MATHSRONPA 1.0

BY

SAMIG

SHERCHAN

Index:

3.1. Analysis of the problem

Page 003 Problem Identification

Page 005 Stakeholder

Page 009 Research the problem

Page 011 Specify the proposed solution

3.2 Design of the solution

Page 013 Decompose the problem

Page 016 Describe the solution

Page 022 Describe the approach to testing

3.3 Developing the solution

Page 026 Iterative development process

Page 140 Testing to inform development

3.4 Evaluation

Page 147 Testing to inform evaluation, Success of the solution, Describe the final product, Maintenance and development.

Page 151 Teacher Comments

Page 154 Refferences

Samig Sherchan

3.1. ANALYSIS OF THE PROBLEM (10 MARKS)

My name is Samig Sherchan, my idea is to make a game with interactions for the user to communicate and learn from the game which will help fix the problem currently occurring. The location of the problem is at my school Rosedale College; the problem has been troubling quite many students so my aim is to reduce this problem from reoccurring constantly.

The problem is based on A Level subject; AS Maths as new students are more likely to struggle adapting after a huge jump from GCSE Maths. Maybe the first few chapters might be GCSE basics however it can get confusing easily. It's a subject that people intend to underestimate when starting Maths however if you do have control over the understanding it can be easy. I personally had similar experiences; I don't want new student to repeat the same mistakes and let the problem increase. So I thought of methods, physically and computational methods that will solve this problem. Based on last years' experience, I gained plenty of knowledge on coding through my teacher's lessons. I started off with universal language Java to personally learning basic of Python, I also got introduced to coding software. Then I researched on educational games based on maths and looked through software which can help me construct my solutions. Additionally, the lack of understanding in A Level can lead to unexpected low grades, even grades like U. Most students wouldn't want that to occur. From the recent results day there had been plenty of student in Rosedale Grade who got U grade in one of the maths unit. I personally researched my classmate's performance and compared their GCSE grade and their result to A Level. Those with A* in GCSE couldn't even managed to get an A overall in AS Maths. One or two students did, due to their preparation and concentration. Those new students also need that preparation and concentration which the solution I am constructing could make it occur. This problem is easily amenable through computational approach because most stakeholders have a highly usage in games. Furthermore, my game can easily ask questions for the users to solve with given multiple choice questions. So I can also keep the coding simple as possible, as the solution is mostly text based so I don't need to use conditional statements every time. Moreover, the solution doesn't involve any cost as the problem was just the understanding of maths, I just needed to make the understanding to new students more clear, with advice from my experiences.

Since my solution has mathematical background, my potential stakeholders would be our math teachers from Rosedale College, they might be interested on my solution and possibly promote it to my school. It could be a great assistance to teachers as well, as some students might be struggling to understand the traditional teaching of mathematics.

Samig Sherchan

My solution can improve their understanding and educate information they might have missed. Another stakeholder that is definitely be interested and gain benefits from this solution is students themselves. Students in school are constantly talking about games, new releases or something special that occurred in a game. So I thought a game would be really effective towards students; many educational game is childish which can fend off student's interest so I thought I would make the game with dark contents which will match their interests so the age rating is quite high so for young student it wouldn't be suitable for them except older students. A game that has a combination of maths and murder case, student's curiosity would really grow when playing this game. They would definitely want to solve the maths question no matter what, which is what I am trying to encourage without the student knowing it consciously. Consciously, most students would lose interest easily, if it was just a maths game, which they wouldn't learn much. Furthermore, public might say games are bad and it promotes violence and other problems however based on the information from a newspaper company 'Independent' that "high video game usage was associated with a 1.75 times increase in the odds of high intellectual functioning." This also can help build up social skills especially it can hone up student's confidence while talking to professionals. (http://www.independent.co.uk/news/science/video-games-children-learning-intelligence-social-skills-study-a6920961.html

Samig Sherchan

Too understand the opinion of my user groups, I had to conduct a questionnaire which will help me develop my game and understand where in the parts of the problem I can decompose and come up with the solution. So I asked around 10 questions to group of new sixth formers which just had completed their GCSE, especially GCSE Maths despite the fact the specification had changed now. Here are the questions:

- Do you enjoy maths?
- Do you like solving murder case games?
- How often do you play games?
- How often do you study?
- Do you think Maths is beneficial?
- What grade did you get in GCSE Maths?
- Do you think interactives are better way of revising than the traditional method (writing from the book)?
- Which exam board do you think has the hardest question for maths?
- 'Past papers are the best method to study for exam'. Do you agree with this statement?
- Pina is currently thinking about her future, but she is confused on whether to study Maths and if it's useful or not for her future, she stated "Maths is useless, I can't really get anything or anywhere from it!" What do you think she should do?

