ESCAPE COMPUTER

SOFTWARE ENGINEERING AND GROUP PROJECT

Team Cullens

University of BRISTOL

Abstract

The aim of the project is to develop a game in terms of computer science in order to help young people to have a better understanding of computer knowledge and inspire their interest.

As we all know, computer technology is more and more popular with students. There are a lot of software around our lives which change our lives. For instance, WhatsApp change communication between us. Thus, it is necessary for students to get some computer knowledge actively aiming for they can develop their own software in the future.

Furthermore, the type of games is various, such as Action-adventure games, Platforming games, Alternate reality games, Sports games, Escape games and so on. The project initial step is to develop an escape game by using some computer knowledge to escape each room. Based on this, the target audience for the project is a teenager aged between 7 and 15. Once there was a concrete concept, a set of team objectives were agreed upon:

- Creating many computer components used in the room in the game.
- Creating many rooms
- Creating the strategy of every room
- A beautiful and attractive interface and sounds
- Implement the game

After the selection of the main objectives, we discuss and implement the development strategy. For the smooth progress of the whole project and the project can be completed within the specified time, a work plan was discussed and agree upon. More specifically, weekly meetings were scheduled, and it is beneficial for the group members to put on their question and help each other. In fact, quick and effective communication is the

basis of the group developing.

To achieve the aim, the combination of various software should be used. Unfortunately, the group members are all lack of experience in developing software. However, the group members all have confidence in the game developing through a period of mutual learning. Finally, a game named “Escape Computer” which is attractive and educational with storyline was our main outcome.

Dedication and Acknowledgements

Team Cullens would like to thank a number of people who helped in the development of this game.

Ian Holyer, for always take questions and offer advice.

Classmates, for helping us choose the final idea

Students, for helping us test games and give feedback.

-----Team Cullens

Mark Distribution Declaration

We, the members of Team Cullens, hereby declare that an equal distribution of marks has been agreed upon and that the work contained in this report was written by the group members for the sole purpose of this module.

Lin Qiu ---21/05/19

YiNi Zhang--21/05/19

WenBin Guo---21/05/19

Lu Sun--21/05/19

JiaXuan Yin --21/05/19

Michael Zhang---21/05/19

Content

List of Tables

1 Game Development Strategy

1.1 Software Development Strategy

The project was structured around an agile development strategy.

Agile development is by no means a one-size-fits-all standardized process. It is more of a process optimization concept of self-adaptation and self-optimization. In game development, it is more important to reflect people's grasp of the quality of the game itself, rather than the review of various documents, which is the most significant difference from traditional software development. Therefore, there is no best process, as long as it is appropriate and can continue to optimize the process is good.

Secondly, we need to have a grasp of the game. Understand what the game contains; What is the core gameplay; What are the main patterns; What is the picture representation; What are the sound effect; What are the stories, how do they fit together, how do they fit together into a game. In different game types, in what proportion these elements are combined, which is more important.

Next, the development process, in essence, is about analyzing requirements, making plans, and starting development. Perhaps the cycle is divided into smaller cycles for iteration. But these small iterations are still waterfall development, no doubt about it. The difference is that in such a short period, our development philosophy and guidelines are guided by agile development philosophy. We are more version oriented, more communication-oriented, and more result oriented, rather than predefined development processes (pre-defined). Every feature, every change has a version that immediately verifies and experiences the functionality. We will have a much more direct and efficient grasp of game changes. Most revisions to the direction of the game happen during development and not just at the beginning or end of the cycle.

1.2 Team Management

Our group is democratic and decentralized. Instead of having a fixed leader, teams assign tasks to each person at different stages. Decisions are made jointly by groups through consultation, and communication between group members is horizontal. Let people with specific tools or skills solve problems related to them so that the whole team can work well together and develop more efficiently.

Generally, we have face-to-face meetings once or twice a week. We usually communicate with each other in the WhatsApp group. At each meeting, we would summarize the current project progress and assign tasks to be completed this week.

1.3 Development Tools

At the beginning of game development, we first listed the optional languages, graphics libraries, and software. Our choice is based on the following points:

- How does our game work on different desktop platforms?
- How to make a game efficiently in a limited time?
- Are we building on technology we already have, or are we building on accessible game technology?
- Shall we use the existing online material resources or draw our materials?

1.3.1 Languages and Libraries

Based on the choices provided in the course notes and our personal experience, we considered the following three languages and corresponding graphics libraries: Java and JavaFX, C#, and Unity.

1 Java and JavaFX

JavaFX is a powerful collection of graphical and multimedia processing toolkits that allow developers to design, create, test, debug, and deploy rich client programs that are cross-platform like Java.

Advantages:

- JavaFX is a part of our learning CS Cours, and we have already gained some experience.
- The default skin looks better, and the components are more abundant.
- Rich multimedia support for playing music and video smoothly.
- Scene Builder can be used for visual interface design, and the code is automatically generated. It can improve development efficiency.
- Similar to the Web, JavaFX can also be developed using 3-tier architecture.

  HTML -- FXML: structure

  CSS --the CSS: appearance

  JavaScript -- Controller. Java

Disadvantages:

- It takes more code than other languages to implement the same action or setup.
- There are many libraries that we need to learn in Java if we want to develop more useful features.

2 C# and Unity

Unity is the 3D engine of choice for most game development teams today, and it works well in 2D.

Advantages:

- Unity has a lot of proper official documentation and tutorials that are helpful and efficient for game development later. The assets store is also reportedly great when compared to other platforms. Tech support for Unity is practical. The tech support members are also highly skilled, and they can be relied upon in case of any technical issues.

- Unity is excellent for cross-platform development. It is time-saving and demands lots of efforts for developing native apps. When using cross-development platforms, a single script can be compiled and used for many platforms.
- Due to the advantages of the framework and architecture, Unity's games can significantly avoid crashes and flashbacks. Because the code is written in C# and is component structured, even if there are mistakes, they won't affect the flow of the system.
- Convenient resource management system. With Unity, you don't have to maintain multiple resources, such as raw and packaged resources. You only need one resource, and its parameters can be set in Unity.
- It is also easier to use compared to many other technologies. Many functions are systematic and easy to implement. For example, features such as physics, collision detection, navigation pathfinding, and scene management would require a complex code, but in Unity, a few buttons and a few steps can do the trick.

Disadvantages:

- None of our team members had learned unity, and we needed to spend extra time learning the new language.
- The most obvious drawback is that it does not allow us to start from a foundation or a template. You must implement all the details. You will have to start from scratch with each game.
- The documentation of several features is entirely out of date, and in some cases, it is altogether non-existent.
- No Linux version.

1.3.2 Graphics Software

There are two kinds of graphics software we can choose, open source software and proprietary software. We compare several major open source programs with their proprietary counterparts in the following section.

1. Gimp and Photoshop

(1) Gimp

GIMP is a free, open source image editing software initially developed for Linux users to process photos. Today, GIMP has released Windows and MacOS versions, attracting more users.

Pros:

- GIMP is open source, free software that anyone can download.
- One of GIMP's most prominent features is lightweight. The software itself is only 200MB in size and running it saves a lot of computing resources. Compared with similar software such as Photoshop, GIMP requires lower configuration of the computer.
- Support for PSD files. You can import PSD files into GIMP for editing, and the imported data will still be layered.
- Users can customize the interface by dragging the tool window.

Cons:

- The quality of the tools is not high. Most of GIMP's tools are not as good as Photoshop, especially when it comes to intelligent processing. For instance, Photoshop's patch tool is much more powerful than GIMP's.
- GIMP is open source software, so updates are slow, and bug feedback is delayed.
- PSD file support is limited. Sometimes layer effects can be problematic when importing PSD files into GIMP.

(2) Photoshop

Nearly all business-related individuals and businesses have used Photoshop. From a professional image processing perspective, no software can match the popularity of Photoshop, which is understandable: Photoshop can provide advanced functionality that no other software can provide.

Pros:

- Most of our team members have some experience with Photoshop, so it will be time-saving for us to use Photoshop rather than learning new software.
- Photoshop supports almost all file formats.
- Adobe has a dedicated team developing and applying new technologies to Photoshop.
- If you choose to subscribe to Adobe's software, you can use CC-provided cloud storage, ranging from 20GB to 1TB.

Cons:

- Photoshop is not free software, and it is relatively expensive.
- Photoshop uses a lot of system resources to avoid tool delays and can be very resource intensive when dealing with large or complicated graphics.

2. Inkscape and Illustrator

(1) Inkscape

Inkscape is an open source vector graphics editor similar to Illustrator, Freehand, CorelDraw, Xara X, and others.

Pros:

- Inkscape is free, with unlimited download usage and even translation to modify it.
- Inkscape supports Windows as well as Linux and Mac versions so you can use it on any system you use.
- Inkscape is made by volunteers who are passionate about their work and put principles and values over profit. This makes for a supportive grassroots community that feels good to be a part.

Cons:

- No ability to output files in a CMYK color format, making Inkscape, not a very good choice for designing anything needing to be printed.

- Users will regularly experience bugs and glitches.

(2) Illustrator

Illustrator is a vector graphics application by Adobe. It is widely recognized as the industry standard tool for vector design and is used by the overwhelming majority of studios, agencies, and professional designers.

Pros:

- All of Adobe suite's applications communicate with each other. Vector graphics created with Illustrator can very easily be imported into After Effects, Premiere Pro, Photoshop and worked with further.
- There are many tutorials on the internet.

Cons:

- Adobe software is sold as a service, meaning it's an ongoing monthly expense.
- Requires high-performance hardware in order to run well.
- If you want to use the Adobe suite, you'll have to do so on Windows or Mac.

1.3.3 Other Useful Tools

- Premiere

Premiere Pro CC is the continually evolving professional video editing program from Adobe. It's the sum of decades of design, feedback, and refinement. The team had used GIMP before in other modules, which made it even more appealing. Thus, it will be useful to create a transition video for a few seconds and put it in the start or end part.

- Microsoft PowerPoint

The primary function of PowerPoint is to display various text, graphics, charts, sounds, and other media information in the form of pictures. Each picture is called a slide, and it provides multimedia technology to make the presentation sound and look good, with images and texts, and it can show the created content in a variety of ways.

What's more, its drawing function is very powerful. All kinds of line shapes have templates that can be used directly, and each figure has similar effects that can be set, such as shadow and background color. In addition, it can be quickly exported to PNG format.

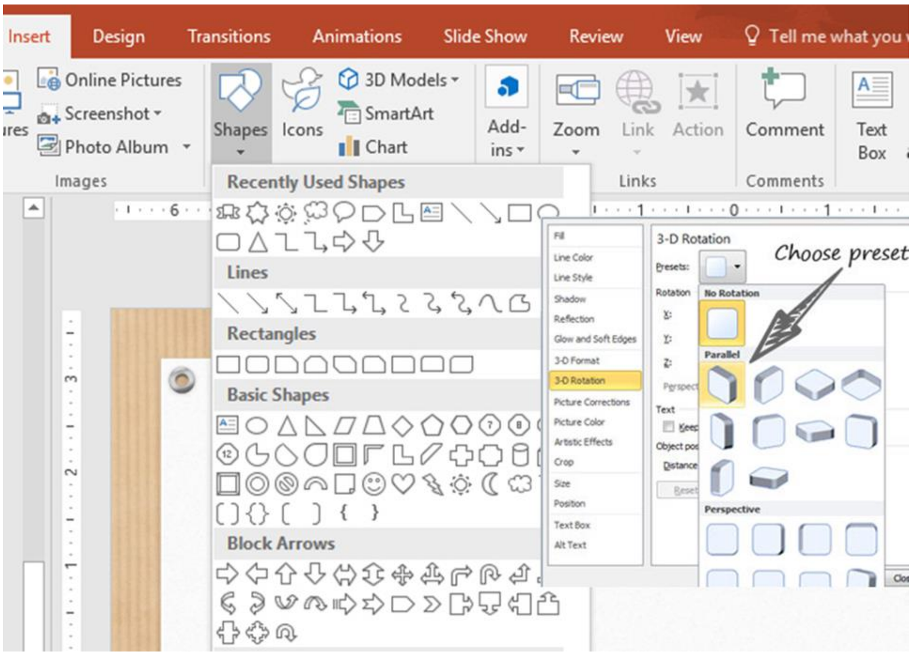


Figure 1 Microsoft PowerPoint Shapes

- Git and GitHub

Git was used for version control of our project. It is an open source distributed version control system that can efficiently and efficiently handle from very small to very large project version management. GitHub was used for cloud storage. It provides a nice and simple way to have their code and version history available online, regardless of what happens to their local machine.

