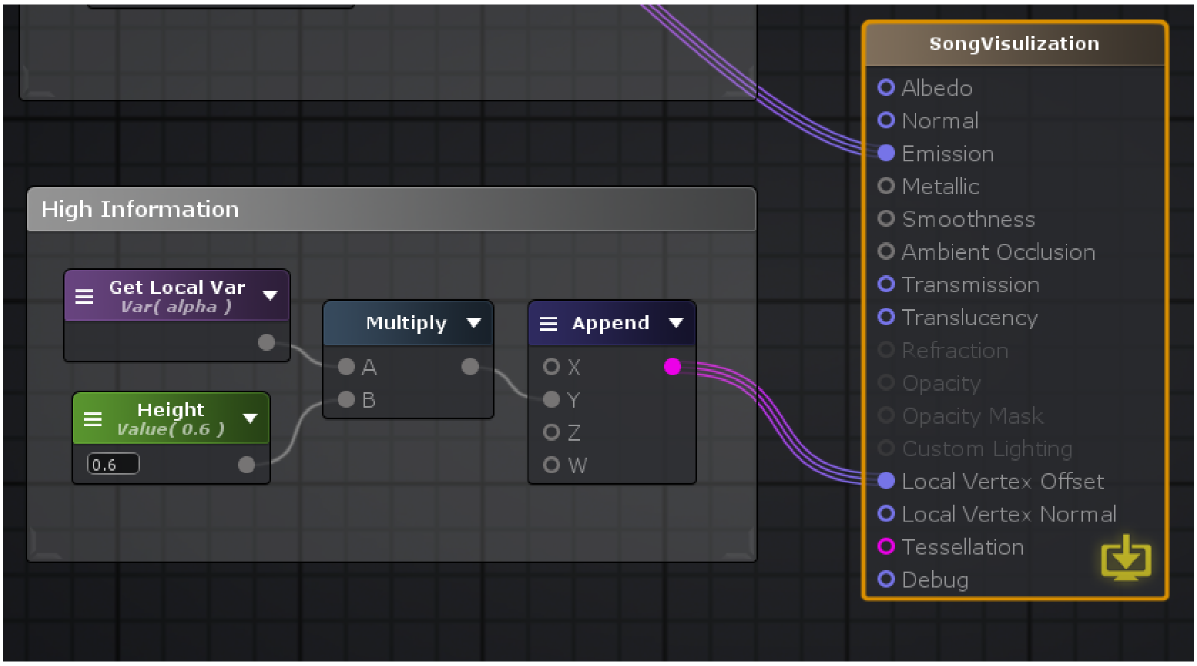

Figure 3.22: Height Information for Vetex in Custom Shader

In addition, the generated 3D surface will be captured by a dedicated camera, and the captured dynamic image will be embedded into the 2D UI for display.

After Prediction

The name of the predicted birdsong will be passed to the reward module. Players get rewards only if the possibility of bird in prediction is above 75% and the bird species exist in the quest list. The reward module will search for the corresponding bird puzzle using the predicted bird species name. It will add a bird puzzle to the list on the right. At the same time, it will also check and update the list of achievements and trigger the achievement completion event.

3.3.3 Item System

Main Categories

Another basic part of this game is the item system. The game item system mainly maintains a database of items, which include the following categories.

- Bird's Puzzles
- Cartoon bird
- Cartoon bird's postcard
- Consumable items
- Characters
- Achievement
- Items that can be placed

The bird puzzle is the bridge between the bird identification system and the game logic. When the player recognizes the bird, he will get a certain number of puzzles of the identified bird. The player needs to collect a sufficient number of puzzles to unlock the full picture of a bird and then get the maximum rewards.

The cartoon bird is a 3D bird pet. 3D bird pets have several different colours. After hatching an egg (obtained from a completely unlocked bird puzzle), the player can get this bird pet. 3D bird pets have activities such as finding food, playing with toys, growing up, flying and travelling. It can be considered as one of the rewards for the recognition of real birds.

Another interesting thing about 3d bird pets is the bird's postcard. When the bird is in travel, it will send the adventure story (written on the postcard) to the player. However, the condition of a bird's travel is that the player needs to go out and look for bird food (appearing in a forest, garden or park). This is another way to engage players to do outdoor activities.

Consumable items are items that players can use directly. For example,"Lucky Clover" can be used to destroy buildings.

The character is a cartoon human 3D model used to represent the player. The model is placed at the location specified by the GPS and has a walking/running animation as the player moves. It will also give an indication to show if the player is in the building. There are several different models available, including some boys and girls with different outfits. Only three models are unlocked as default. Players can unlock new model by walking a certain distance.

Achievements are conditions that players can achieve, such as identifying five birds.

Items that can be placed include architectural blueprints, bird food and bird toys. They are objects that can be found in treasure chests (appearing in forests, gardens and parks) or by unlocking bird puzzle. Players can use them to create 3D game objects near their location. 3D bird pets will find bird food or toys nearby and have interaction with them.

Attributes

All items have some common attributes including:

- Name
- Type
- Atlas
- Item level
- Description
- ID
- Parameters

Name is the item name. The name is written in "Key.Value" form. For example, the "Bird Food" item's name is "item.name.birdfood01". This name is not a display name (show on the UI) because it needs to be converted to the target language when it is set. The player can switch the target language to English, German or Chinese in the settings.

To convert a "Key.Value" pair to a display name, there is a table that specifies all the conversions. This table is written to a CSV file containing all the displayed text. Developers can add new languages by adding new columns to this file. Figure 3.23 shows an example.

KEY	English	Deutsch	中文
Window.Title.Bag	Bag	Tasche	背包
Window.Title.Setting	Setting	Einstellung	设置
item.name.birdfood01	Donuts	Donuts	甜甜圈
item.des.birdfood01	[B56E50]Donuts, birds' favorite food!	[B56E50]Donuts, das Lieblingsessen der Vögel!	[B56E50]甜甜圈，小鸟最爱的食物！

Figure 3.23: A Example in CSV file of Language Setting

Type is the category of the item. It has been described above.

Atlas is an icon set file that contains several different icons. Instead of using separate icon files, the icon set file can load multiple icons at once. It saved the loading time and drawing calls for graphics rendering. Figure 3.24 shows a character atlas that contains all character icons and 3D bird pets icons.

Figure 3.24: An Example for Atlas, An atlas file contains different icons

Item level is a number from 0 to 9. It can indicate the importance level of the item. For display, players can see different colour frames related to different levels. Figure 3.25 shows 3 different level and corresponding frames.

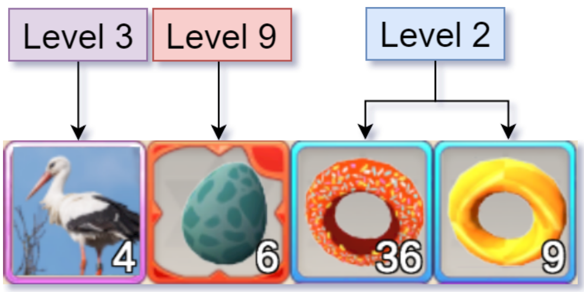

Figure 3.25: Different colour frames, such as level 3 is purple, level 9 is red, and level 2 is a blue frame. The right bottom number is the amount of item.

This description is the text describing the function of the project. In fact, the description stores a "Key.Value" pair, same with the name attribute. For display, it will be converted the

"Key.Value" to the target language based on the language CSV table.

The ID is a unique number of the Item. Items can be distinguished from other items by ID because their names may be the same.

Parameters is a list that contains the constant values of an item. The elements of this list are "key.value" pairs, which use a string as a key and a data object as a value. Data object includes three types of public variables, including string values, numeric values and game object prefab. Figure 3.26 shows an example of parameter setting for "Luck Clover".