Most of this question are past related rather than to what they are going to do now, but learning from their past can help me identify the mistakes that are likely to be made in A Levels. So it makes my solution easier to create. So I started to do an initial interview asked around 7 people from the sixth form who was willing to contribute which I asked these question, some were continuing maths at A Level while some students were dropping maths.

I didn't want to just focus on maths student but wanted to hear from other students as well however majority of interviews was aimed at maths students. So I gave myself 1 week to conduct the interview as I didn't want to take much time as it would waste time for other stages. The group I interviewed were quite honest with their answer and quite interested to see the final product of my solution.

Their names were Kierthigan, Ashley, Chisa, John, Kate, Abdi & Ciel. They didn't want to write their full names unfortunately due to privacy concerns. Down below show the results in text format and graphically as well:

Samig Sherchan

First Question:

- **Yes (3)**
- **No (1)**
- **Maybe (3)**

Second Question:

- **Yes (4)**
- **No (2)**
- **Maybe (1)**

Third Question:

- **Under 1hr (3)**
- **1hr to 2hrs (2)**
- **2hrs+ (2)**

Fourth Question:

- **Under 1hr (3)**
- **1hr to 2hrs (2)**
- **2hrs+ (2)**

Fifth Question:

- **Yes (3)**
- **No (1)**
- **Maybe (3)**

Sixth Question:

- **A* (2)**
- **A-B (3)**
- **C and under (2)**

Seventh Question:

- **Yes (3)**
- **No (2)**
- **Maybe (2)**

Eighth Question:

- **AQA (1)**
- **Edexcel (1)**
- **OCR (5)**

Ninth Question:

- **Agree (4)**
- **Disagree (2)**
- **Maybe (1)**

Tenth Question:

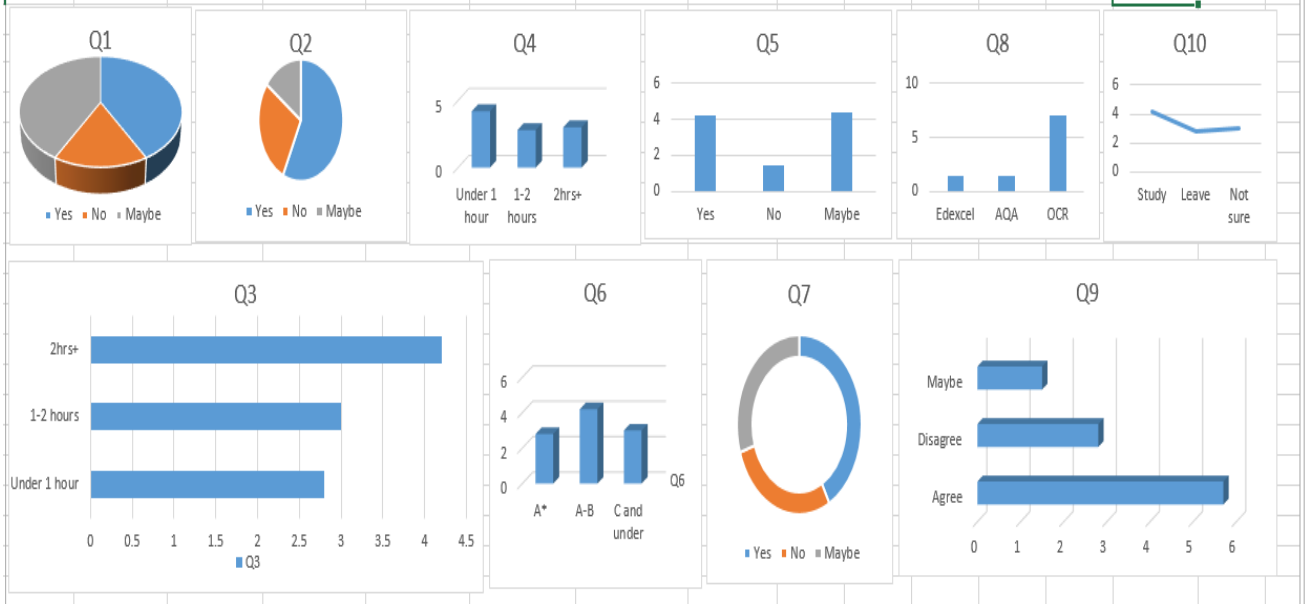
- **Study Maths (3)**
- **Leave Maths (1)**
- **Not sure (3)**