- Google Drive

It is like a server or place where you can store all your files and data into it safely. Users can store all your files such as videos, photos, Google Docs, PDFs, etc.

One of its features is that users can share files or folders with anyone and choose whether the sharing object can view, edit, or comment on a user's files.

1.3.4 Conclusion

Considering all the advantages and disadvantages above, our team finally made the decision that we would use C# and unity, Photoshop, Illustrator, Premiere, and Microsoft PowerPoint to develop our game.

While we need to learn how to use unity and C#, it's worth the cost. Unity is very handy for game development. And the version that we choose to use is Unity2017.

Also, our group decided to hand-draw all the material from online sources, which would take a lot of time. Since our game required a lot of graphic elements for design, we decided to use Photoshop and Illustrator, which we already had some experience.

Besides, the primary platform our team used was Windows, so it was convenient and efficient to use Microsoft PowerPoint to design small components and quickly export them to PNG format.

Finally, it is agreed that we use Git and GitHub to control our version of the game, and use Google Drive to collect all the image sources to share them with all team members.

2.Concept Development

2.1 Background

At a time when scientific and technological competence is vital to the nation's future, the weak performance of students in computer science reflects the uneven quality of current science education. The research proposed games could play a significant role by addressing many goals and mechanisms for learning some knowledge about science: the motivation to learn science, conceptual understanding, science process skills, and identification with science and science learning.[1]

It is definitely that the development of the game is an evolution of an attractive and viable concept. The most important aspects of the concept are the implementation of the overall thrust, which are:

- Designing for Quiet in an age of Speed

In the era of fast entertainment, many games often use short and fast gameplay, filled with joyful background music, and frequently stimulate people's senses. These turbulent elements can make children lose their focus and become less impatient (new age screen children's syndrome). Thus, trying to design a quiet game for children to give children a better attention. It is recommended to use more natural and realistic sounds in the game to reduce unnecessary noise and avoid complicated background music.

- Reveal yourself slowly

According to the research, Negrete et al. suggests that narratives are indeed an alternative and an essential means for science communication to convey information in an accurate, attractive, imaginative, and memorable way. To present scientific information through stories, novels, comics and plays should be regarded as an important means to transmit information in the repertoire of both science teachers and science communicators. [2] Furthermore, the process of telling a story to a child should be gradual, with many details for the child to dig on, rather than to tell the child the outcome.

- Add a touch of danger

Children's favorite stories have more or less risky elements because just the right danger will make children more immersed as well as they can learn the danger of excessive protection in reality. In fact, it may not be helpful to children. Instead, children should be allowed to learn in order to grow up and face the real world.

- Leave youngsters with questions

Instead of directly presenting information to children, it is better to ask questions for children in the game and let the children explore with problems, which encourage children to explore the next step by solving the puzzle's sense of accomplishment.

- sweat the details

Images play a very important role in children's education and games. Artistic graphics are an important factor in children's enlightenment. What is more, children are more likely to be touched by the little details in the game, so when there is a chance, design some small details that will make people “Wow” in the game, and impress the children. Therefore, it is necessary for children’s game with a storyline, which could make children more interested in the knowledge spread from the game.

- Use powerful inspirations

As mentioned above, images are an important source of enlightenment for children. Visually appealing designs will greatly expand the imagination of children, and these graphics will be a source of inspiration for children in the future. For instance, one image which is not only very visually appealing and imaginative but also contains many cultural connotations and good wishes can give children deeper impression and affect their conception of the game with a lifetime.

- Easy to use

When doing game design, try to match the rules of physics, so that children can get started and play, which is more abstract. For example, games such as 3D marbles and Tetris use physics bounce and gravitation to conform to the world. Children can learn how to use them through simple interaction. In other words, usability is seen as one of the major criteria to ensure the quality of work-related products. The common attributes of usability are effectiveness, efficiency, and satisfaction.[3] Furthermore, the ease of use of a game's controls and the interface is closely related to fun ratings of that game. In addition, fun should also play a role in the design of computer games for children because enjoyment is one of the primary motivations for children to interact with technology.[4]

In summary, designers tend to become gods when making games, which is not conducive to children's play and growth. However, only when they become friends with children can they better design games or communicate with children.

2.2 Idea Selection

The development of the concept was based on the former criteria. In terms of the computer knowledge, after discussed, we are all agreed on the basic computer knowledge including the type of the variable, the basic computer architecture, the Hexadecimal conversion, etc as the entrance of grasping the computer science. The difficulty is how to combine with the attractive game. We initial have three ideas all around the above knowledge in order to implement the game.

2.2.1 Idea 1

The first idea is referring to the space game.

Figure 2 space game

It means that there are various target objects that are going to destroy your plane. The target object is including various size, and when the objects are destroyed the player would get a different score. To be specific, when the ‘int’ object is destroyed, the player would get the two marks. The “int” would have a smaller size and higher speed than “char” object. Thus, the “int” object would have more score than “char”. Also, the plane the player used would be changed by using the coins in the store. The method of getting coins is answering the question about computer technologies. The name of the plane also named by C, Java, etc. And the level is named by Program oriented, Object-oriented etc. It is our first idea which is not attractive enough. However, it is saved as our alternative theme.

2.2.2 Idea 2

The second idea is referring to Computer Castle.

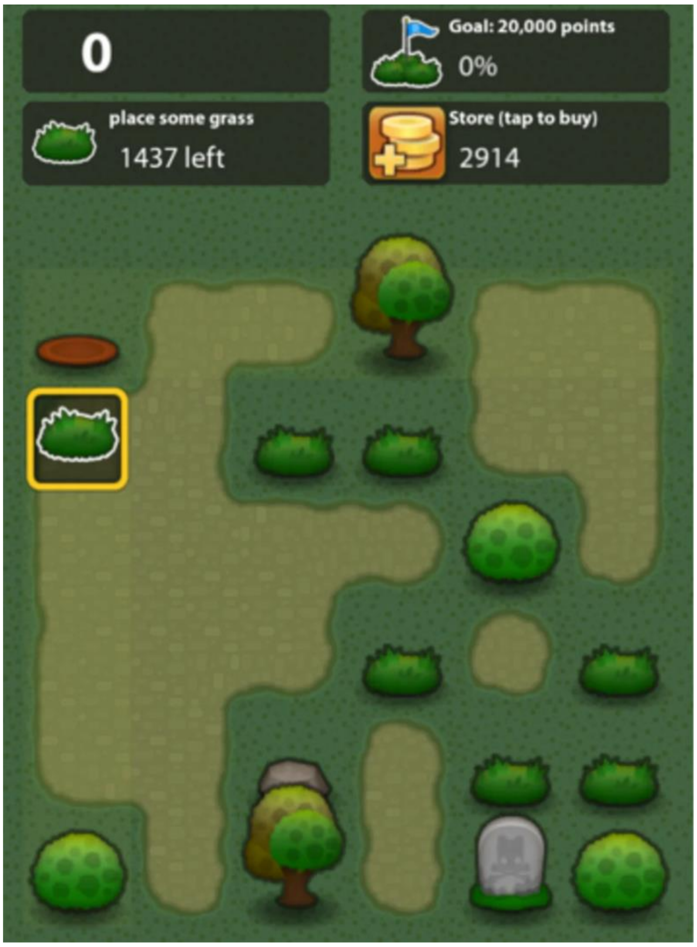


Figure 3 Computer Castle

The main rule is placing the computer components to form a whole computer; avoid the bears disturb! Specifically, the player clicks on an open square to place the objects and three same objects will make a match and turn into a new valuable object. What is more, four or more objects will turn into a super object

Furthermore, bears will appear and move around to disturb you build, and the play can trap the bear to a stone or combine three bears to a treasure. It is a game which has more focus on the experience of game rather than responding the computer knowledge.

2.2.3 Idea 3

The third concept is Escape Computer, which is our final choice.

Figure 4 Escape Room

The background is a junior school student wake up and find yourself in a black box. He tries his best to escape from the computer, but when the student succeeds, the world would become the wasteland full of rubbish. The game would have four rooms: an Input device, Control unit, Arithmetic Unit, Memory. The output device would be used in the final room to escape from the computer. Each room has different methods to escape, which are related to the escape game and computer technology.

2.3 Final Concept

The development of these ideas coincided with the team's research, and finally, the concepts were voted upon by the class group. The selection presented to the group is the last one. We all think it is a relatively attractive game than the left as well as it is well combining with computer knowledge. Thus, the third is our final selection because escape rooms have grown in popularity in the past few years.

What is more, after we are doing some research, we suggest escape rooms are games in which players need to complete a series of challenges to win. According to the research, where the first generation of escape rooms focused on difficult logic puzzles, escape rooms today have now evolved into fully immersive environments with high quality props and effects. [5] Additionally, the theme of escape rooms can make a difference to the play experience.

2.4 Programming Style

As we all know, a standard programming style plays a significant role in the program implementation in the group. There are a lot of initial codes from the website, tutorial, and so on which meant that is was not inconsistent. However, when the group works, it is impossible that the whole group code has a different programming style. This is because it not only reduces the readability of the code, but also reduces the effectiveness of the group work. Thus, it is evident that we need a home style so that all code conformed to the same standards. Also, it makes the programmer pay more focus on the code rather than the appearance.

3 Work Plan

There are five main stages including the made theme, previous study, software implement, evaluation, and wring a report in our group, and we make a work plan to achieve the final goals more effectively. The details of the work plan are shown in the following figure. In particular, the report would be completed by the 20/5/19, but we prepared the resource we need with our development of the software.

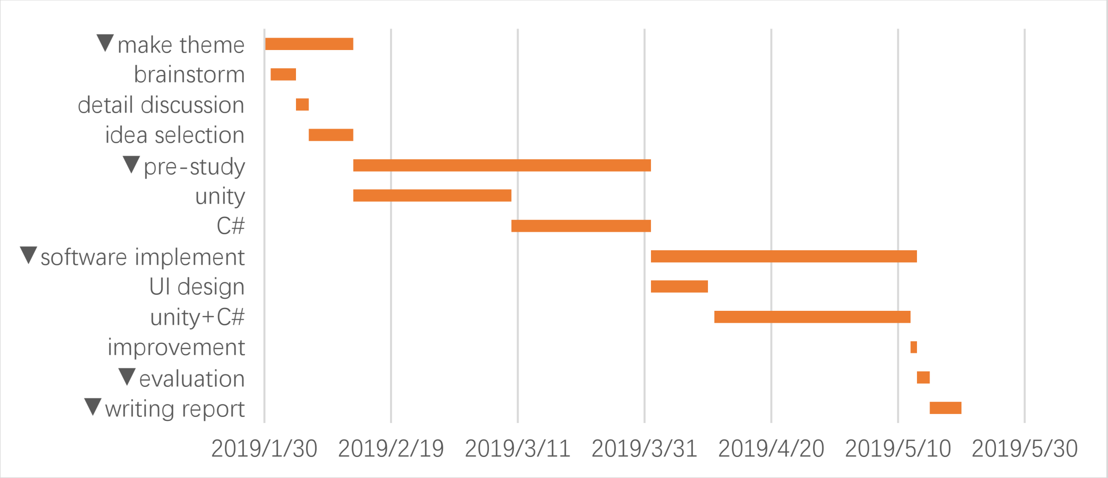


Figure 5 workplan

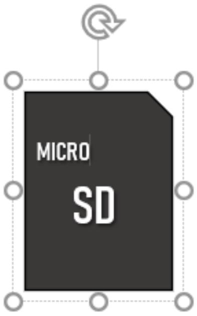
4 Game Implementation

4.1 Component Design

There are many components in this game, which we decided to design them by ourselves, such as a computer, CD, lock, notebook, USB disk and so on. These components are placed in each room, so the style and the color of these components must correspond to the background image, which is our design concept. After discussing, we decided to use PowerPoint and Photoshop these two tools to design all the components.