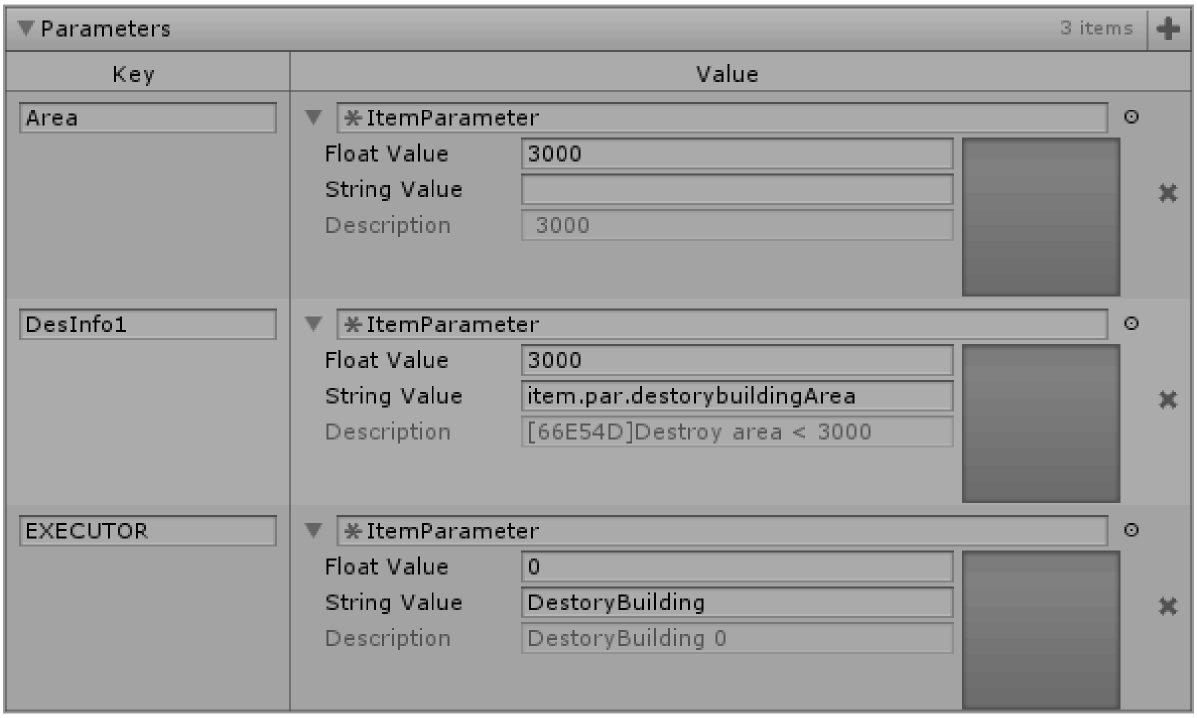

Figure 3.26: The parameters of "Lucky Clover". Including 3 elements, the key "Area" indicates the area of the building that the item can destroy. The Key "DesInfo1" shows the display information. The key "EXECUTOR" provides the script name for execution.

Editor Window

To simplify item editing, the third Unity plug-in Odin[17] is used to create an editor window for the database. Figure 3.27 shows the interface of this Editor. Developers can add or edit items through it. It also has some useful features for a quick edit, such as adding an ID for item (button Generate ID), adding the parameter "EXECUTOR" (green button), and so on.

[17] https://odininspector.com

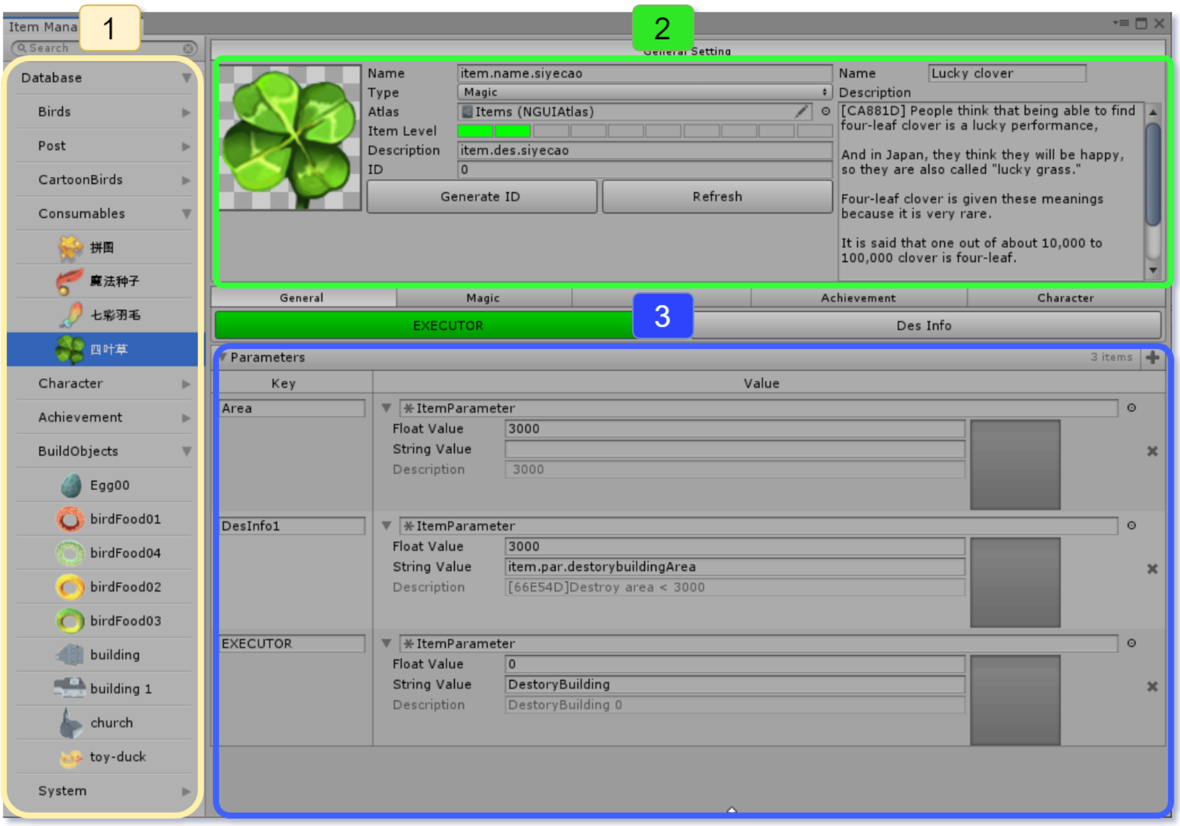

Figure 3.27: Item Editor, Part 1 is the category and item selector. Part 2 is the attributes settings. Part 3 is parameter configuration.

3.3.4 AR MAP

To display the player's surroundings, a 3D map was developed based on 2D maps. Figure 3.28 shows the basic workflow for generating 3D environments from 2D map. Here the 3rd-plugin Archimatix[18] is used to convert the 2D shape to 3D object.

Download 2D Map Information

First of all, the 2D map information is downloaded from Openstreetmap(OSM), which provides details of each element on the map. There is an API introduction [19] that can be used to download map data from Openstreetmap.

To retrieving map data around the player, a range should be given to URL as :

https://api.openstreetmap.org/api/0.6/map?bbox=left,bottom,right,top

In this URL, left, bottom, right, and top are the range of a box in which the data is retrieved. It uses longitude and latitude to define the box range. For example, to get map data for

[18] `http://www.archimatix.com`
[19] `https://wiki.openstreetmap.org/wiki/API_v0.6`

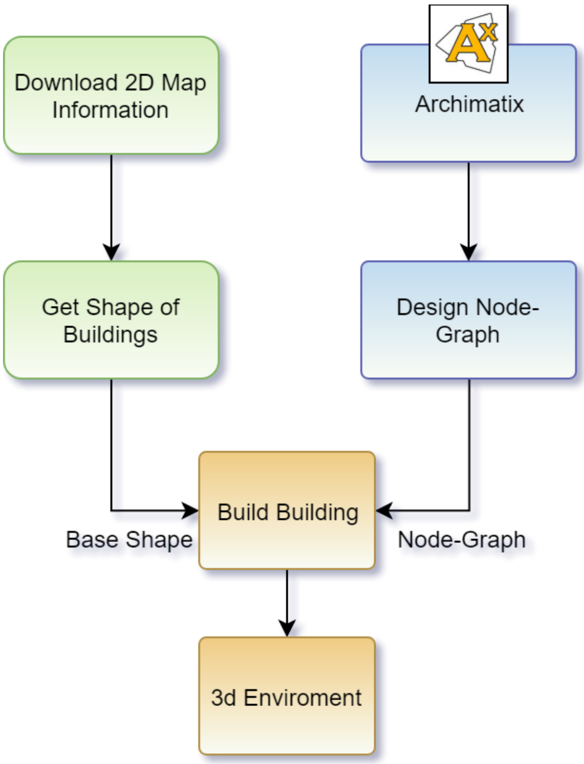

Figure 3.28: Basic workflow for generating 3D environments from 2D map

Duisburg-Esson university, the URL is `https://www.openstreetmap.org/api/0.6/map?bbox=6.79758,51.42642,6.80186,51.42809`

Get Shape of Buildings

The data model in OSM contains three basic components, which are nodes, ways and relations. Most of the components have a tag which describes the function of elements.