Samig Sherchan

Based on the interview conducted it seemed most students think OCR is the toughest exam board in Maths. Furthermore, the grade the students got in GCSE also leads to an increase chance of the student choosing maths for A Levels. Furthermore, it also has an effect on how the other question are answered. Also, the results are showing that it's better to make it game dominated rather than maths because the user group game usage is quite long.

	Kierthigan	Ashley	Chisa	John	Kate	Abdi	Ciel
1	No	Maybe	Yes	Yes	Maybe	Yes	Maybe
2	Yes	No	Yes	Maybe	Yes	No	Yes
3	2hrs+	Under 1hr	1hr to 2hrs	2hrs+	1hr to 2hrs	2hrs+	Under 1hr
4	1hr to 2hrs	Under 1hr	1hr to 2hrs	Under 1hr	2hrs+	2hrs+	Under 1hr
5	Maybe	No	Yes	Maybe	Yes	Maybe	Yes
6	A-B	C and under	A*	C and under	A*	A-B	A-B
7	Yes	Yes	Maybe	No	No	Yes	Maybe
8	OCR	Edexcel	OCR	AQA	OCR	OCR	OCR
9	Agree	Agree	Disagree	Agree	Agree	Maybe	Disagree
10	Not sure	Leave Maths	Study Maths	Not sure	Study Maths	Study Maths	Not sure

Name
Kierthigan
Ashley
Chisa
John
Kate
Abdi
Ciel



Here is the graphical format of the result, most answers have 'Yes' and 'No' equally shared. Meanwhile, at question 8 the option 'OCR' dominated the question; most likely due to their understanding through social media or never experiencing any exams from OCR. All results are in different graphical format, with different colours to indicate each user. Most answers are correlated with other answers. People who aced their GCSE at question 6 more likely to like to enjoy Maths (answered in question 1). Furthermore, their GCSE grades resulted on whether they are going to continue maths in A Level.

Samig Sherchan

Additionally, here is also the result of the personal interview with our maths teacher. She had well detailed explanation about her thoughts on maths and students as well as the game I was planning on making.

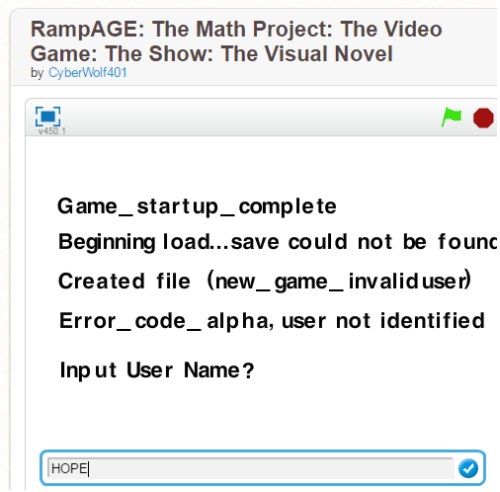
	Question	Reason	Response
1.	Do you think the school struggles in Maths the most?	This can help understand if Maths is the weakest or strongest subject in the school. Statistics can set the difficulty of my project.	No, I do not think Maths is the weakest subject. For the past few years, 78% of our student have been achieving A*- C in Maths. I think we are steadily improving.
2.	How do you feel about Maths specification changing?	From 2017 Maths will be all in one, so the specification will change, this could make it harder similar to the structure in GCSE Maths.	I personally think, this is going to make it harder for the upcoming students from GCSE. Students may be confused however they will be familiar with the GCSE structure so I don't think it will affect the new students too much hopefully.
3.	Do you like teaching maths?	This question is bit personal, but I want to see the teacher point of view as well.	That is a good question, I been teaching maths for a decade, it has been fun educating new student and old. Good experience. However, I guess there are one or two students that misbehave or don't understand which can be troublesome.
4.	Which continent do you think is dominant in Maths?	Well there many students around the globe, every continent has their own understanding in maths.	Statistically, Asians are shown to be better due to tougher education and more learning, however if you have the dedication to learn maths you can achieve anything, so you don't have to be an Asian to be good at maths, but practicing is the key.
5.	How much homework do you give a week?	Homework can be boring and tiring but more you learn the better?	Yes, I personally think Homework is the best way to enhance my student's performance, they might moan and complain however it has increased their performance in class as well as exams.
6.	Do you think the best way of improving is all Teachers responsibility or students themselves?	A lot of student depend on their teachers to teach them everything, while the teacher's job is to educate but students must be doing their own extra revision too right?	Yes, that's right. Our job is to educate the students however the students must revise at home on what they did as school, as soon as you go home you'll slowly start to forget. So it's good to revise and it is partially student's responsibility as well.
7.	Do you advise students to pick maths in A-Level?	I mean choosing an option in A-Level can be tough plus maths becomes optional in A-Level.	Yes, I do try to advise my students to choose maths. Maths is known throughout the world, it's easier to get into universities. However, it is the students decision.