4.1.1 PowerPoint Design

The implementation process with PowerPoint consists of main two steps. The first one is to draw the picture of SD card by simple line and add some letters on it by using text that is shown in Figure 6. Then, as Figure 7 indicates, we need to implement the 3-D rotation of the before picture to make it look more vivid.



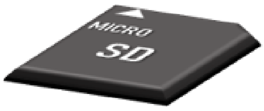


Figure 6 line drawing of SD card **Figure 7 SD card**

4.1.2 Photoshop Design

A key point of design with Photoshop is that grasp the use of layer design. For example, the implementation of the lock that is descript in Figure 8 includes three layers. The first layer is the grey bottom panel, then we can add the second layer that contains the black screen and white button, finally, we add the third layer, numbers on the second one. The finished product is shown in Figure 9.

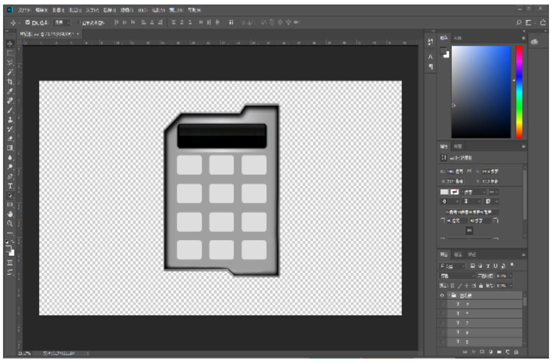


Figure 8 the implementation in Photoshop

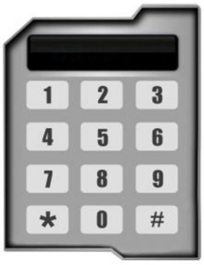
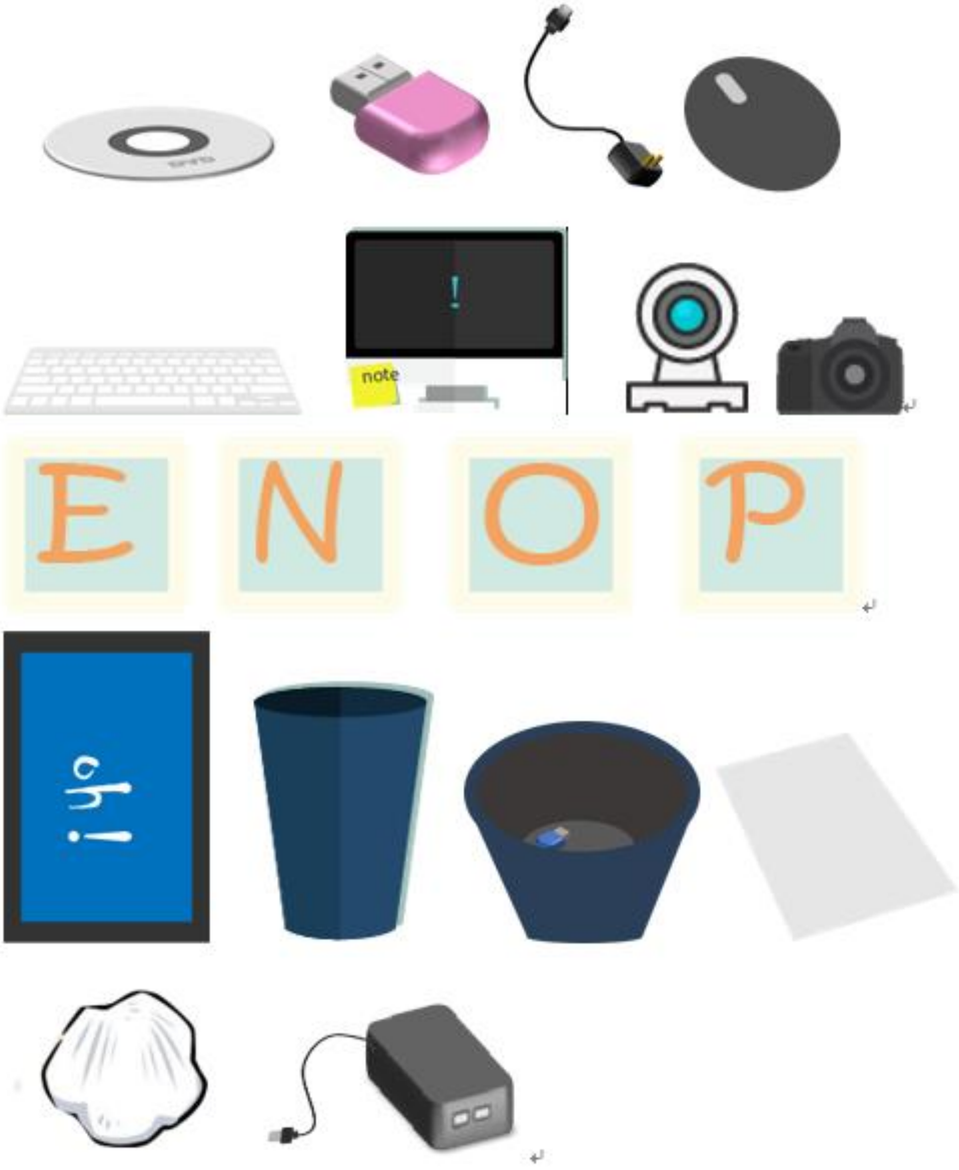


Figure 9 lock

The whole components we designed can be seen in Figure 10.



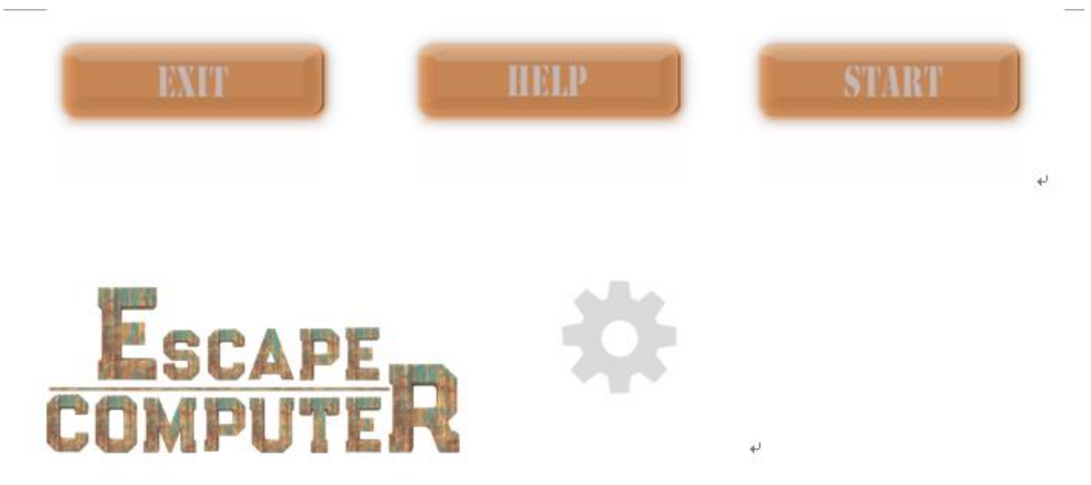


Figure 10 components

4.2 Drag-and-Drop Design

4.2.1 Description

while a player is playing this game, drag and drop operations would be used frequently. Specifically, to escape from room 1, the player needs to find all the input devices and drag them to the right item box. Only when all the correct devices are found, the player can succeed to enter the next room. Also, when a device is dropped in the right item box, it will become bigger than it is before, which make the it look more clear. This visual effect is shown in Figure 11. Moreover, drag-and-drop operation is also used in room 4 that the player has to drag a proper device to the computer.

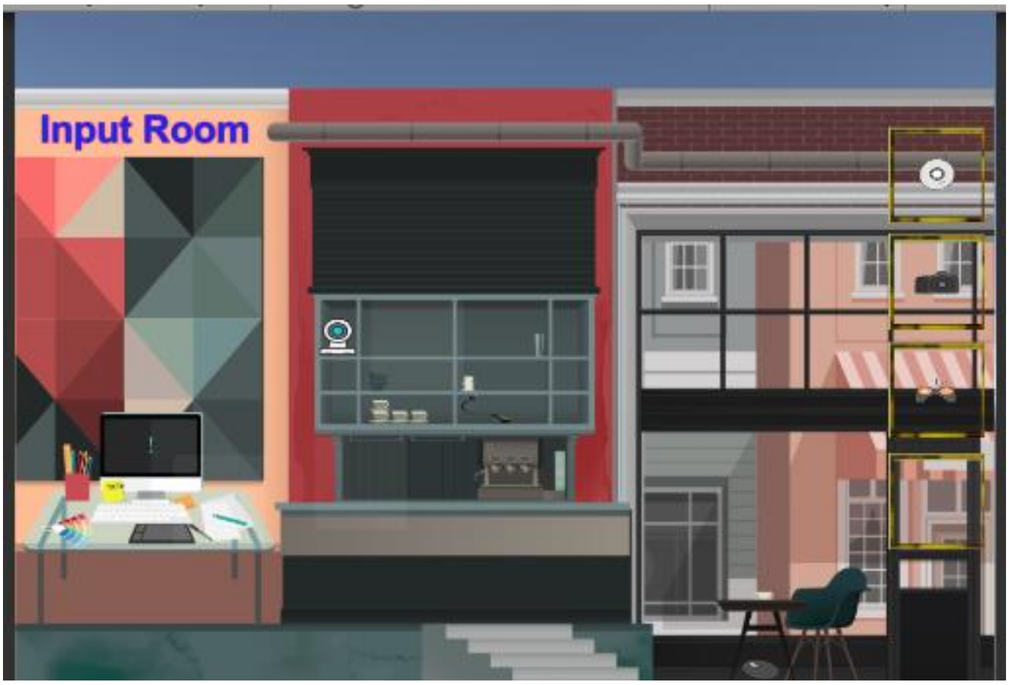
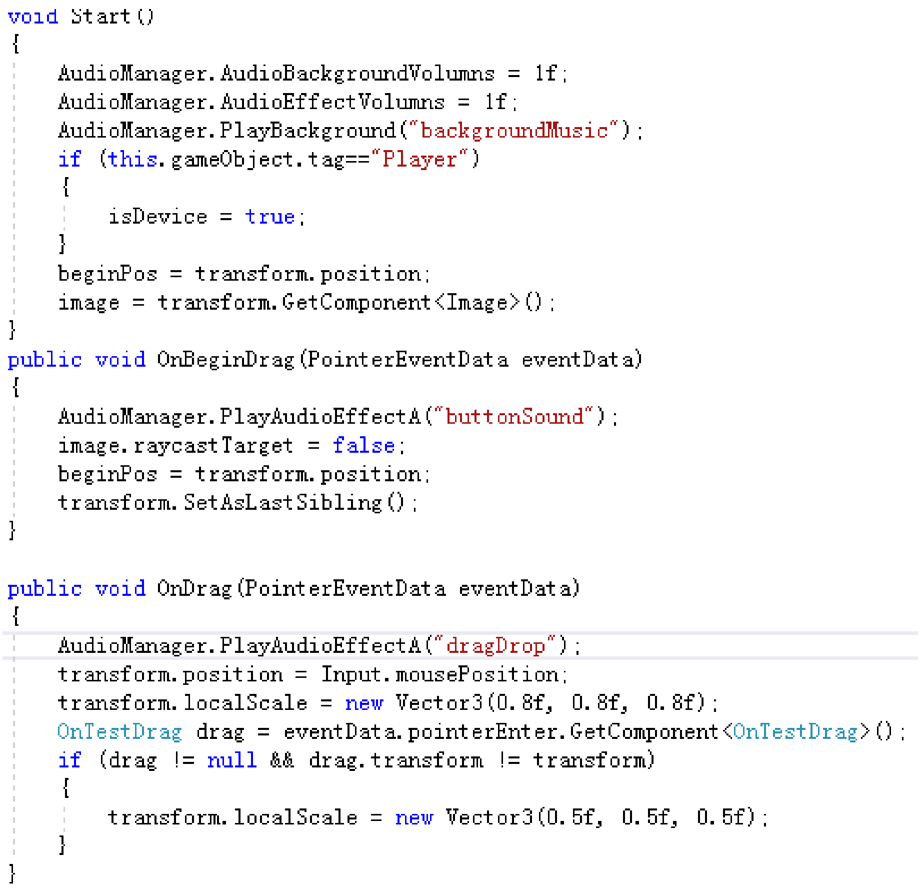


Figure 11 Room 1

4.2.2 Program Implementation

The Drag-and-Drop function is implemented by importing EventSystem class of unity. In detail, this class includes Handles input, raycasting, and sending events. And it is responsible for processing and handling events in a Unity Scene. A Scene should only contain one EventSystem. The EventSystem works in conjunction with a number of modules and mostly just holds state and delegates functionality to specific, overrideable components. When the EventSystem is started it searches for any BaseInputModules attached to the same GameObject and adds them to an internal list. On update each attached module receives an UpdateModules call, where the module can modify internal state. After each module has been Updated the active module has the Process call executed. This is where custom module processing can take place.