A node is a point on the map. It has longitude and latitude information to describe its position.

A way is a list of nodes that represent the shape of a street or building.

Relations can contain a list of nodes or ways or even other relations. It is used to represent the complex relationship of objects. This is useful for large structures that contain multiple sections.

Figure 3.30 shows the node(green circle) and the ways(red line) of building LX.

Archimatix

Archimatix is a powerful 3rd modelling plugin for Unity. It uses node-based parameters to describe the 3D model. These nodes are usually 2D shapes. By some operations such as extrude and lathe, it can convert 2D into 3D objects.

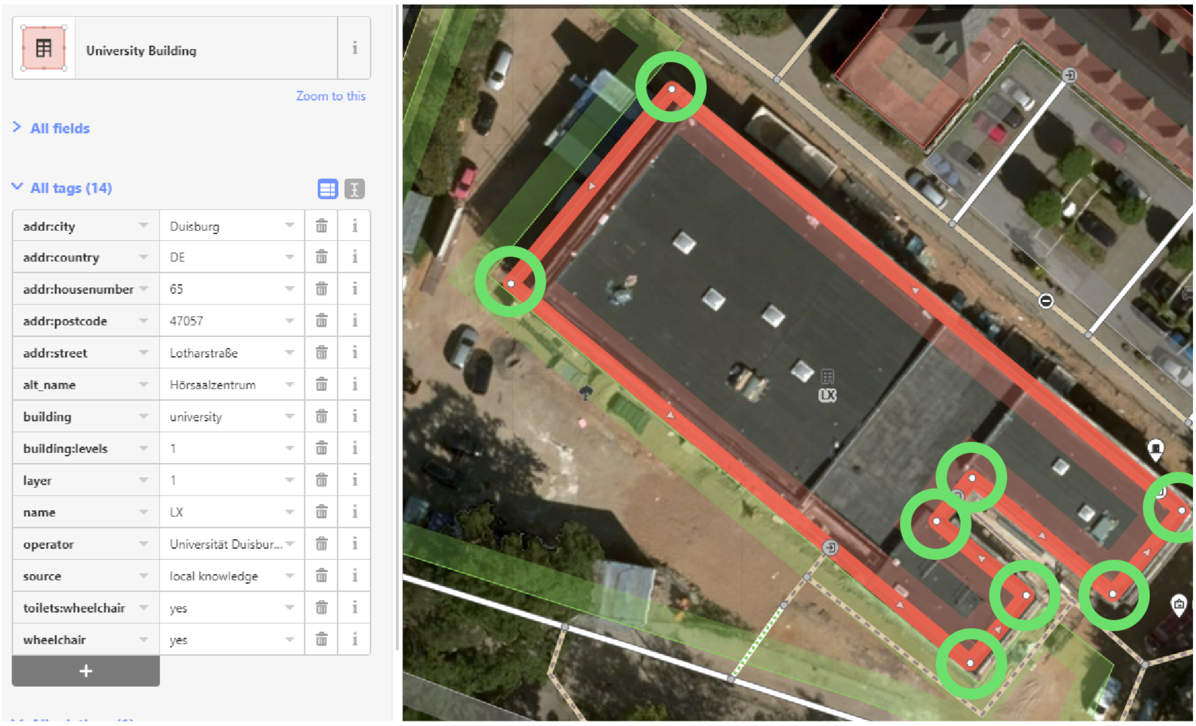

Figure 3.29: Nodes and ways in OSM, green circle: nodes, red line: ways, on the left is tags list

Node Graph and Build Building

Figure 3.30 shows the nodes configurations for building LX. And on the left of the figure is the generated 3D model. Like buildings, other elements can also be created using Archimatis, such as roads, gardens, forests and bus stops.

In practice, elements will be created in 100 meters around the player. As the player moves around, the map will dynamically generate new elements or disable old elements which are far away from the player.

3.3.5 UI

The UI system was built with the help of a third plugin, NGUI, which is an open-source and powerful UI system plugin for Unity. It makes it easy to quickly implement different components such as buttons, scrolling views, dragging bars and tables/lists etc.

Figure 3.31 shows the main UI of the game. Players will see it first when they log in.

When the player clicks the first button on the right, the camera will be positioned at the player's current position as the centre.

The second button is to open the backpack. In the backpack, there are 4 panels, item inventory, collections, character, my birds and setting. The opened backpack is shown in Figure 3.32

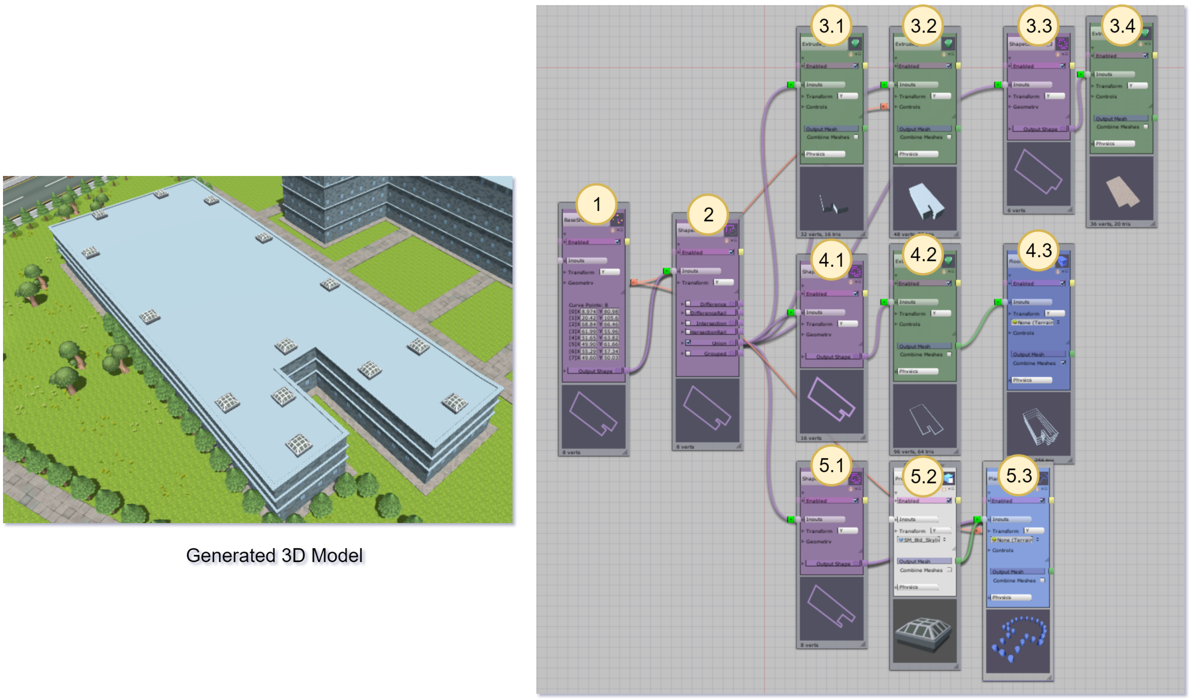

Figure 3.30: The node configuration used for the building(right) and generated 3D model(left). Node 1 and node 2 are the process of input 2D shape. The 3.1-3.4 nodes build the root and side parts. 4.1-4.3 make the side floor. 5.1-5.3 nodes build the window at the top.

Figure 3.31: The main user interface. The centre is the character. The three buttons on the right represent locate the player, open the backpack and bird recognition. At the top right is a panel showing the star number (walking distance).

Figure 3.32: The opened backpack. On the left is the panel that displays the item details (the bird puzzle is currently displayed). The right part is a table of items. There is a filter at the bottom to classify the items. The tags on the right are used to switch panels.

The Collections panel is shown in Figure 3.33. In collection panel, there are 3 sub panels which are My Collection, See Surrounding and Achievement.(Figure 3.34)

Figure 3.33: A collection panel. Now it is showing the identified birds. The right part shows the names of the recognized birds, the number of times of identification and the first/last time of recognition. Three buttons on the left to switch between the My Collection, See the surrounding and Achievements.

Unlocked characters display their icons on the Characters panel(Figure 3.35). The locked characters will be covered in white. If the player reaches the condition to unlock the new character, a star will appear on the character slot. The player can switch or unlock new characters by clicking the character slot.

In the My Birds panel(Figure 3.36). There is information about the cartoon 3D birds, including the states of birds as well as a mailbox for bird postcards(Figure 3.37).