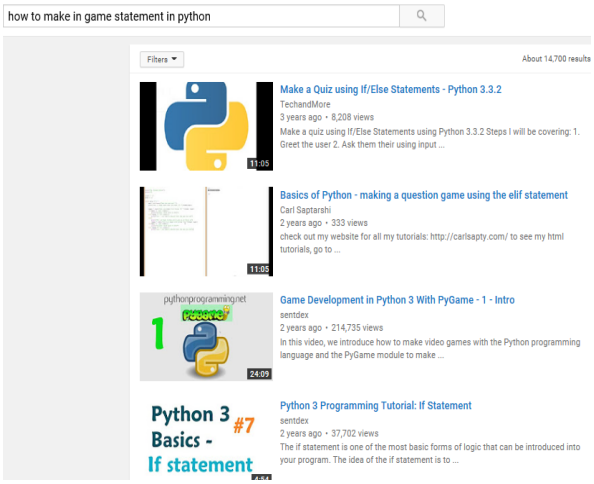
First of all, based on maths' teachers opinion, maths isn't the weakest subject in Rosedale College, instead its one of the strengths in this school. Around 78% managed to gain A*-C grades successfully. Plus steadily the school is getting better each year. Furthemore, maths is also changing starting next year, the whole specs are changing, and our maths teacher views was the exams will be more pressured and students might struggle a bit however the most of the topics will remain the same, so the teachers will have no difficulty teaching the new specs which puts the relief for both students and teacher. There is also correlation that practicing will improve mathematical skills despite where the origin of the person. This really supports my idea as the game is a form of practicing and it will also help improve the students.

Furthermore, every solution must have essential features before it can even start. Based on my knowledge of visual novel I had some experience playing it before however I still had to research the features needed that I could've missed. Then I navigated to a forum website called 'Lemma Soft', where there was thread listing all the features I must include in my game. First of all, I needed it to be touch friendly and compatible in basic platforms like Windows and Android. Most audiences use portable devices so it's a good consideration to create a game for android as much as creating it for Windows. Furthermore, android requires less power and resources in order to run. PC would take some time to load and boot up, it requires more power than portable devices. Portable devices can be accessed anytime rather than just accessing the game at home. The solution also needs a simple and elegant animation language with good user interface. Keeping the game simple as possible will reduce complexity in coding as well as it will help user understand easily. Just in case it's safe to create a short manual which will guide on how to use the solution. The solution must also have readable text with simple background colour, as multicolour can be disturbing. Furthermore, if the text is too small or too big it can in-attract the users as well. Common games have a simple white text box with black text which is easy to read and simple as well. So, I wanted my solution to include that basic feature. Furthermore, every visual novel need a GUI, without that interface it's useless as visual novel is all about the interaction between the user and the game. Most visual novel maker already has prebuilt interface, which the programmer just needs to code and create the game.