The following code is the main part to realize this function.



```
void Start()
{
    AudioManager.AudioBackgroundVolumns = 1f;
    AudioManager.AudioEffectVolumns = 1f;
    AudioManager.PlayBackground("backgroundMusic");
    if (this.gameObject.tag=="Player")
    {
        isDevice = true;
    }
    beginPos = transform.position;
    image = transform.GetComponent<Image>();
}
public void OnBeginDrag(PointerEventData eventData)
{
    AudioManager.PlayAudioEffectA("buttonSound");
    image.raycastTarget = false;
    beginPos = transform.position;
    transform.SetAsLastSibling();
}

public void OnDrag(PointerEventData eventData)
{
    AudioManager.PlayAudioEffectA("dragDrop");
    transform.position = Input.mousePosition;
    transform.localScale = new Vector3(0.8f, 0.8f, 0.8f);
    OnTestDrag drag = eventData.pointerEnter.GetComponent<OnTestDrag>();
    if (drag != null && drag.transform != transform)
    {
        transform.localScale = new Vector3(0.5f, 0.5f, 0.5f);
    }
}
```

Figure 12 code implementation of Drag-and-Drop

4.3 Environment Design

The game is set in the background that a child was trapped in the computer and need to go through the computer component room and find the way out. So, the environment of this game should show the computer component feature. Furthermore, as the game is design for teenagers to educate them to learn knowledge about computer, so the background should be appealing enough for teenagers. Finally, we decided to adopt flat design which allows interface designs to be more streamlined and efficient. Moreover, it is easier to quickly convey information while still looking visually appealing and approachable. Additionally, it makes it easier to design an interface that is responsive to changes in browser size across different devices. With minimal design elements, websites are able to load faster and resize easily, and still look sharp on high-definition screens. As for our game’s background, the rooms, we used warm color design style and some ordinary furniture, such as sofa, desk, chair, and so on, to make the game scene be more attractive.

4.4 Scene Design

There are four theme rooms, which means four levels in our game. The player needs to finish different kinds of duties in each room whose difficulty is gradually increasing that can help teenagers get knowledge related to computer science.

4.4.1 Input Room

The first theme room in the game is named Input Room, which is shown in Figure 8. The duty in this room is to collect all the input devices, including video camera, keyboard, mouse, and gamepad. We also put some unrelated devices, such as CD, the data line, and so on to improve the difficulty. By finishing this duty, children can distinguish input devices with others.

We used Adobe Illustrator, which is a vector graphics editor, to add the corresponding components to the background scene. In terms of program implementation, this room is the most difficult one to complete among these four rooms, which is mainly reflected in the use of Rigidbody and Collider concepts of unity.

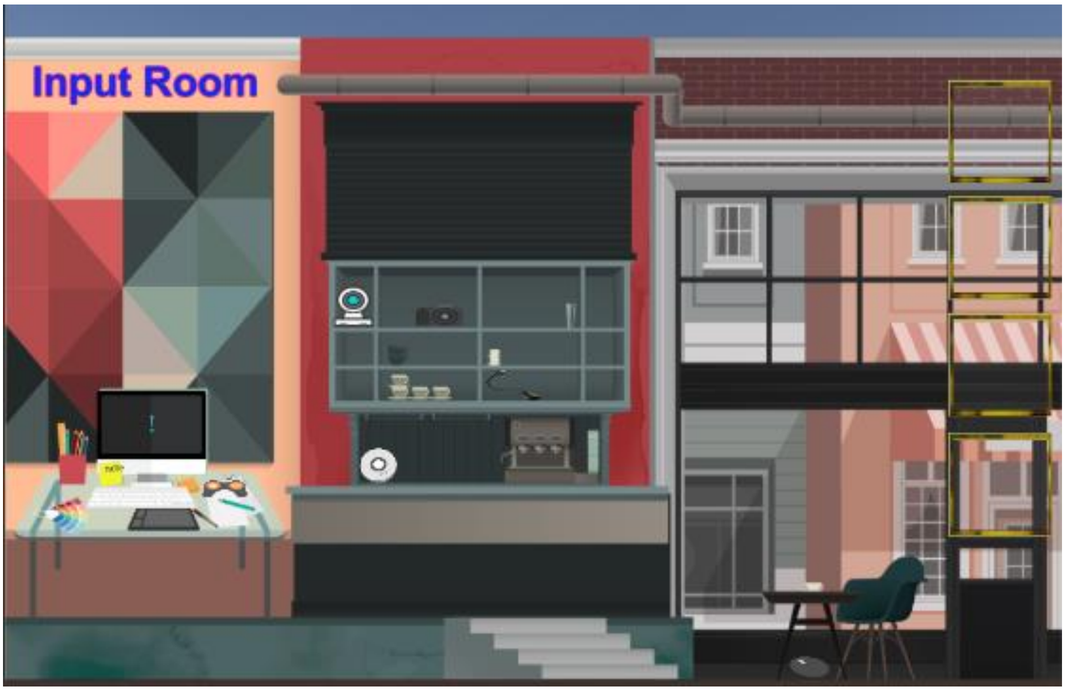


Figure 13 Input Room

4.4.2 Control Room

The second theme room is called the Control Room that is presented in Figure 14. To escape from this room, the player must answer serval questions correctly in terms of the clue. In detail, there are four pictures on the wall which have different letters on them. The questions will only appear if you click on these images so that will form a word ‘open’. These questions consist of binary, data type, loop statement, and rounding operation. It is definitely that the player will acquire these related knowledge after finishing the duty in this room.

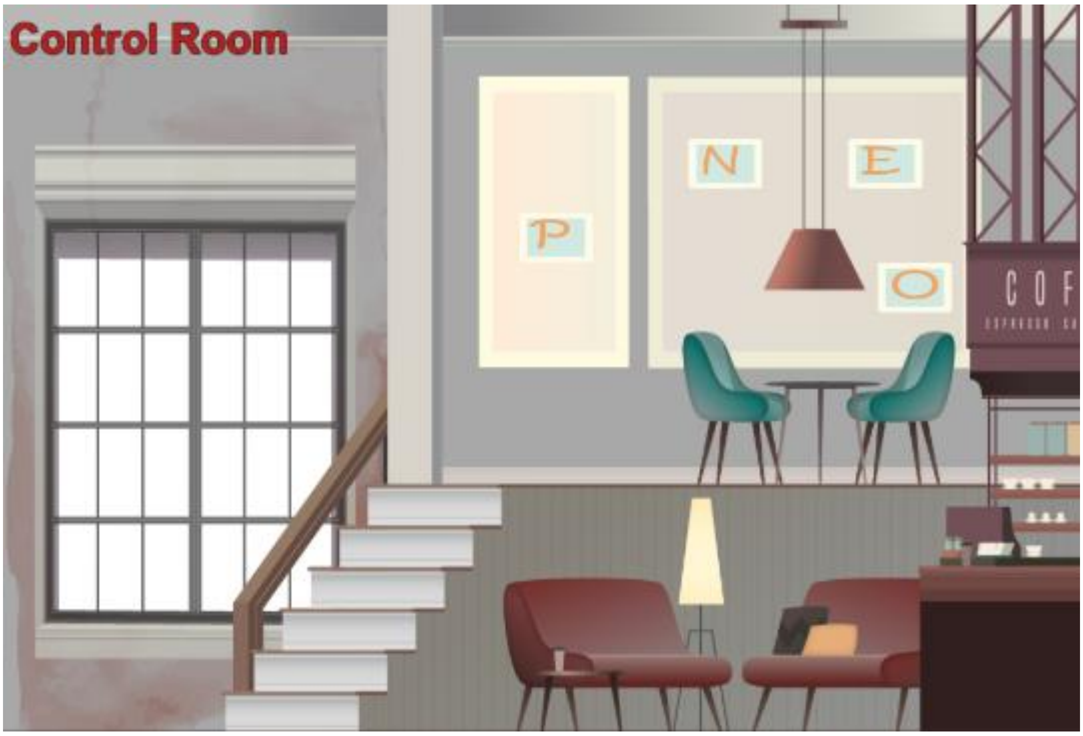


Figure 14 Control Room

4.4.3 Arithmetic-unit Room

The third theme room is Arithmetic-unit Room. In this room, the player needs to find several arithmetic questions by observing clues in the room. In other words, the player must click some certain items to find the questions. The first question about binary conversion exists in the blackboard on the wall, whose answer is 21. When the player clicks the paper lay on the floor, he or she will find the second question regarding the logic operation, whose answer is 0. The last question is hidden in the spitball on the ground related to memory calculation, and its answer is 519. Finally, the player should click the lock on the door and input the password, which is a combination of before three answers. In fact, the password is 210519, that means our deadline for Software Engineering and Group Project. The whole duty here could inspire teenagers' abilities of Observation, calculation, and creation.

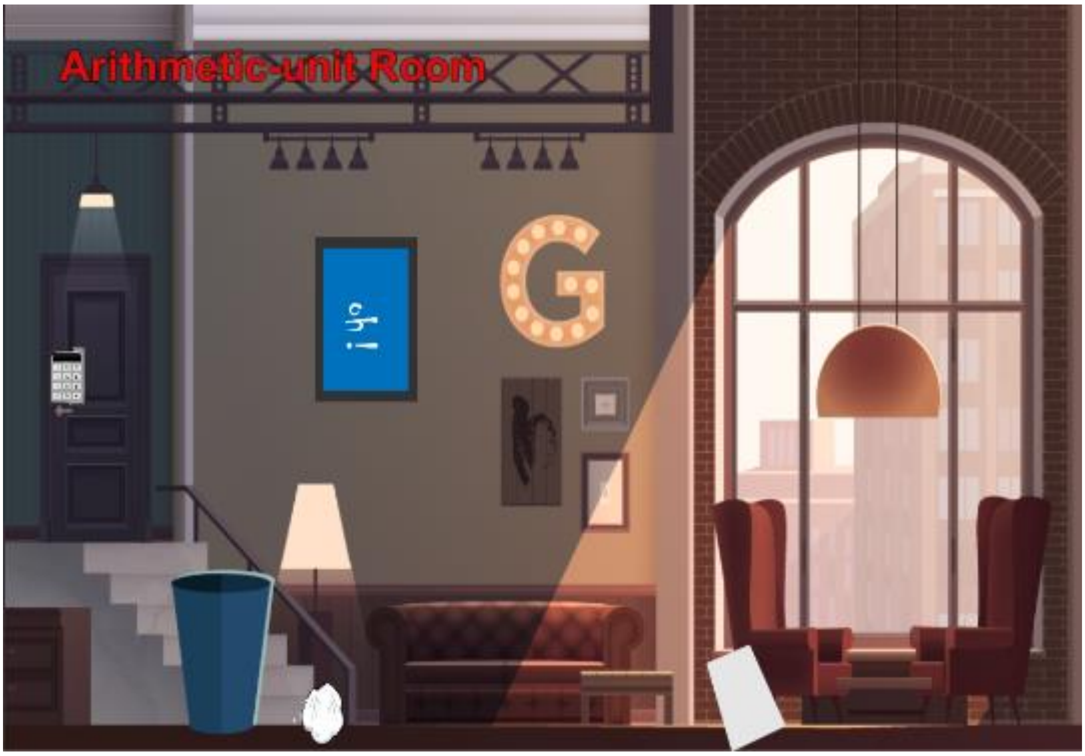


Figure 15 Arithmetic-unit Room

4.4.4 Memory Room

The last theme room is Memory Room that is shown in the following Figure. In this room, the player needs to click the camera on the desk to find the clue that prompts the player to find a proper device to escape from this room. Same with the second room, there are some deceptive items here. The correct operation is to click the blue trash can, and a USB disk can be found in it. After dragging this USB disk to the computer, a window for entering the password, which the player should observe this room to find it, will pop up. The right password is distributed in the wall that is 2645. If the correct password is entered, the player will finish the whole game and succeed to escape from the computer.