Figure 3.34: Left: See the Surrounds panel, which displays the birds that may appear based on the location of the player. Right: Achievement, which shows the process of achieving progress.

3.3.6 Character

First, there are four components on the character game object: all character controls, GPS, motion control, and player information.

All character control components will manage the character model. Its features include changing characters, unlocking new roles, recreating models, and setting up animation controllers etc. For animation, a Blend Tree(Figure 3.38) is used to control the animation. This blend tree will take the speed of the player as an input parameter. Depending on the speed, it mixes the animation from idle(speed = 0) to walking(speed = 0.33) and then to sprinting(speed = 1).

The GPS component is used to manage the functions of the mobile phone's GPS module. When the game starts, it will enable the GPS function of the phone. And will get latitude and longitude per second. When the location changes, it also sends an event to notify other components (such as motion components and maps) to update.

When the GPS component sends an update event, the motion control component smoothly moves the character game object. It also calculates the moving distance and adds it to the star number. The model will also detect surrounding trees and create treasures near the trees.

Figure 3.35: Character panel. It shows the unlock characters on the right table. On the left is a 3D model display case, and there are two slider bars on the bottom and left to rotate and scale the 3D model.

Figure 3.36: The bird panel shows a cartoon bird pet. On the left is a list of pets for birds. The right is the bird's information, incluing bird's name, age, hungry value and percent of growing up. At the bottom right is a mailbox containing postcards of the bird.

The creation of the treasure chest is controlled by the "ChestSetting" script. It records each treasure chest created and prevents the creation of treasures from being too close to another box.

Figure 3.37: Postcard. A story(left text) with an image (background) can be written here. Birds can also send by some gift(right list) to the player.

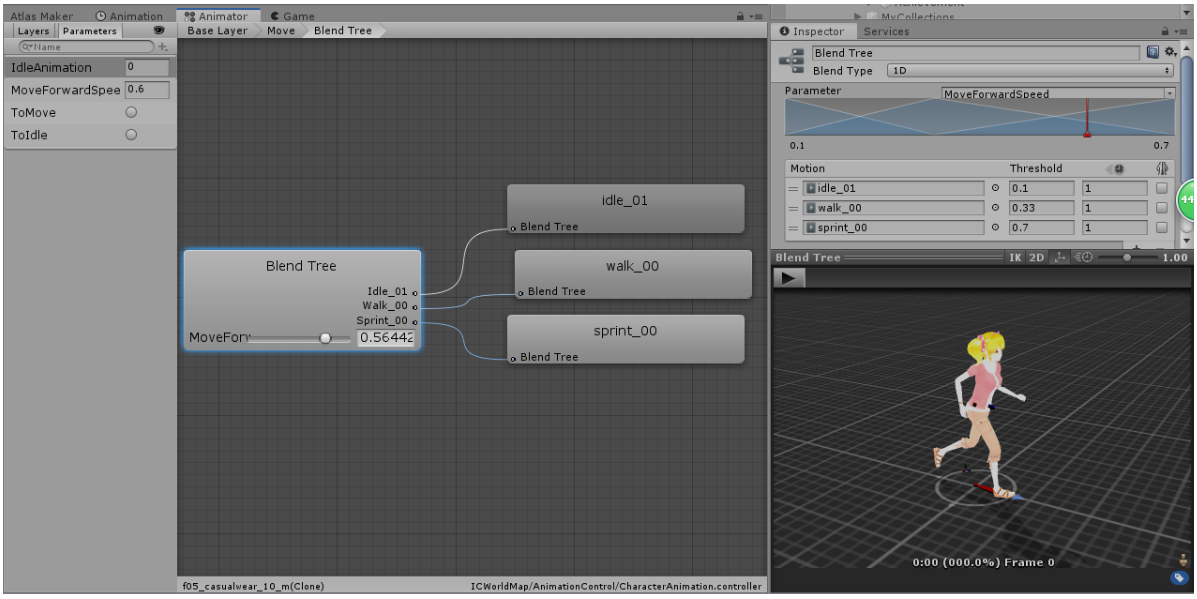

Figure 3.38: Blend Tree for character animation. Blend form idle, walk and run animation according to the speed.

3.3.7 3D Cartoon Bird Pets

3D Cartoon Bird Pet comes from a bird egg. The bird egg is a placeable object that the player can place it in the surrounding environment. After the player places the egg on the ground, a countdown timer will be activated. The default time is one days. In other words, this egg will become a cartoon bird pet after one day (Figure 3.39).

Figure 3.39: The bird was born. Find food and be happy.

There are five states on born bird: idle, dead, 50% growth, 100% growth and travel. The bird was idle at the beginning. It will look for food near his nest and then go eat. After eating, the bird's hungry value will increase. After one hour, the hunger value will decrease by 1. If the hunger value is zero, the bird will go die state.

Dead birds are not really dead. It just lay on the ground without any other movement. When a bird food is present near the nest, the bird will return to idle and go eat.

Idle birds will also play with bird toys randomly. Each play will increase bird's growth value by 1%.

50% grow up birds will fly, but it will still fly around their nests. 100% grow up birds will fly away and send back postcards.

3.3.8 Server

The server is used to collect the player's recognition results. When the player identifies a bird species in the game, the system sends the results to the server. The server is written by Nodejs[20], which is a JavaScript runtime engine and primarily used to build web servers.

The server will listen on port 1200. The server is built on a VPS (Virtual Private Server). The server's IP address is posted on the link:

`https://aixiguo.github.io/BirdGo/newlink.txt`

The game client will first access this link to get the latest server IP address. Then use the server API to communicate with it. The following list shows the server API.

[20] `https://nodejs.org/en/`

1. Management Page
    - This is an admin page that allows to manually get or publish data.
    - Protocal: GET
    - Example URL: `http://172.245.118.139:1200`
2. Hallo Signal
    - Used to test the connection to the server. If the server is running, it will return an ACK signal to the game.
    - Protocal: POST
    - Example URL: http://172.245.118.139:1200/CMD
    - Post Data:"Hallo"
    - Receive Package:{"ACK":true}
3. Report a New Species
    - Used to send a report to the server. The server will log it to the database.
    - Protocal: POST
    - Example URL: http://172.245.118.139:1200/CMD
    - Post Data Format:"Report [Name Latitude Longtitude Possibility Time]"
    - Post Data Example:"Report Goose 51.40312 6.763519 90.1 10/07/2019-12:10"
    - Receive Package:{"ACK":true}
4. Get All Data
    - Used to download the database from the server. The list of recorded bird species is sent back as a json package.
    - Protocal: POST
    - Example URL: http://172.245.118.139:1200/CMD
    - Post Data:"Fetch"
    - Receive Package:{"dat":{json package};"ACK":true}

4 Evaluation

Unlike most computer and video games that require players to play games in front of the computer, Bird Go needs players to go to places where the birds exist, such as forest, parks, countryside. In addition, Bird Go also needs to combine the entertainment of the game and bird finding. Thus, An empirical user study is conducted to evaluate whether Bird Go has a positive impact on these factors. The most important thing is to exam the time spent by the player, the number of recorded birds by players, the player's experiences and entertainment on the game, and the influence of Bird Go on the player's scientific interest. To assess the effectiveness of Bird Go, I divide the main aspects into two tasks. The first task is to measure the user's overall gaming experience with Bird Go, especially with regard to the subjective feelings about the game difficulty and overall satisfaction. The second task is to focus on the quality of the recordings from players, especially the number and authenticity of recordings.

4.1 Objectives

The objectives are :

- to exam the overall user satisfaction of Bird Go.
- to evaluate whether Bird Go has a positive effect on the birds searching and birdsong recording.
- to compare the time and frequencies of the player's stay in areas where birds may appear and the amount and frequency of birds found.
- to analyse the collected data from players.

4.2 Methods

4.2.1 Participants and materials

There are 10 (6 female) participants who have not played Bird Go were invited to play the game within a week. They have age ranging from 22 to 28(M = 26, SD=2.46). They were all full-time master students and were not employed. They also live in different locations(in Duisburg) and have different physical activity level.

Since this game is only available for the android platform, therefore, volunteers are recommended to use Samsung s9 or similar equipment. Volunteers can play Bird Go at home and outdoor. But because the recognition algorithm is designed for birds in Germany. Therefore, the range of player movement is limited in Germany. The device should have the ability to access the Internet while volunteers are playing the game. Because when the player successfully recognizes the bird, the recognition result will be sent to the server through the Internet.