Finally, apart from the limitation I faced, there was plenty of success in my work, I started to create the solution as a practice, it was rather easy than expected I just needed to learn few instructions and the rest was all up to me. I didn't really struggle because there was nearly enough sample of codes to reuse or understand if not then I had the official website with all information needed. Furthermore, the surveys and questionnaires I gave to random test samples in the school all had the result that supported my plan. Even the interview question to the math teachers thought the game was an interesting idea which could help the students understand better. I had to research the games in depth, even the existing and samples that will give me ideas when designing mine. I came across samples which related to my ideas, what I was interested on was the declaring statements, inputs and other conditional statements. So I searched on how to create an input in all languages by watching YouTube videos and reading Wikipedia.

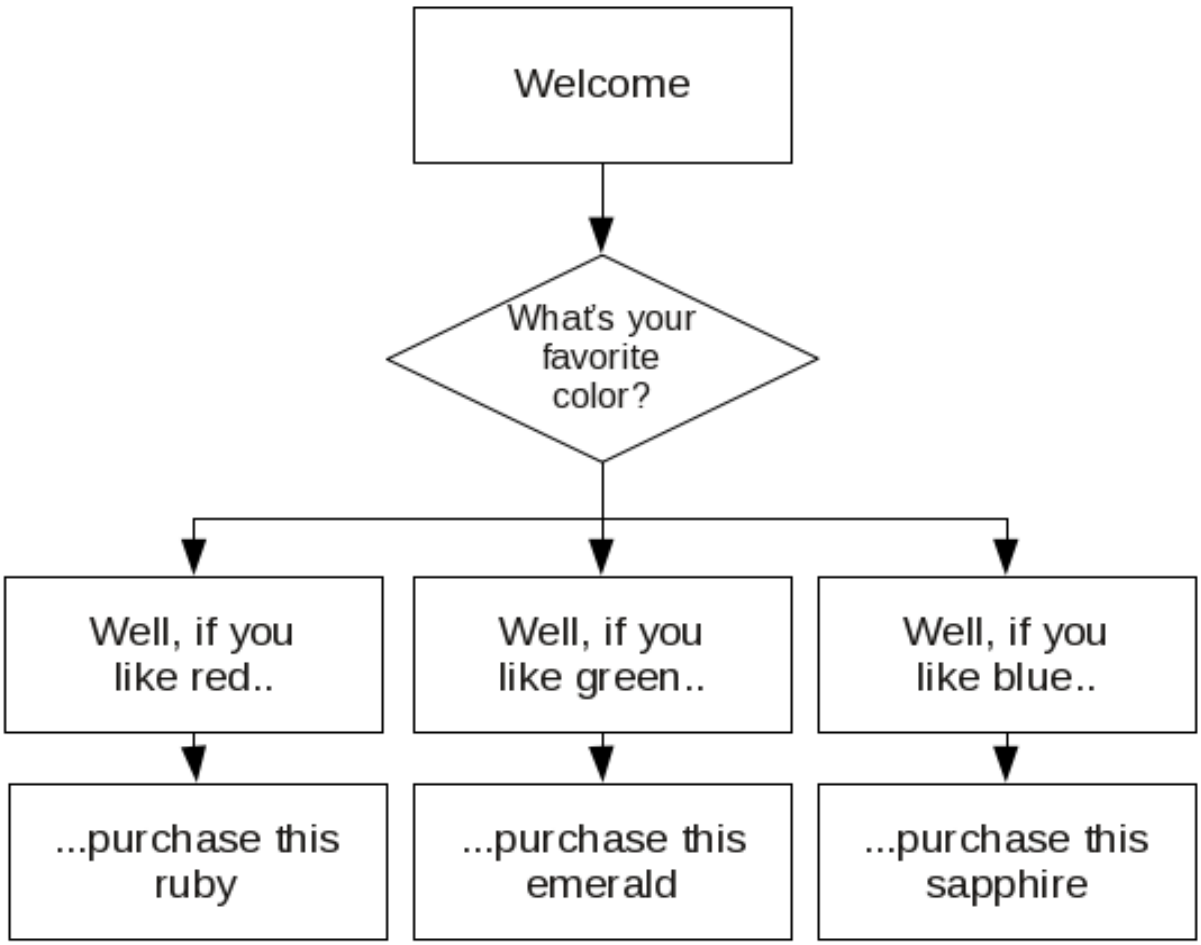
Samig Sherchan





The first sample was sourced from 'Scratch', it had a parameter where the user could input their names, plus the name would be a constant variable and will always appear in the game. So I had to learn some new variables as well. After countless of videos I finally managed to understand how to create a variable to make user's input constant, plus the data won't be lost as well. It did inclue some IF statements. For example, if the users puts an input then it will be constant, else if the input will be a default name set by me. I continued to study on types of games that I could make and a game that will make the most effective solution. I also searched for difficulties in terms of languages and simplicity, I wouldn't want a game with advance coding that I might never understand with also limited time given I had to go for something suitable. So, I decided to create a picture and text based interacting game which is commonly known as visual novels which are quite popular at eastern Asia. To make a visual novel I required a suitable software, so I decide to choose a Python based software called Ren'Py. Ren'Py doesn't only include Python, it does contain its own language to make it easy for the users. Furthermore, Ren'Py also includes important coding variables and statements. For example, IF statements, allows me to declare variables and include other conditional statements. It has equivalent statements to Python. However, for my script of the game I decided to dominate the script with python rather than Ren'py variables. Ren'Py wasn't the only software I had in mind, there was PyGames as name suggest it's also a python based but majority of the code is PyGame variables. After all Ren'py was built using PyGames so the features were the same. There was another easy software to create visual novel which was on Steam, however it didn't really require programing so it wasn't really useful in the end.