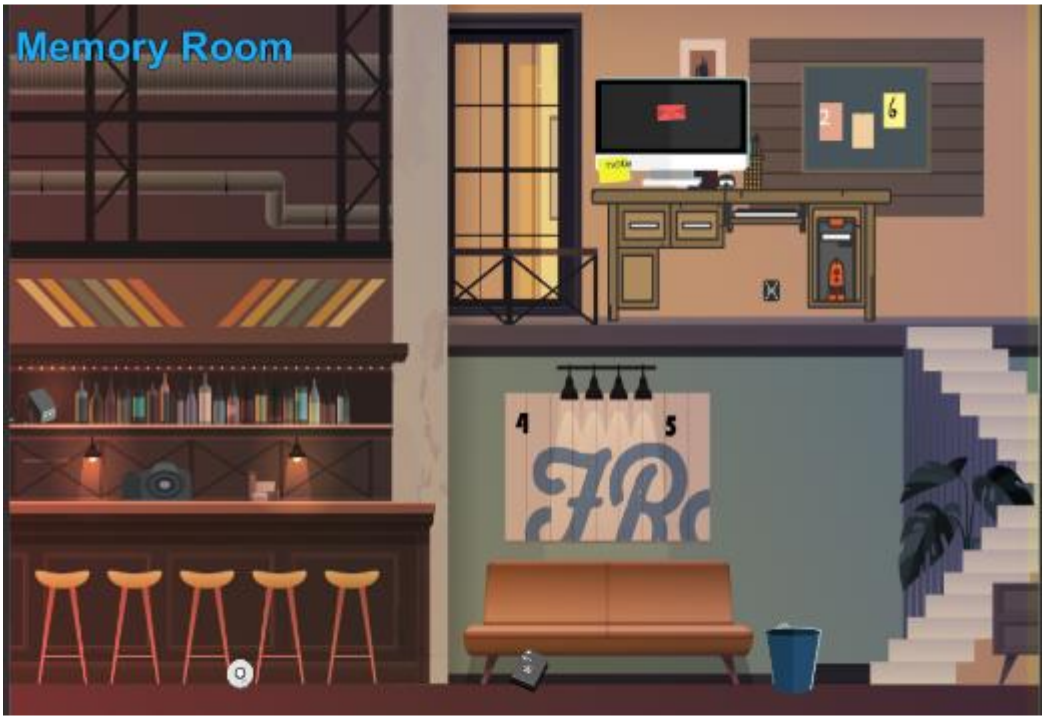


Figure 16 Memory Room

4.5 Menu Design

The game starts with the menu scene, in case the title of the game and basic buttons could be directly shown at the beginning of the game. The menu scene includes the main scene of the game and buttons for start and exit. There are also buttons for help and setting. When the user finishes the game successful, the game will also end with an end menu scene which has two buttons for replay and exit.

4.5.1 Description

The start-up interface of the game is designed to show the title of the game and main menu of the game. There are both control buttons and setting button in the start menu.

The control buttons include the start button, help button, and exit button. When clicking the start button, the game start and the player will enter the first room, input room. When clicking the exit button, the game window will be closed. When clicking the help button, the instructions of playing the game will be showed on the screen. In the beginning, we designed to show a separate help interface when the help button is clicked. However, this tuned out to be an ineffective design because the instruction is too short while the code quantity will be increased greatly.

The setting button could adjust the volume of ambient and clicking sound. Both the ambient sound and clicking sound are default as the maximum and could be adjusted by dragging the scroll bar.

The end menu of the game contains two buttons, replay, and exit. When clicking the replay button, the game will be restarted as a new game. When clicking the exit button, the game window will be closed.

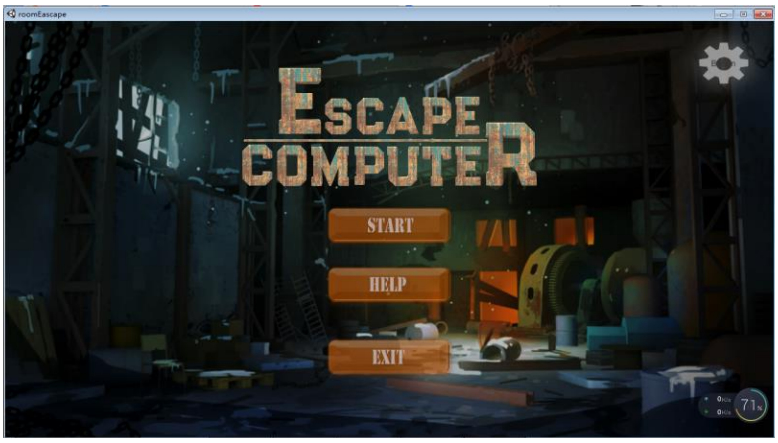


Figure 17 The start-up menu

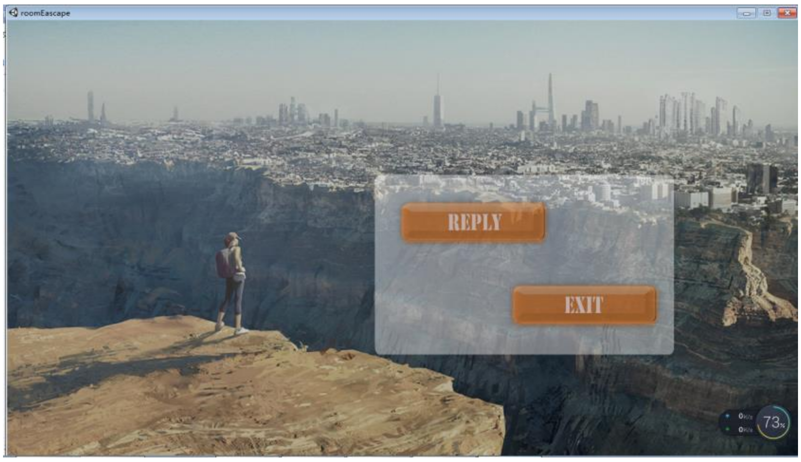


Figure 18 The end menu

4.5.2 Implement

Menus are executed in two stages. The first stage is to create basic menus, and the second stage is to create more elaborate menus.

In the first stage of the design and implement a menu, our team first design the button style and font. The menu used in the game is designed by PowerPoint.

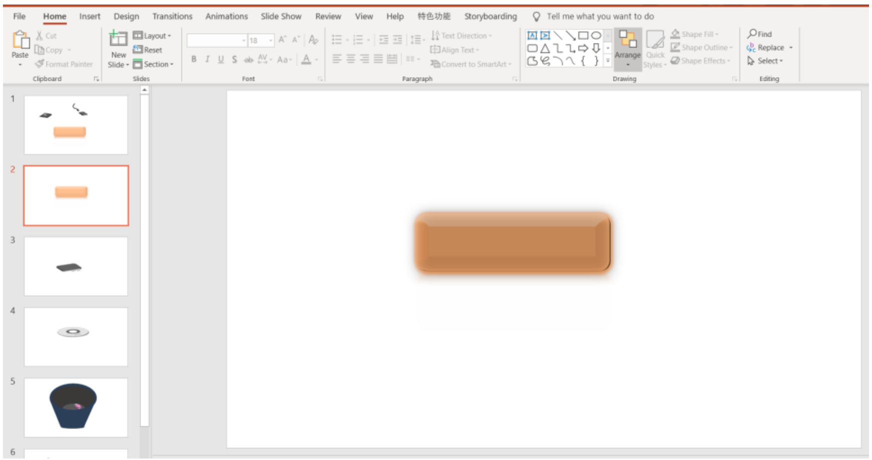
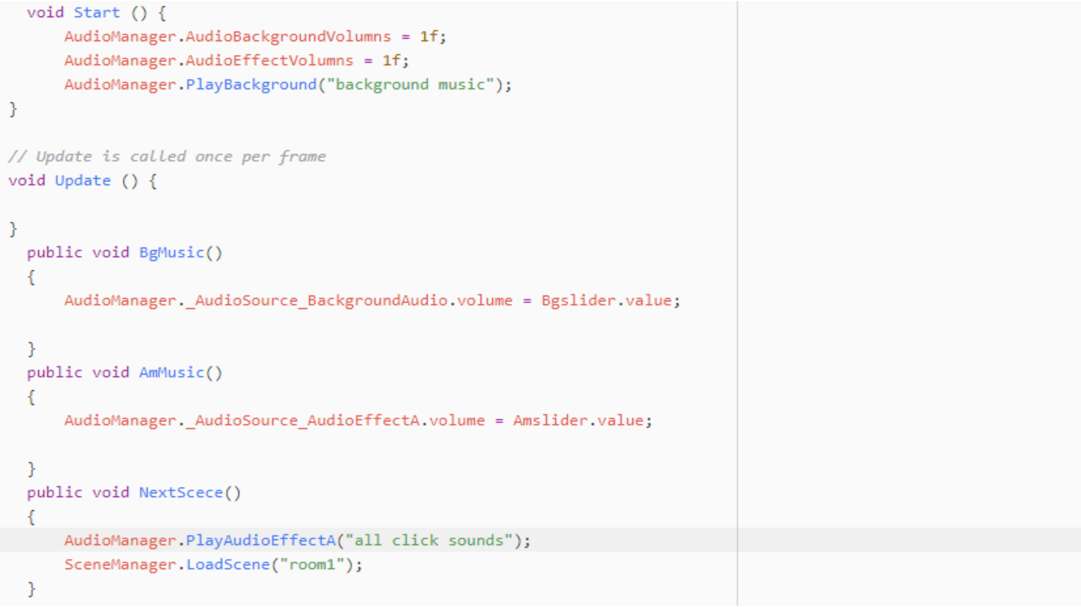


Figure 19 the design of the button

Then our team learned the basic operations of Unity and created a course menu with the technology of creating canvas and buttons. Our team members showed their creativity in this stage. Then we work on the c# code to link the control buttons with other scenario and make the setting button implement the actual function. After testing the menu could work well, we have made some modification in the style of the button and make it more suitable for people's aesthetic need.



```
  void Start () {
      AudioManager.AudioBackgroundVolumns = 1f;
      AudioManager.AudioEffectVolumns = 1f;
      AudioManager.PlayBackground("background music");
}

// Update is called once per frame
void Update () {

}
  public void BgMusic()
  {
      AudioManager._AudioSource_BackgroundAudio.volume = Bgslider.value;

  }
  public void AmMusic()
  {
      AudioManager._AudioSource_AudioEffectA.volume = Amslider.value;

  }
  public void NextScece()
  {
      AudioManager.PlayAudioEffectA("all click sounds");
      SceneManager.LoadScene("room1");
  }
```

Figure 20 implement the function of buttons

4.6 Sound and Video

The sound used in the game including a piece of background music, a piece of finishing background music and several sound effects, a video is also added in the game which is served as the game ends.

The background music is a crucial part that will influence the game effect. The background sound used in this game should not only be suitable for the game but also heighten the atmosphere. What’s more, as the game is designed for teenagers aged between 7-15, the background sound should be attractive and interesting to the target audience.

The sound effect means the sound produced when the user operates. The clicking sounds are a necessary supplement which could give positive feedback for the user when they do the valid action. The sound effect including the click, drag, drop, pick up items and so on.