Since there were no tutorials in the game, participants need to see an introduction demo about how to play the game before they start to play. The demo includes instructions on UI functions, how to find treasure chests, how to record bird songs and identify bird species, how to use bird puzzles to unlock bird images to get bird eggs and how to feed cartoon bird pets. The players' task is to do their best to get more bird song recordings and try to feed one or more cartoon bird pets within a week. At the end of the week, volunteers needed to fill out a questionnaire.

The questionnaire consists of four parts: the first part questions about basic personal information. In the second part, a System Usability Scale (SUS)(Brooke et al., 1996) is applied to evaluate the availability of Bird Go. The third part is to evaluate the player's experience using the User Experience Questionnaire (UEQ) mentioned by Laugwitz et al. (2008). The last part contains specific questions made by myself according to the approach about the game and bird song recognition.

The first part of the questionnaire will ask about the players' basic personal information such as age, education, a background of playing similar games, knowledge about the birds and current live location. These questions help to understand the background of players and their possible activity range.

The second part contains 10 Likert-style questions from SUS. For example, the first question is "I think that I would like to use this system frequently". For each question, the player gives a score between 1 and 5, which represents the player's degree of agreement to the sentence. After that, SUS calculate a SUS score for each participant. And an average value is then calculated from all the SUS scores of participants. And this value represents the overall system usability.

The next part is the UEQ questionnaire. Similar to SUS, UEQ provides users with 8 Likert-style questions of user experience descriptions. For example, the first question is "obstructive/supportive", and the user selects a number from 1 to 7, which means that the user prefers the first description (obstructive) or the last description (supportive). These eight sets of description can be divided into two groups. The first 1-4 sets are group one and represent pragmatic quality. The rest sets are group two and describe hedonic quality.

The last part includes questions which are not covered in the three sections above. This part focuses on the effect of reward mechanisms on encouraging players to look for birds and whether Bird Go can balance entertainment and scientific research. The questions in this part are such as the frequency in walking/ jogging to find treasure chests and how many birds are found in one day. This part also includes some questions from a study (Craig et al., 2003), which researched the short form of the International Physical Activity Questionnaire (IPAQ).

And these questions can be used for measuring three specific types of activities, including strenuous exercise, moderate-intensity exercise, and walking in the past seven days.

The whole questionnaire can be found in the appendix A.

4.3 Results of Questionnaire

4.3.1 Part 1: Personal Questions

There are 10 (6 female) participants. They have age ranging from 22 to 28(M = 26, SD=2.46). And they are all graduate students. Only 2 of them said they had played a similar game with Bird Go. And most of them (80%) said that they know a little about birds, such as their species name, sound, habit. Only two students said that they known a lot about birds.

For the questions about participants' residence, six students provided their addresses, four of them lived near the university, one lived near Duisburg Central Station, and one participant lived in a village far from the city.

4.3.2 Part 2: System Usability

The average score of SUS is 78.8 (Max:92.5, Min:75, SD:6.25). Figure 4.1 shows the score details for each participant. In this figure, vertical bars represent scores of participants. These scores represent the system usability value for each participant.

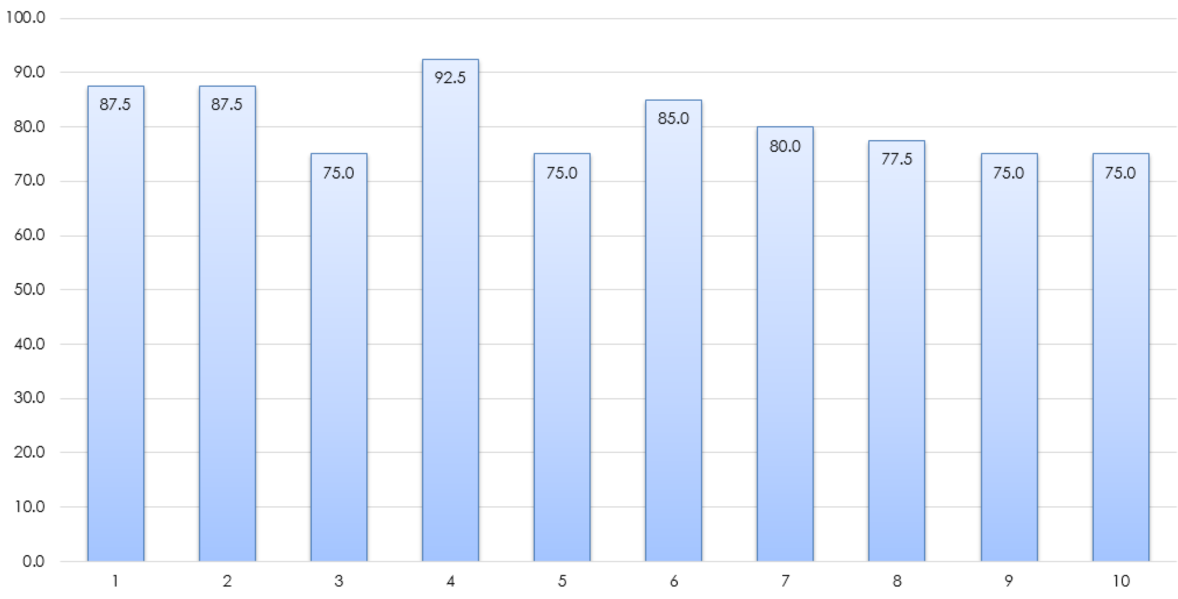

Figure 4.1: Results of SUS

4.3.3 Part 3: User Experience

For USQ, I got an average value of Pragmatic Quality as 1.575(Good) and the average value of Hedonic Quality as 2.4(Excellent). And the overall average value is 1.99(Excellent). Figure 4.2 shows the average value for each questions. The blue horizontal bars represent the average values of Pragmatic Quality, and the orange horizontal bars indicate the average values of Hedonic Quality questions.

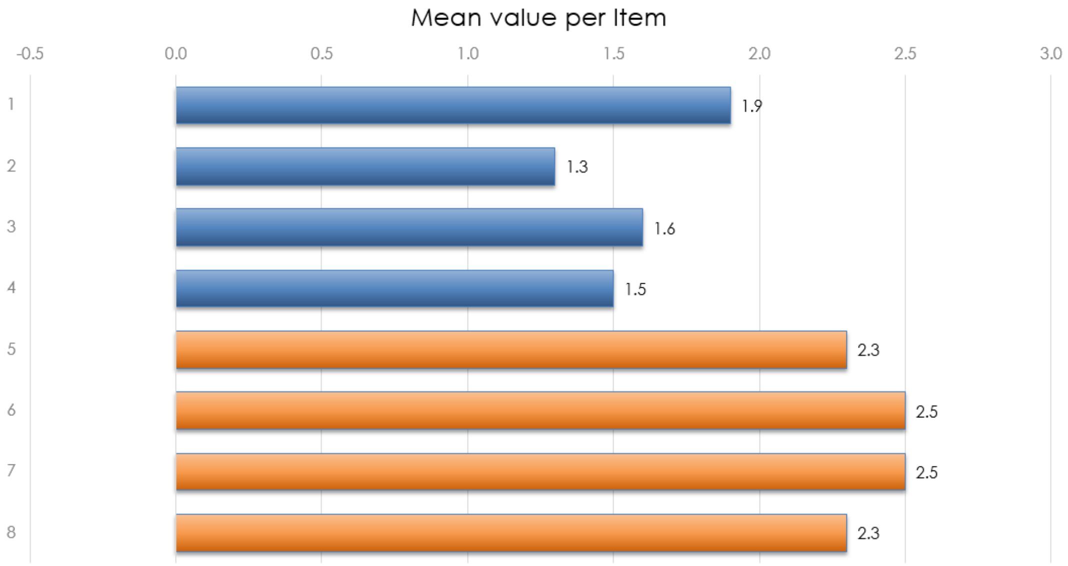

Figure 4.2: Results of UEQ

4.3.4 Part 4: Questions About the Game

The following tables (Table 4.1, Table 4.2) show the results of these part. Only question 4.5 is a multiple-choice question, the rest are single-choice questions.