(https://en.wikipedia.org/wiki/Ren'Py)

Samig Sherchan



This is the basic features of many visual novels, as there are always multiple choices which leads to different endings. Each choice has its own endings; you cannot go back unless you reset unless it returns you back to the choices. The decision decides everything on how the story goes so it's always important to create new scenes and decisions if that particular choice was chosen. As the stories continues the decisions either become worse or better depending on what choices are chosen so there is always the save button to save you progress. Most users have multiple save progress due to their curiosity to find out on what happens if they take each choices.

However, creating the solution will require some hardware and software. The solution requires hardware like PC that has a suitable performance to use, alongside compulsory input devices like keyboard and mouse. The solution will also require the output devices to view the results, most of it is visual so monitor is useful, on the other headphones might be required in case of in game music as most users would want an audio playing rather than just staring at the screen. Music can help set the mood and heighten the interest of the user. I would also require a software that can run smoothly in my PC without taking much of the RAM, luckily the software I was going to use didn't require too much processing power, it was a simple software that can produce good results. I also needed to install code editor to code, after coding I also needed to access the results quick as possible however it takes time to load up. Another restriction I faced was accessibility of the software in some of the school computers. Some of the computers required administrator rights which I would need to ask the teachers or technician. Which they would give me permission to use it due to my project. So, it was quite troublesome due to restriction to most application in the school.

Software Requirements	Justification
OS: Windows XP SP1 or Later	An Operating system required to run Ren'Py.
Ren'Py	Software required to create visual novels.

Hardware Requirements	Justification
Processor: Pentium 4 1.7 GHz	Minimum processing power required to run Ren'Py and its product.
RAM: 128 MB	Minimum RAM required to run this software and games.
60 MB Hard Disk space required	Minimum space required to run this software as well as its games depending on objects and coding used.
Keyboard and Mouse	Hardware devices required to interact with the game and software.
Direct-X 9.0c	Minimum Direct-X required for the software, this handles tasks related to it.