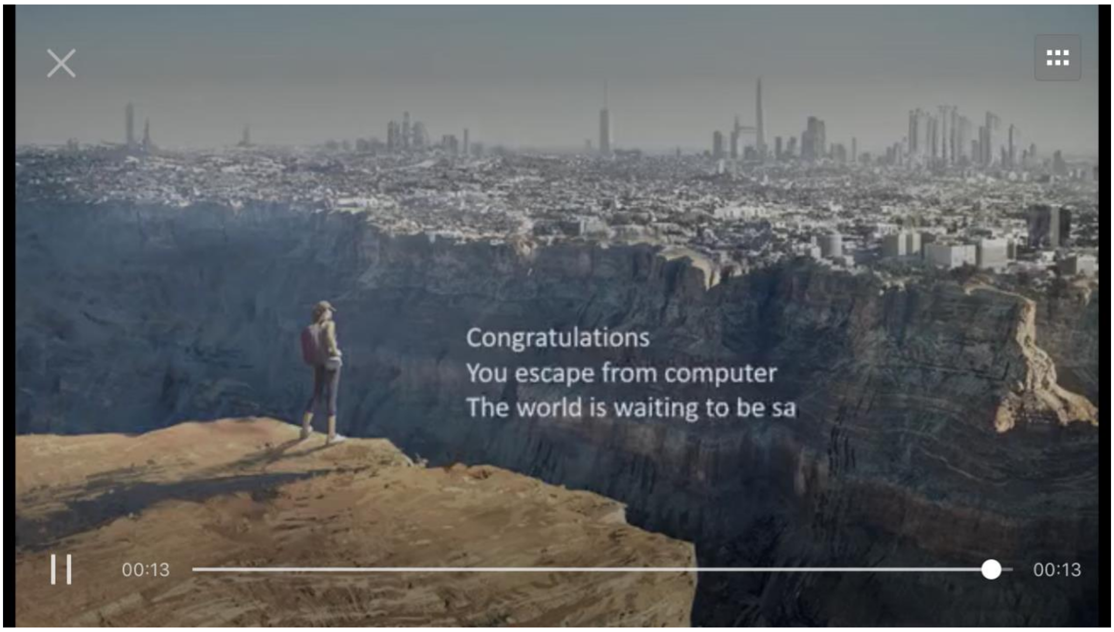
A video is added at the end of the game. When the player has finished the game successfully, the video will be played to show a scene of time tunnel which indicated the player has escaped the computer and came to the open field. The video will also give some hint to show the plot of the story.

4.6.1 Description

The background sound(シェフ from Hisaishi Jō)is royalty-free which is download from music software QQ music. This song is brisk and houmous with is both suitable for the game scene and attractive to young people. This is corespondent with our design concept of ‘Designing for Quiet in an age of Speed’ and ‘Reveal yourself slowly’. The ending background sound is also a piece of plain music (Horizon) downloaded from QQ music. This music is grand and desolate, which accords with the ending scene.

The sound effects come from an online source. Thousands of sounds are selected and we finally choose the sounds that could both show the action features and are harmonious with the background sound.

The ending video is made by ourselves by PowerPoint. The video shows a scene that the player has successfully escaped from the computer and is passing through a tunnel. Finally, a wild field is showed to the player. This indicated that the word has trapped with electric components and if human fail to deal with the relationship with computer appropriately, people will be replaced by the machines they created and the world will be a mass. The players are encouraged to save the world with their own effort.

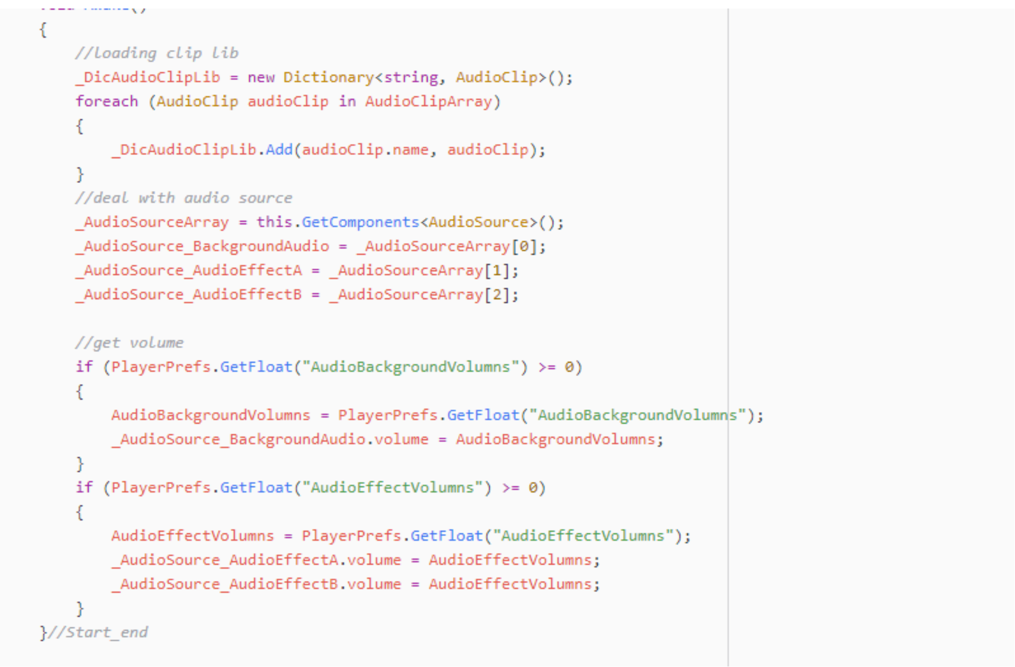


Congratulations
You escape from computer
The world is waiting to be sa

Figure 21 the ending scene

4.6.2 Implement

The code of playing ambient music and sound effects is in the AudioManger class. The code implements a part of background sound and ending sound is divided into four parts: loading audio clip, play ambient music, play a sound effect, change ambient music volume, and change sound effect volume.

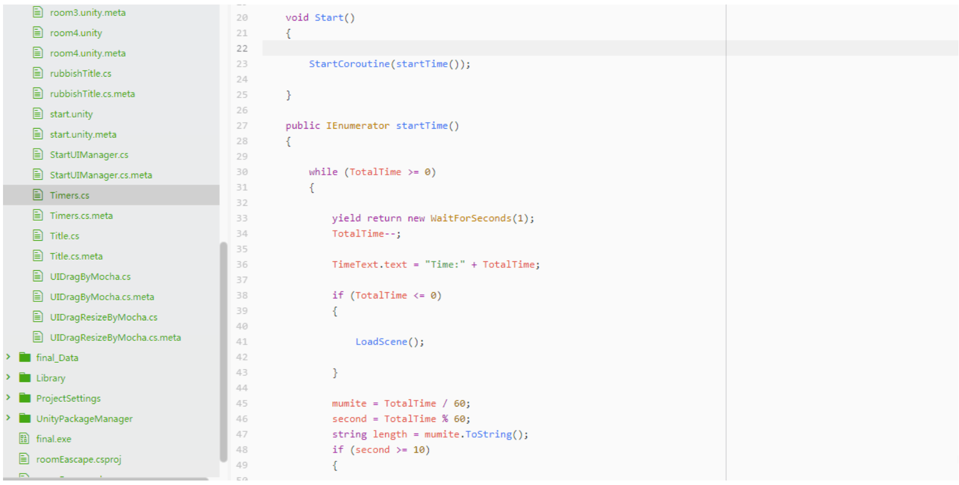


```
{
    //loading clip lib
    _DicAudioClipLib = new Dictionary<string, AudioClip>();
    foreach (AudioClip audioClip in AudioClipArray)
    {
        _DicAudioClipLib.Add(audioClip.name, audioClip);
    }
    //deal with audio source
    _AudioSourceArray = this.GetComponents<AudioSource>();
    _AudioSource_BackgroundAudio = _AudioSourceArray[0];
    _AudioSource_AudioEffectA = _AudioSourceArray[1];
    _AudioSource_AudioEffectB = _AudioSourceArray[2];

    //get volume
    if (PlayerPrefs.GetFloat("AudioBackgroundVolumns") >= 0)
    {
        AudioBackgroundVolumns = PlayerPrefs.GetFloat("AudioBackgroundVolumns");
        _AudioSource_BackgroundAudio.volume = AudioBackgroundVolumns;
    }
    if (PlayerPrefs.GetFloat("AudioEffectVolumns") >= 0)
    {
        AudioEffectVolumns = PlayerPrefs.GetFloat("AudioEffectVolumns");
        _AudioSource_AudioEffectA.volume = AudioEffectVolumns;
        _AudioSource_AudioEffectB.volume = AudioEffectVolumns;
    }
}//Start_end
```

Figure 22 implement of loading audio clips

When playing the video, a timer is used. At the end of the countdown, the game will jump to other scenes.



```
void Start()
{

    StartCoroutine(startTime());

}

public IEnumerator startTime()
{

    while (TotalTime >= 0)
    {

        yield return new WaitForSeconds(1);
        TotalTime--;

        TimeText.text = "Time:" + TotalTime;

        if (TotalTime <= 0)
        {

            LoadScene();

        }

        mumite = TotalTime / 60;
        second = TotalTime % 60;
        string length = mumite.ToString();
        if (second >= 10)
        {
```

Figure 23 The implement of timer

4.7 Interactive Mode

The interactive method in this game includes drag and drop, click and enter. The drag and drop action is detailed described in section 1.2. the click action is used when getting hint information and answer single choice questions. The enter action is introduced when entering the password.

4.7.1 Description

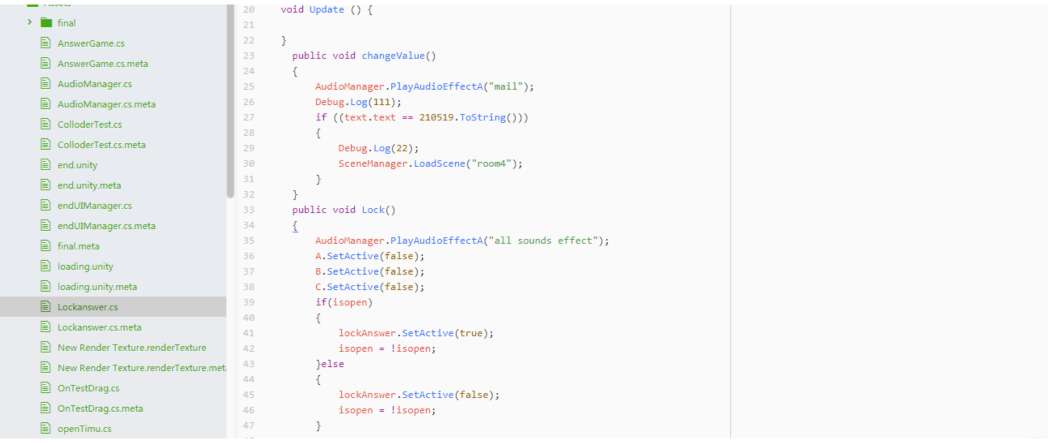
The click action is widely used throughout the game. In each room, there are some hints hiding in the furnishing. Some devices or decorate in the rooms has very obvious suggestibility and the player could click them to get the initial clues to continue to solve the problems. For example, in the first room, Input room, there is an exclamation mark on the computer screen which can easily be found in the center of the room. When the user clicking the exclamation mark, it will show hints about how to find the clues to escape from this room. There were some arguments about implement the method of the first hint. Some members argued that the simple and basic hint shows directly to the player as the game is designed for teenagers and should not be very difficult. However, we achieved the agreement that the hint should be got by finding and clicking. Because this will add the play-ability and interestingness of the game.

Furthermore, when the players are answer single choice questions in the next two rooms, the player needs to click the answer he chose. The game will give different reactions when the player clicks different choices.

The entry box is used twice in the game. The first enter box is showed as the input field of the coded lock. When the player enters the right password which is combined with the question answers into the text field, the game will change to the next scene. Another enter box is also a text field which could enter the password get from the clues in the room.

4.7.2 Implement

The implemented code of the lock answer is shown as follows. The number entered is converted into a string and compared with the correct answer. If the number entered correctly, the scene will be changed.



```
void Update () {

}
  public void changeValue()
  {
      AudioManager.PlayAudioEffectA("mail");
      Debug.Log(111);
      if ((text.text == 210519.ToString()))
      {
          Debug.Log(22);
          SceneManager.LoadScene("room4");
      }
  }
  public void Lock()
  {
      AudioManager.PlayAudioEffectA("all sounds effect");
      A.SetActive(false);
      B.SetActive(false);
      C.SetActive(false);
      if(isopen)
      {
          lockAnswer.SetActive(true);
          isopen = !isopen;
      }else
      {
          lockAnswer.SetActive(false);
          isopen = !isopen;
      }
```

Figure 24 Code implement of lock answer

4.8 Conclusion

The whole design and implement process is divided into several small tasks and realized step by step. Every small task is responsible for at least two group member. Every member has participated in the coding process and was in charge of a particular part. The code is subject to the supervision of all group member and gathers the wisdom crystallization of all members. Sub-tasks are useful in the overall strategy of the projects. In addition, each sub-task has some challenges for the group members and lead us to think deeper and learn more.