Questions	Volunteers(%)
4.1 Time spent each day in playing Bird Go	
$\leqslant$ 15 min	10
16–30 min	80
31–60 min	10
>1h	0
4.2 Frequency in walking/ jogging to find treasure chest	
Never	0
Rarely	0
Sometimes	30
Most of the time	70

Table 4.1: Questionnaire

Questions	Volunteers(%)
4.3 How many birds do you find in one day?	
0-5	70
5-10	20
10-15	10
15-20	0
> 20	0
4.4 The bird prediction from birdsong is ?	
bosh, nonsense, total false.	0
I have no idea.	30
Sometime accurate sometime not.	50
Most are right.	10
Exactly.	10
4.5 The bird song you recorded comes from? (Multiple choice)	
Forest, Park, Garden.	90
Near my house.	80
Zoo.	10
From TV/audio/internet.	20
Noise(human voice, traffic sound, other animals etc).	50
4.6 How many stars do you collected in one day?	
0-500	30
500-1000	60
1000-2000	10
>2000	0
4.7 Getting a cartoon bird pet is?	
too hard and I don't know how.	10
not easy but ok.	20
ok, I can get them sometime.	20
great, and I love them.	50
easy and I get too much of them.	0
4.8 How many bird pets do you have?	
0.	10
1-5.	80
6-10.	0
>10.	10
4.9 Do you learn more bird knowledge through Bird Go?	
Not at all.	0
A little.	30
I have learnt some.	50
A lot.	20
4.10 Do you want to continue to play the Bird Go? And why?	
No, It is boring.	0
Yes, I want to continue my contribution for bird scientist.	40
Yes, It is really fun.	60

Table 4.2: Questionnaire(continue)

4.4 Results of Mapping of Bird

Figure 4.3 shows the mapping of songbirds. During volunteers played Bird Go, The server received 207 reports, including 25 species of birds from players. These reports include recognized bird names, location information (latitude and longitude), prediction accuracy, and recording time. Based on this data, the server drew a bird distribution map. On this map, a circle on the map represents a recording of birdsong. Different bird species have different circle colour. The list on the right shows the species name of the bird and its circle colour.

Most of the recording took place in 3 areas, University and Zoo, Sportspark and Wolfsee. The rest occurred mainly in parks or gardens. Only small parts of bird reports were in the city centre and residential areas.

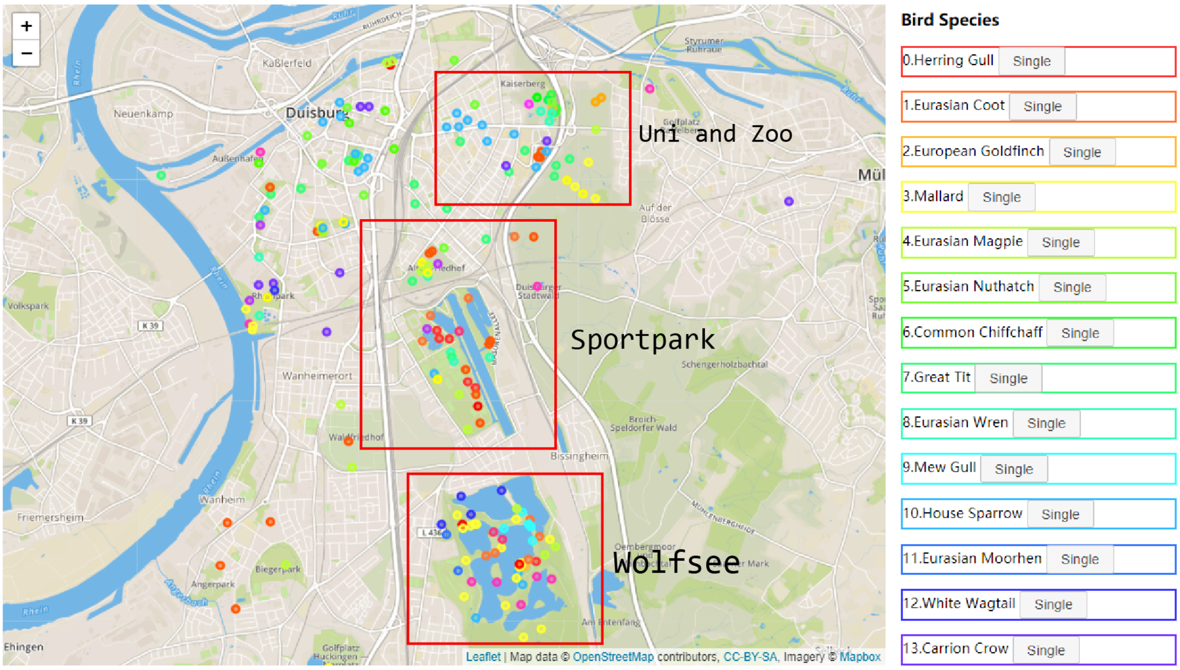

Figure 4.3: Bird Distribution Map. A circle represents a record. The list on the right shows the name of the bird and the corresponding color (14/25 are shown)

4.5 Analysis

Results form Questionnaire

From part 2 of the questionnaire, the usability of Bird Go was evaluated as good with a SUS-score of 78.8. This score means that the game still needs improvement. Tutorials or more help text should be added to the game so that players can better understand the game logic.

In the third part, UEQ's hedonic quality score is relatively higher than the pragmatic quality. This means that most players think Bird Go is fun and creative. However, a low pragmatic quality score indicates that Bird Go is still too complicated or confusing for players. Players need more help to complete the tasks in the game.

Some results can be obtained from the fourth part. First, the time of playing Bird Go of most players is between 16-30 min. And most players said that they only play when they are outdoors. Second, when the player plays Bird Go, the treasure mechanism can effectively attract the player's interest. And most players enjoy the process of treasure searching. However, some players say that they rarely do treasure-hunting because they are not familiar with the game and don't know how to find them.

Next, most players can find 0-5 birds. But they don't trust these predictions very much. Some of them said they got birds from the wrong sources (human voices, traffic noise, and audio on TV or the Internet). Some of them went to the zoo and had many recordings and bird rewards.

In other hand, the mechanism of the cartoon bird pet has successfully attracted the attention and interest of the player. Most of them like bird pets. They like this virtual pet and want to continue feeding them. Few players have get a lot of bird pets and are tired of finding new birds. This means that this bird pet mechanism also has the disadvantage that it may lead the player away from the Bird Go's original goal (finding bird). It should be improved in the future.

Another important point is that most participants say they can learn about birds. Some of them said they learned a lot of bird information. This shows that Bird Go can improve the player's understanding of the bird, thus guiding the participants to pay more attention to bird research and protection.

In the end, most players want to continue playing Bird Go, and their reason is that it is very interesting. Some of them also said they want to continue because they want to contribute to bird research. In a word, this survey proves that Bird Go can encourage players to find birds. Bird Go can also combine the entertainment and scientist task. Players get a lot of fun from it and complete the bird song searching simultaneously. During the game, however, the bird prediction algorithm needs to be improved because more players didn't completely trust the results.

Results form Database

At the same time, I analyzed the database of collected bird information. The results of mapping songbirds are displayed on the server website (Figure 4.3).

In order to search for birds, it can be seen from the map that most players first choose the nearest natural park. At the same time, the park has also greatly attracted the attention of the players. Reports in these areas are more frequently. This indicates that Bird Go has a positive effect on the birds searching.

However, some participants went to the zoo to record more bird calls. The server received a lot of reports from the zoo area. That is not what an ornithologist wants. Also, some recordings may be fake because they come from the city centre. And players want to get recording rewards even at home. Therefore they record Birdsong from TV or audio. That is, however, a disturbance for the correct recordings. How to filter out the fake recordings should also be a challenge for the further work of Bird Go.

On the other hand, the mapping of songbirds can help analyze the species distribution of the bird. After clicking the button to the right of the species name (right part of Figure 4.3), the distribution of that species is displayed on the map. For example, figure 4.4 shows the distribution of the Graylag Goose.