5 Future Expansions and Improvements

According to the bugs which still exist in the program, there are plenty of aspects on which the project could improve. According to what we gained from the experience of this project, here are some possible improvements which are not beyond our ability, to make this project able to attract more teenagers so that the educational purpose is better achieved.

5.1 More Rooms

Currently, there are only four rooms; a player may feel unsatisfied after fleeing all of them and still yearn for more. More rooms with more kinds of educational functions are able to better embody the theme of this project. Meanwhile, as for the names of the new rooms, there are plenty of computer components which could correspond with the rooms, e.g., CPU, CD-ROM, and printer.

The current rooms could also be categorized into more specific ones. For instance, the input room could be replaced by many rooms, including the keyboard, mouse, camera, etc.

5.2 More Puzzles

Currently, the puzzles include to collect stuff and to answer the questions. There could be more sorts of puzzles e.g. a parkour game where the player should discern different computer components and get the right ones for higher points, a plane battle game where the player's plane fights with computer viruses, a Tetris-featured game where their same computer components in a line will disappear, etc.

For puzzles of stuff collecting, there could be a global backpack for stuffs in different room, which constitute a key. Different stuffs may form into a new one. For instance, when a graphics card and a screen are collected, the player can choose to assemble the two kinds of stuff into a monitor.

5.3 Better sound Effects

Currently, all sounds are in correspondence with different interfaces, including the start interface, the room interface, and the ending interface, and different events are e.g. picking up stuff. The background music could be linked with the current stage in the game line. For instance, when the player has just entered the room, the background music is grim. As the player solve more puzzles, the background music becomes bright.

5.4 Multiple Story Lines

Currently, the storyline is single threaded which means that to get into room C from room A; the player has to pass room B. With multiple storylines, different room form into a net instead of a single line, which means that the player may have various choices to get the destination.

In this case, there should also be a reminding function to tell the player whether all the rooms have been reached, or which puzzle is remained to be solved, in case the player cannot enjoy all the contents of the game.

5.5 Multiple Views in One Room

Currently, there only one view for each room, in which all stuff in the room is visible. With multiple views, there could be hidden stuff in the room which needs to be discovered in other views, e.g., in the drawer, behind the closet, under the desk, etc.

What's more, the room could be constructed with 3D models so that the player is able to choose any view. 3D modeled rooms will also be able to make it harder to find the stuff as it can be hidden in secret, rarely found place.

6. User Manual

6.1 Game Background

You are an ordinary middle school student. One morning, when you open your eyes, you find that everything around you becomes very strange. As you explore your surroundings, you gradually find yourself falling into a computer. You start to feel scared and want to escape. But you find that this is not a very easy thing to do, because everything here seems to have been carefully designed, and the only way to get out of here is to solve a lot of puzzles until you find an outlet to escape this computer.

6.2 How to Play

Our game has four main rooms and an escape tunnel

- Input room
- Control room
- Arithmetic Unit
- Memory room
- Output tunnel

Each room has different ways of escaping. Players need to follow the prompts in each room to find how to open each room and eventually escape from the computer. For example, some rooms randomly place objects, such as boxes, little ball of paper, in which little ball of paper will contain some computer-related problems (binary, data type, etc.), players need to choose the correct answer and combine these clues to escape from each room.

6.3 Detailed Instruction

6.3.1 Start Interface

This page is the main interface of the game, which is the beginning interface. This interface includes four buttons, including Start, Help, Exit, and the Settings button in the upper-right corner of the screen. The Settings button adjusts the volume of the background and click. The operation in this interface is quite easy, just to click with the mouse. Then it comes to game interfaces.

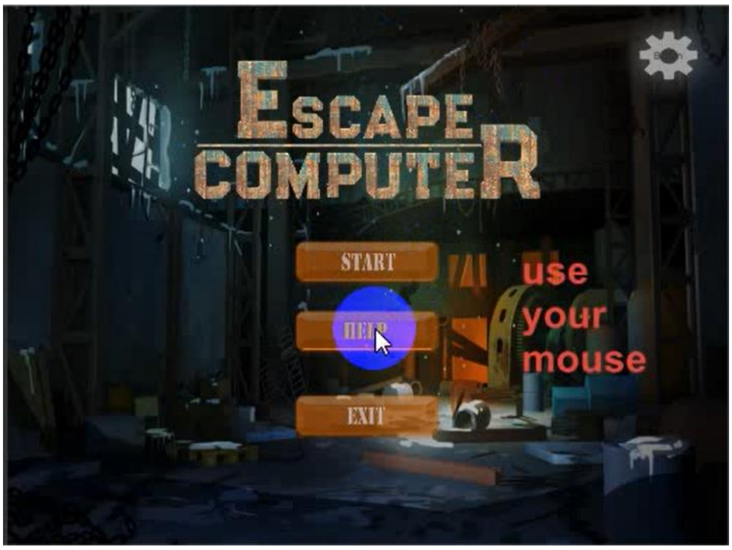


Figure 25 Start interface

6.3.2 Input Room

The first room is called the input room. In this room, there is a computer on the desk. Click the screen, and there appears a board saying that you have to find the right input devices in the room. The room is a mess, and there are some misleading ones. The right input devices are the keyboard, the mouse, the camera, and the gamepad. The wrong ones are the CD, the SD card, hard disk, and the USB disk. The right ones can be dragged into the backpack on the right. Once all four of them are collected, the next room appears.

6.3.3 Control Room

The second room is called the control room, where there are four posters on the wall with letters “O”, “P”, “E” and “N” respectively. Click the posters with the right sequence, and each time a question pops out. To solve these questions, the player has to get the advanced knowledge of computer coding. For examples, what is the type of 3.14? In JavaScript, 5+8+”1” equals to? For this loop, the final value of I is 5. 5/2.0 is? After answering all four questions, the next room appears.

6.3.4 Arithmetic Room

The third room is called the arithmetic room. In this room, the player also needs to answer some questions. But unlike the control room, the problems in this room are all about the computer's arithmetic unit. Player need to look for different components to find the questions to answer. For example, if you click the notebook, then the question pops out: what is the binary number 10101 as a decimal number. The answer is 21. The blackboard is with the question of what is the result of 1 && 0 and the answer is 0. On the ground, there is a spitball with the question to convert 5314.56 KB into MB, and the answer is 519. Combining all three answers you will get 210519, which is the deadline of this project.

6.3.5 Memory Room

The last room is the memory room where there are some memory devices on the ground, including a USB disk, an SD card, a hard drive, and a CD. Drag them onto the computer and only if the USB disk is chosen will the password box pops out. The password is right on the wall. Input it, and you finally flee the computer.

6.3.6 Ending Interface

At last, it is ending interface. In this part there is a small animation, the animation shows that after shuttling out of the output device, the player successfully escaped from the computer, but the picture appears to be barren, which means that if the human over-reliance on the computer rather than reasonable treatment of the relationship with the machine, then in the near future, artificial intelligence will surpass the human. In the end, humans will be replaced by the machines they make. We hope that the audience of this game will have new thinking on the relationship between computers and humans after playing this game.

7 Evaluation

7.1 Playability Testing

Playability testing is a key part of the complete game development cycle. It exists not to find problems in the game's functionality and performance (even in playability testing you can also find these problems), but to verify the game's design by analyzing the player's playing process, providing feedback, and finding and fixing problems in the game. The benefits are:

- Be able to make targeted adjustments to areas that do not meet expectations more quickly
- If the design objectives of the game are well defined in advance, it is easier to find something unexpected in the test. Even failing to find answers to these questions after testing can help you plan for future testing

7.1.1 Participants

Our participants can be roughly divided into the following three types:

- Developer Tester： This type of tester comes from the game's internal development team. They were also the first testers of the game. Designers based on the existing development experience and reference of the same type of game, find out the problems in the game design, these opinions for the overall direction of the game plays a vital role.
- Kleenex Tester： This type of tester refers to a person who has no prior exposure to the game and who is not likely to play the same game again. A limited understanding of the game tends to make it easier for the players to spot confusing aspects of the game, and their feedback is an important basis for assessing whether the game is attractive enough to new players. So, we chose eight kleenex testers to play our game.

7.1.2 Questionnaire

After testing the game, we invited players to fill out our questionnaire. The questions of questionnaire are divided into three types, multiple choice questions, blank-filling questions, and scoring parts. According to the data analysis of the questionnaire filled by players, we will modify and improve the detail of the game.

More than half of the participants were between the ages of 20 and 25. Among the participants, seven finished the game within 5-10 minutes, and only one finished the game within five minutes. The majority of participants think the memory room is the most challenging room because most of the participants fail to associate the answer to the question with the password for the opening. Some participants thought the game should have more prompts. Another participant said he liked the ending of our game, which he said made the game educational.

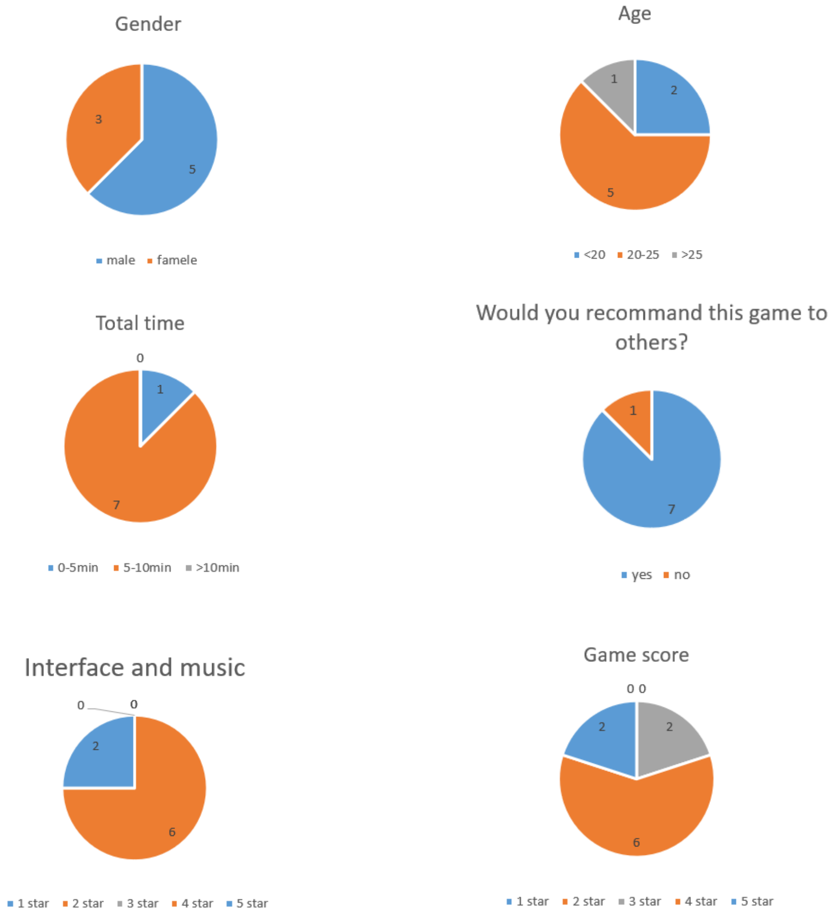


Figure 26 Result of the questionnaire

7.2 Sound Testing

There was also a survey on the background music and the sound. It is said that in the rooms, the style of the background music kind of conflict with the style of images. The confliction made the ambiance a little weird which corresponded with the story. The sound effect when completing all puzzles were said to be clownish, making the player feel relaxed. The new background music after it was said to be hopeful as congratulation to the player

8 Challenges and Achievement

8.1 Challenges

8.1.1 New Language

The biggest challenge is we need to learn new software and language by ourselves. Because our team had no previous experience writing games and had never used Unity or C#. This presents us with a very big challenge. Each of us spent a lot of time learning about software usage and C# language specifications from scratch. In the early stage of the project, we found unity's teaching video on the Internet and share everyone's difficulties and experiences by assigning regular learning content on a weekly basis and meet twice a week. Therefore, we overcame this big difficulty.

8.1.2 Story Designing

In the game design process, we want to reflect the theme of escape from the computer, but also want to make this game educational, so in the design of each room and the corresponding room level, we have added a lot of computer-related knowledge. For example, the conversion of the base, the basic loop logic of writing code, the concept of input devices, and so on. In addition, because we are a room escape game, we hope our game has a certain playability, so we spend a lot of thought to set many levels, in order to escape the room must pass these levels.

At the end of the game, we think of a very profound ending, which is about the relationship between humans and machines. If humans overuse computers, then shortly, artificial intelligence will replace humans, and the world will be full of mechanical ruins. We hope that through our games, more people can rethink the relationship between humans and computers and not over-rely on machines.

8.1.3 Interface Designing

Because the game is developed for middle school students, so in the design of the game interface, we hope our game is colorful and attractive. We also hope our graphics to match our computer theme. As one of our members is very good at painting, we chose bright colors to draw many computer related devices original parts during the painting process and used Adobe Illustrator software to combine our painting materials to form our room.

8.2 Learning Outcomes

8.2.1 C# and Unity

To build a complete game has always been an aspiration for a computer science student. We did not know how to get into this field, how to learn effectively, or even how to learn it. Finally, we chose C# and Unity, starting from zero.

C# has a similar name with C, but to us, it is more like Java with some feature of HTTP. When we first found “using” instead of “import”, we just wonder whether every programming language has its figure of speech to call external classes. We even came up with some more possible words like “calling” or “saying”.

Unity provided us a new platform to visualize graphics we constructed in C#. We are luckily able to drag the button where we wanted it to be, or expanded it to the expected size, instead of repeatedly modify its attributes with plenty of times running the interface. Another advantage we found is that Unity is really to learn for a starter. There are lots of video tutorials online with instructions of every step.

It is quite convenient to link the Unity component with a C# script. As each C# script is individual, it is easily tested with a black-box strategy, which is the primary type of testing in this project. After adding a C# script link, e.g., with functions of dragging, we just checked whether it could be dragged as the mouse moved, whether its shape changed as expected when being dragged, and whether the sound effect acted right.

8.2.2 Graphics

We made images applied in the project with the help of Adobe Photoshop, Adobe Illustrator, Microsoft PowerPoint. At first, we thought about Gimp, but we felt more skilled in PS for we had much more experience in it. We made some reference with some open-source pictures from a Chinese website and then made our images.

We discussed the overall style of our pictures, including the background image of the starting and ending interface, the layout of each room, and different button styles. Finally, we chose a style which is a little gloomy at first but brighter in the rooms, in that our story is about being tracked in the computer and saving the world after succeeded.

8.2.3 Sounds

We got open-source music, including ambiance songs and sound effects, from Chines websites. Similar to the graphics, the sounds should also be in correspondence with the overall style of the story. We made the background music quite gloomy not matter in the starting or game interfaces. Though the background image in the game interfaces is brighter, the gloomy background music makes the ambiance style closer to the sound.

After escaping, there appears a clownish sound which conflicts with the room background music so much. We intended to relax the player after a series of intense puzzles. Following it, there is a comforting music which congratulates the player for completing the game.

8.2.4 Teamwork

This project can be divided into several parts, each with part of the workload. We started learning C# and Unity together and shared ideas of this project with each other. One of the most difficult parts of C# programming was in Input Room, where different kinds of stuff are to be dragged into the box. Members in the team discussed how to make the stuff appear at a certain location, as well as the logistics about the judgment whether the player has collected all four right ones.

Apart from programming, another workload is assigned to each member. One member who is an expert in Adobe Photoshop and Adobe Illustrator was in charge of image making. One member who has a keen sense of aesthetics was in charge of the modification of the layout of each interface. One member who is good at programming was in charge of question sets. One member who is enthusiastic and friendly was in charge of cheering others up when they were in trouble. One member who was good at the speech was in charge of the presentation.

During the process, we encountered plenty of problems. Usually, we raised the question in the WeChat group and discussed it together. When the question is hard to solve, we booked a study room in Queen's building and tried to figure it out together.

8.2.5 Testing

Game testing has some common characters in software testing. It aims to find deficiencies in the software. The testing was implemented by the instruction of the game. During testing, we found some bugs, e.g., in the Input Room; it popped out the next room after dragging the right stuff anywhere. Meanwhile, we did some modification on the location where each stuff was originally put to make the interface more harmonious.

To be honest, we did more modification than we expected. When just finishing the first version of the game, we thought we did a quite good job for we did overcome a lot of difficulties and thus each one of us felt satisfied. However, when starting testing, pretending we were the children this game was oriented, we found lots of deficiencies.

With more times of testing, we found more deficiencies, and it seemed that we could always be unsatisfied as long as we imagined we were the children who played the game for the first time.

8.3 Main Achievements

After many times of modification, we believe that the final version of the game has a storyline with complete logistic. The educational parts are contained in the start, process, and end of the game.

In each interface, it is able to transfer data between the player and the logistic controlling system behind it. The interface tells the controlling system the instruction the player gave, and the controlling system processes the instruction and tells the interface what should be shown on the screen, as well as the sound displayed.

While programming, we managed to call classes within other packages, instead of making every function by our own. At last, we leave the API of the game available, with which we are able to make further modifications or improvements.

9 Conclusion

Unity is an excellent platform for C# application, especially when the project is concerned with human-machine interaction, e.g., a game. With Unity, it is quite convenient when it comes to the location, size modification of a specified image, as images are visible in the interface. Indeed, the interface of Unity makes it clear which category different C# scripts behind it belong to, as it lists out plenty of possible functions. What’s more, it is quick to switch between different interfaces.

Both image and sound have a lot to do with the overall style of the game. In one interface, images on different layers should share a similar hue, or the image with a lighter hue would seem abrupt, and the user is likely to ignore the others. When the background music has a different style with the images, the overall style is likely to be closer to the background music.

The overall programming idea of the project should be the result of the discussion of all group members, e.g., the intention of each function. It would be more convenient for the following modification after the first version of the project.

Usually, it takes more than once of modification before the final version of the project is obtained. The modification is a long process, and group members may have different opinions on it, as one member may have a different view of aesthetics with others. For instance, the button size and location of the start interface have a significant impact on the appearance. One member may think that the button should be large and at the center of the interface to attract the user at first sight, while the other member may believe that this way the user might just ignore the background image. Each group member should take others’ opinions into consideration and be confident that the final version of the project would be a better one.

10 Reference:

[1] Honey, M.A. and Hilton, M., 2011. Learning science through computer games and simulations. *National Academies Press*.

[2] Negrete, A. and Lartigue, C., 2004. Learning from education to communicate science as a good story. *Endeavour*, *28*(3), pp.120-124.

[3] Barendregt, W., Bekker, M.M. and Speerstra, M., 2003, September. Empirical evaluation of usability and fun in computer games for children. In *Proc. IFIP Interact* (Vol. 3, pp. 705-708).

[4] Inkpen, K., 1997, June. Three important research agendas for educational multimedia: Learning, children, and gender. In *AACE World Conference on Educational Multimedia and Hypermedia* (Vol. 97, pp. 521-526).

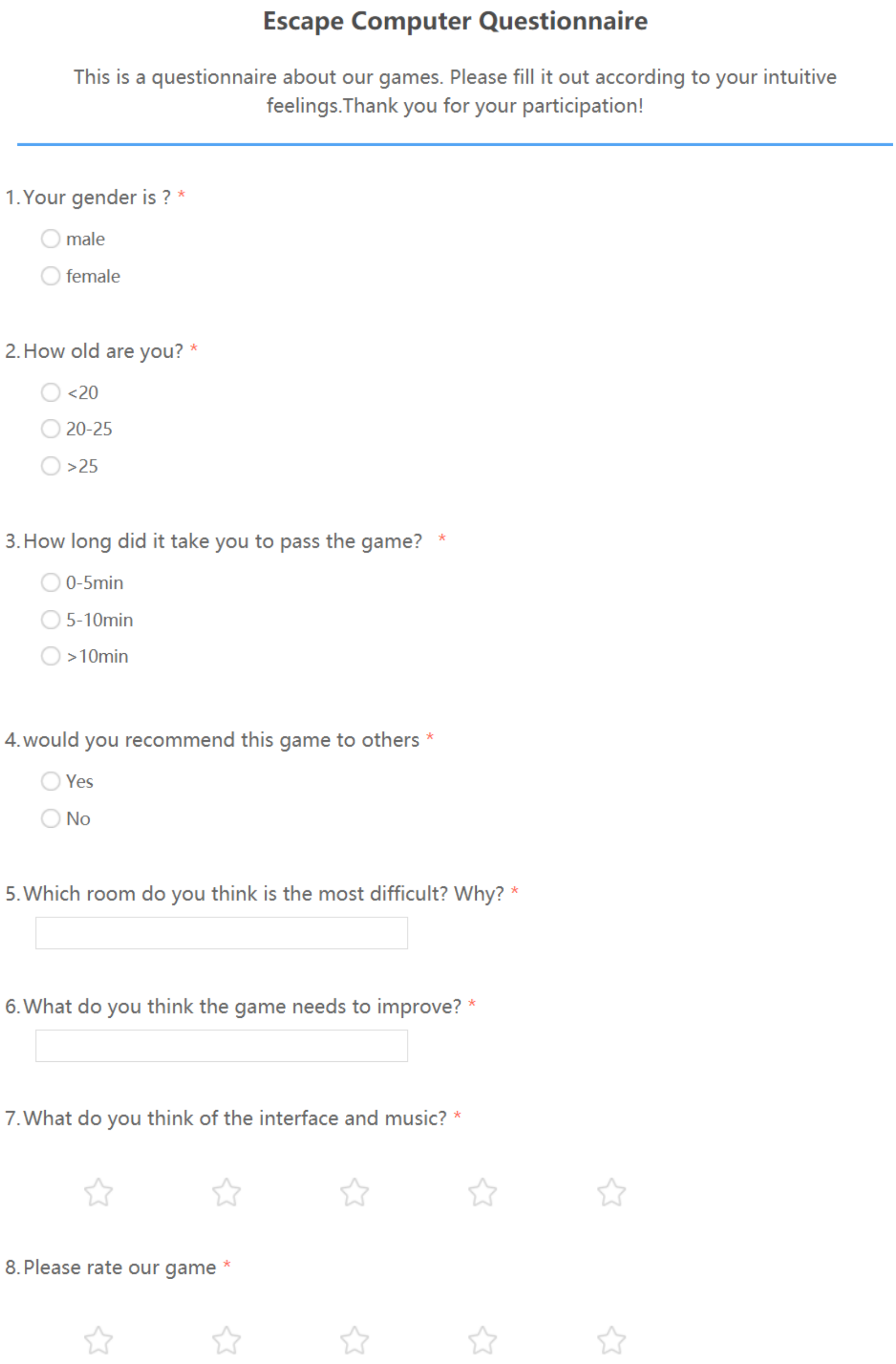
[5]Wiemker, M., Elumir, E. and Clare, A., 2015. Escape room games. *Game Based Learning*, *55*.

11 Independent Statements

We declare that the code of the game and the report are completed by our own. All team members agree that this report reflects their experience with the game exactly, so we chose not to write an individual report. We have worked together, learned more knowledge and achieved the desired results.

Figure 27 Our team photo

12 Appendix



Escape Computer Questionnaire

This is a questionnaire about our games. Please fill it out according to your intuitive feelings.Thank you for your participation!

1. Your gender is ? *

○ male
○ female

2. How old are you? *

○ <20
○ 20-25
○ >25

3. How long did it take you to pass the game? *

○ 0-5min
○ 5-10min
○ >10min

4. would you recommend this game to others *

○ Yes
○ No

5. Which room do you think is the most difficult? Why? *

6. What do you think the game needs to improve? *

7. What do you think of the interface and music? *

☆ ☆ ☆ ☆ ☆

8. Please rate our game *

☆ ☆ ☆ ☆ ☆