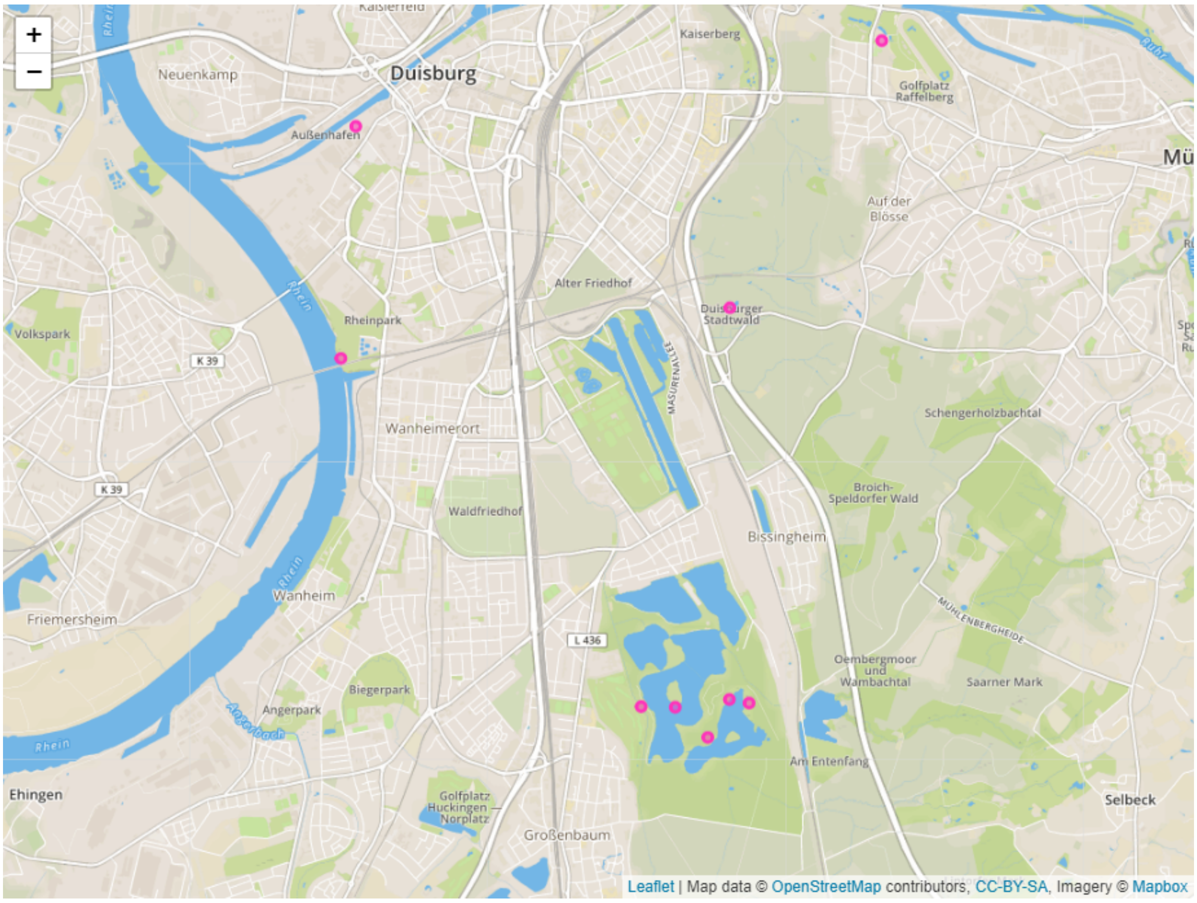

Figure 4.4: Distribution of the Graylag Goose

5 Conclusion and Futher Work

This thesis studied and designed a citizen science project using games with a purpose. The game called BirdGo is used to encourage people to go out and enter places where birds often live. The players are also encouraged to help bird scientists to observe and record songbirds.

5.1 Summary

This thesis mainly focused on and answered the following questions:

- How to recognize bird species from birdsong?
- How the games with a purpose could encourage players to do the scientific research (here is to find and record bird)?
- How to combine the fun and scientific research in the game with purpose?

The main findings are: First, the Deep Residual Neural Network (ResNet) can be used to identify bird species from birdsong. And it has higher accuracy than general neural networks such as VGG. Even with reduced parameters, ResNet's model can still maintain relatively high precision for large-scale data. And after optimization, the model can be used in the game engine Unity.

Secondly, through questionnaires, I found that the rewards mechanism is the most important factor that motivates players to help with scientific research. Also, the cute/interesting game elements and designs can give players more attraction. The goals in the game can greatly encourage the player to do the special activity. Players are excited when he/she gets rewards from the game.

Finally, using the advantage that the game can encourage the player to find and record bird songs. The scientific research can be combined with the game. Players can help scientist and also have a lot of fun through it. More important is that the player has also learned a lot of knowledge by taking part in scientific research. They are willing to continue these scientist research since they start to believe that this research is meaningful and interesting.

5.2 Further work

In general, this thesis presents a baseline for mapping of songbirds using a citizen science game. It has been proved that it can encourage players to find birds, and it has the ability to recognize bird species from the player's recordings. However, it still has some shortcomings. First, the recognition algorithm cannot effectively separate the interferences. Noise such as human voice or traffic noise can also be predicted as a birdsong. The model has great effect only when there is only one birdsong in the predicted time interval (one second). When there are multiple categories of birds exist in one interval, the model can't separate them.

For this issue, I recommend using other CNN models that are more suitable for mobile devices, such as MobileNet. On the other hand, it's best to use a powerful server for prediction algorithm, and the game client needs to send the audio to the server. However, this will also cause a lot of data transfer over the Internet and delay problem. Another key impact of predictive accuracy is noise reduction. Different noise reduction methods have different impacts on the final prediction. The method mentioned in this thesis still needs to be improved.

Second, game design has some disadvantages, such as performance issues when the player moves too fast (on a car or train). Players can easily perform some cheating actions, such as recording bird songs on TV. They can also enter the zoo instead of the forest to get more recordings of birds. The game also requires a tutorial because the player needs to understand the game firstly.

There are some strategies that may be helpful for these problems. For example, it is recommended to consider the player's location. When the player is located in a building or city centre where birds rarely appear, a warning is displayed to the player that the recording function is disabled because of the false location. In addition, the area where the recording function is disabled should also cover Zoo or bird museum.

In terms of games, 3D scenes should be optimized because the loading time is too long. In addition to using Archimatix to build a custom 3D world, there are also some better tools for building 3D virtual worlds based on maps, such as MapBox. Loading a real map in the game may also help players' orientation.

In addition, birds should be researched on a wider and longer-term. It is necessary to expand the number of players and perform long-term data collection. The server also needs to implement a data classification function and more functions for analysis of collected data. It is also necessary to cooperate with bird scientists in order to help them analysis bird information and identify bird species.

Appendix

A Questionnaire

Questionnaire of Bird Go

Part 1: Personal Questions (Single Choice)

1.1 Your gender is?
- A. Male.
- B. Female.

1.2 Your age is ____.

1.3 Your education is:
- A. Undergraduate student.
- B. Graduate student.
- C. PhD.

1.4 Do you ever play a game similar with Bird Go?
- A. No.
- B. Yes.

1.5 Do you know about birds? Such as their species name, sound, habit?
- A. No.
- B. A little.
- C. A lot.
- D. I am bird expert.

1.6 Where do you live?
- A. It is a secret.
- B. ____.

Part 2: System Usability

	Strongly Disagree				Strongly Agree
1. I think that I would like to use this system frequently.	☐	☐	☐	☐	☐
2. I found the system unnecessarily complex.	☐	☐	☐	☐	☐
3. I thought the system was easy to use.	☐	☐	☐	☐	☐
4. I think that I would need the support of a technical person to be able to use this system.	☐	☐	☐	☐	☐
5. I found the various functions in this system were well integrated.	☐	☐	☐	☐	☐
6. I thought there was too much inconsistency in this system.	☐	☐	☐	☐	☐
7. I would imagine that most people would learn to use this system very quickly.	☐	☐	☐	☐	☐
8. I found the system very cumbersome to use.	☐	☐	☐	☐	☐
9. I felt very confident using the system.	☐	☐	☐	☐	☐
10. I needed to learn a lot of things before I could get going with this system.	☐	☐	☐	☐	☐

Part 3: User Experience

Choose the square closest to your feel about Bird Go								
obstructive	☐	☐	☐	☐	☐	☐	☐	**supportive**
complicated	☐	☐	☐	☐	☐	☐	☐	**easy**
inefficient	☐	☐	☐	☐	☐	☐	☐	**efficient**
confusing	☐	☐	☐	☐	☐	☐	☐	**clear**
boring	☐	☐	☐	☐	☐	☐	☐	**exiting**
not interesting	☐	☐	☐	☐	☐	☐	☐	**interesting**
conventional	☐	☐	☐	☐	☐	☐	☐	**Inventive**
usual	☐	☐	☐	☐	☐	☐	☐	**leading edge**

Part 4: About the Game

4.1 Time spent each day in playing BirdGo

A. 15 min – 30 min
B. 31 – 60 min
C. 1h to 2 h
D. > 2 h

4.2 Frequency in walking/ jogging to find treasure chest

A. Never
B. Rarely
C. Sometimes
D. Most of the time

4.3 How many birds do you find in one day?

A. 0-5
B. 5-10
C. 10-15
D. 15-20
E. > 20

4.4 The bird prediction from birdsong is?

A. bosh, nonsense, total false.
B. I have no idea.
C. Sometime accurate sometime not.
D. Most are right.
E. Exactly.

4.5 The bird song you recorded comes from? (multiple choices)

A. Forest, Park, Garden.
B. Near my house.
C. Zoo.
D. From TV/audio/internet.
E. Noise(human voice, traffic sound, other animals).

4.6 How many stars do you collected in one day?

A. 0-500.
B. 500-1000.
C. 1000-2000.
D. >2000.

4.7 What do you think about getting a cartoon bird pet is?

A. too hard and I don't know how.
B. not easy.
C. ok, I can get them sometime.
D. great, and I love them.

E. easy and I get too much of them.

4.8 How many bird pets do you have?

A. 0.
B. 1-5.
C. 6-10.
D. >10.

4.9 Do you learn more bird knowledge through Bird Go?

A. Not at all.
B. A little.
C. I have learnt some.
D. A lot.

4.10 Do you want to continue to play the Bird Go? And why?

A. No, It is boring
B. Yes, I want to continue my contribution for bird scientist.
C. Yes, It is really fun.
D. ____(you can write your own answer).

Do you have any suggestions for the Bird Go?

____

____

____

References

Able, K. P. (1999). *Gatherings of angels: migrating birds and their ecology*. Cornell University Press.

Alexander Hars, S. O. (2002). Working for free? motivations for participating in open-source projects. *International Journal of Electronic Commerce*, *6*(3), 25–39.

Althoff, T., White, R. W., & Horvitz, E. (2016). Influence of pokémon go on physical activity: study and implications. *Journal of medical Internet research*, *18*(12).

Andersen, E., O'Rourke, E., Liu, Y.-E., Snider, R., Lowdermilk, J., Truong, D., ... Popovic, Z. (2012). The impact of tutorials on games of varying complexity. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 59–68).

Barrow, M. (2000). *A passion for birds: American ornithology after audubon*. Princeton University Press.

Batson, C. D., Ahmad, N., & Tsang, J.-A. (2002). Four motives for community involvement. *Journal of social issues*, *58*(3), 429–445.

Bogost, I. (2011). Persuasive games: Exploitationware, gamasutra: The art & business of making games. *URL: http://www. gamasutra. com/view/feature/134735/persuasive_games_ exploitationware. php, 3rd May*.

Brooke, J., et al. (1996). Sus-a quick and dirty usability scale. *Usability evaluation in industry*, *189*(194), 4–7.

Buyuksalih, I., Bayburt, S., Buyuksalih, G., Baskaraca, A., Karim, H., & Rahman, A. A. (2017). 3d modelling and visualization based on the unity game engine–advantages and challenges. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *4*, 161.

Choi, K., Joo, D., & Kim, J. (2017). *Kapre: On-gpu audio preprocessing layers for a quick implementation of deep neural network models with keras.*

Clary, E. G., Snyder, M., Ridge, R. D., Copeland, J., Stukas, A. A., Haugen, J., & Miene, P. (1998). Understanding and assessing the motivations of volunteers: a functional approach. *Journal of personality and social psychology*, *74*(6), 1516.

Cohn, J. P. (2008). Citizen science: Can volunteers do real research? *BioScience*, *58*(3), 192–197.

Cooper, S., Khatib, F., Treuille, A., Barbero, J., Lee, J., Beenen, M., ... others (2010). Predicting protein structures with a multiplayer online game. *Nature*, *466*(7307), 756.

Craig, C. L., Marshall, A. L., Sjöström, M., Bauman, A. E., Booth, M. L., Ainsworth, B. E., … others (2003). International physical activity questionnaire: 12-country reliability and validity. *Medicine & science in sports & exercise*, *35*(8), 1381–1395.

Curtis, V. (2015). Motivation to participate in an online citizen science game: A study of foldit. *Science Communication*, *37*(6), 723–746.

Garris, R., Ahlers, R., & Driskell, J. E. (2002). Games, motivation, and learning: A research and practice model. *Simulation & Gaming*, *33*(4), 441-467. Retrieved from `https://doi.org/10.1177/1046878102238607` doi: 10.1177/1046878102238607

Gura, T. (2013). Citizen science: amateur experts. *Nature*, *496*(7444), 259–261.

He, K., Zhang, X., Ren, S., & Sun, J. (2016a). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778).

He, K., Zhang, X., Ren, S., & Sun, J. (2016b, Jun). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Retrieved from `http://dx.doi.org/10.1109/CVPR.2016.90` doi: 10.1109/cvpr.2016.90

He, K., Zhang, X., Ren, S., & Sun, J. (2016c). Identity mappings in deep residual networks. In *European conference on computer vision* (pp. 630–645).

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-r., Jaitly, N., … others (2012). Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, *29*.

Holohan, A., & Garg, A. (2005). Collaboration online: The example of distributed computing. *Journal of Computer-Mediated Communication*, *10*(4), JCMC10415.

Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J., & Weinberger, K. (n.d.). Snapshot ensembles: Train 1, get m for free. arxiv 2017. *arXiv preprint arXiv:1704.00109*.

Huang, G., Liu, Z., Maaten, L. v. d., & Weinberger, K. Q. (2017, Jul). Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Retrieved from `http://dx.doi.org/10.1109/cvpr.2017.243` doi: 10.1109/cvpr.2017.243

Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2261-2269.

Kahl, S., Wilhelm-Stein, T., Klinck, H., Kowerko, D., & Eibl, M. (2018). Recognizing birds from sound-the 2018 birdclef baseline system. *arXiv preprint arXiv:1804.07177*.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kogan, J. A., & Margoliash, D. (1998). Automated recognition of bird song elements from continuous recordings using dynamic time warping and hidden markov models: A comparative study. *The Journal of the Acoustical Society of America*, *103*(4), 2185–2196.

Koumura, T., & Okanoya, K. (2016). Automatic recognition of element classes and boundaries in the birdsong with variable sequences. *PloS one*, *11*(7), e0159188.

Krebs, V. (2010). Motivations of cybervolunteers in an applied distributed computing environment: Malariacontrol. net as an example. *First Monday*, *15*(2).

Laugwitz, B., Held, T., & Schrepp, M. (2008). Construction and evaluation of a user experience questionnaire. In *Symposium of the austrian hci and usability engineering group* (pp. 63–76).

Newman, G., Graham, J., Crall, A., & Laituri, M. (2011). The art and science of multi-scale citizen science support. *Ecological Informatics*, *6*(3-4), 217–227.

Ornithology. (2006). W.H. Freeman.

Prestopnik, N. R., & Tang, J. (2015). Points, stories, worlds, and diegesis: Comparing player experiences in two citizen science games. *Computers in Human Behavior*, *52*, 492–506.

Redmon, J. (2013–2016). *Darknet: Open source neural networks in c.* `http://pjreddie.com/darknet/`.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 779–788).

Reed, J., Raddick, M. J., Lardner, A., & Carney, K. (2013). An exploratory factor analysis of motivations for participating in zooniverse, a collection of virtual citizen science projects. In *2013 46th hawaii international conference on system sciences* (pp. 610–619).

Rotman, D., Preece, J., Hammock, J., Procita, K., Hansen, D., Parr, C., … Jacobs, D. (2012). Dynamic changes in motivation in collaborative citizen-science projects. In *Proceedings of the acm 2012 conference on computer supported cooperative work* (pp. 217–226).

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., … Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, *115*(3), 211-252. doi: 10.1007/s11263-015-0816-y

Ryan, R. M., & Deci, E. L. (2000). Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, *25*(1), 54–67.

Ryan, R. M., & Deci, E. L. (2009). Promoting self-determined school engagement. *Handbook of motivation at school*, 171–195.

Schenk, E., & Guittard, C. (2009, 01). Crowdsourcing: What can be outsourced to the crowd, and why ?

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sullivan, B. L., Wood, C. L., Iliff, M. J., Bonney, R. E., Fink, D., & Kelling, S. (2009). ebird: A citizen-based bird observation network in the biological sciences. *Biological Conservation*, *142*(10), 2282–2292.

Von Ahn, L. (2006). Games with a purpose. *Computer*, *39*(6), 92–94.

Zagoruyko, S., & Komodakis, N. (2016a). Wide residual networks. *CoRR*, *abs/1605.07146*.

Zagoruyko, S., & Komodakis, N. (2016b). Wide residual networks. *Procedings of the British Machine Vision Conference 2016*. Retrieved from `http://dx.doi.org/10.5244/c.30.87` doi: 10.5244/c.30.87