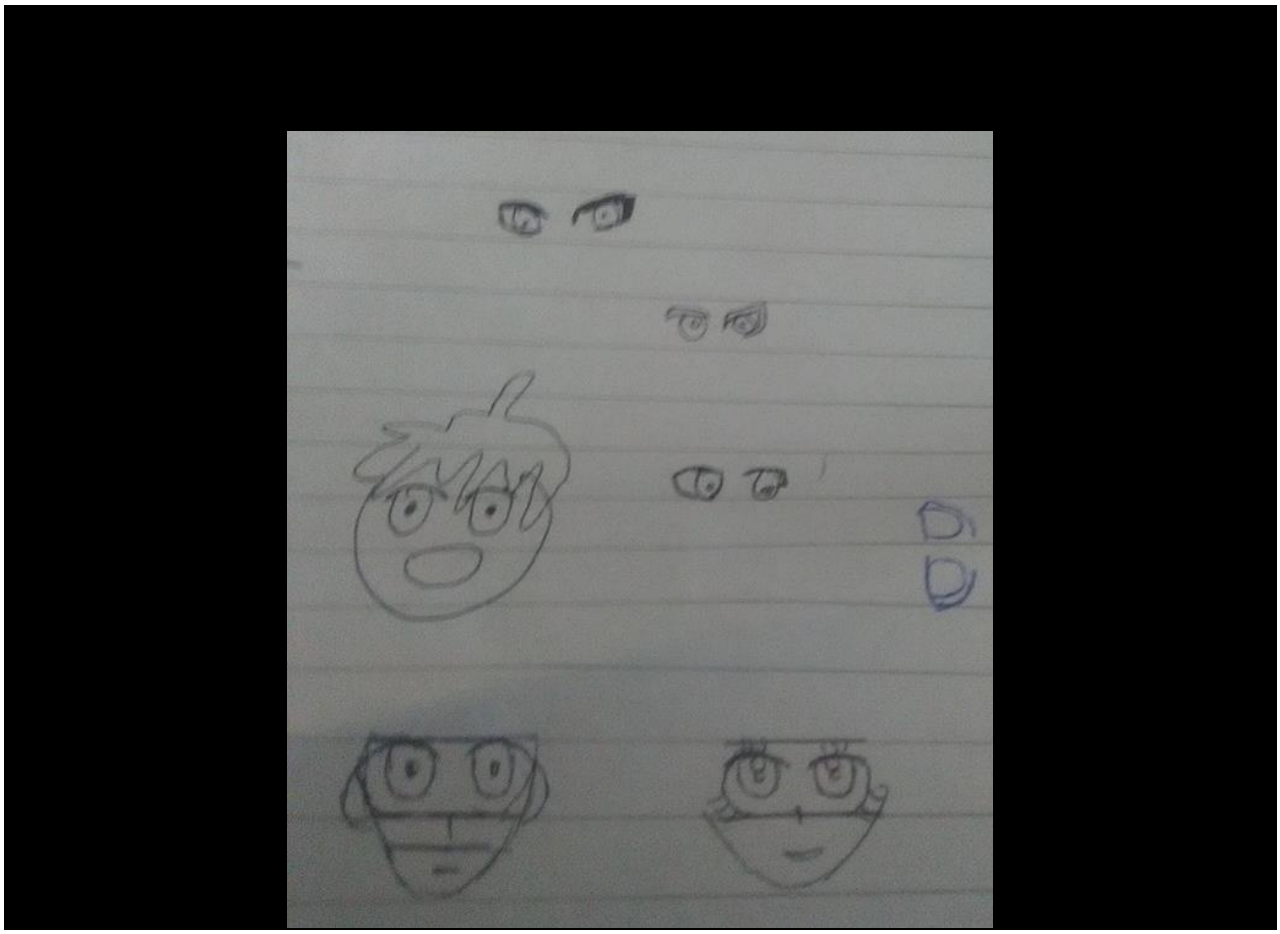
Samig Sherchan

Teacher Comments:

- Described the features that make the problem solvable by various computational methods and amended the approach.
- Identified suitable stakeholder for the stakeholder and described those stakeholders alongside how they will contribute in this proposed solution.
- Researched the problems in depth, looked at existing solutions that relate to the problem. Identified and described suitable approaches that aids the research.
- Identified and described the essential features of the proposed computational solution.
- Identified and explained any limitation the proposed solution might face when creating it.
- Specified the requirements for the solution including any hardware and software requirements.
- Identified measurable success criteria for the proposed solution. Also defined the purposed of the selected criteria.

3.2. DESIGN OF THE SOLUTION (15 MARKS)

Designing the solution wasn't easy as I thought it would, I needed to create resources for my solution, of course I can't really just give a block of information to my users telling them to solve a maths question this would easily bore them. I required to decompose it, and make the solution easy as possible. I needed to combine two resources in one; make the maths question appear while the background of game is running so it did took some time to learn it. I also needed different type of images and use other applications like photoshop and fireworks to make my solution dynamic as possible. Firstly, I required to create my design of characters, which I first planned it on paper, while I created list of objects to go to my game. Below in the image, has many type of prototypes before creating my actual design.

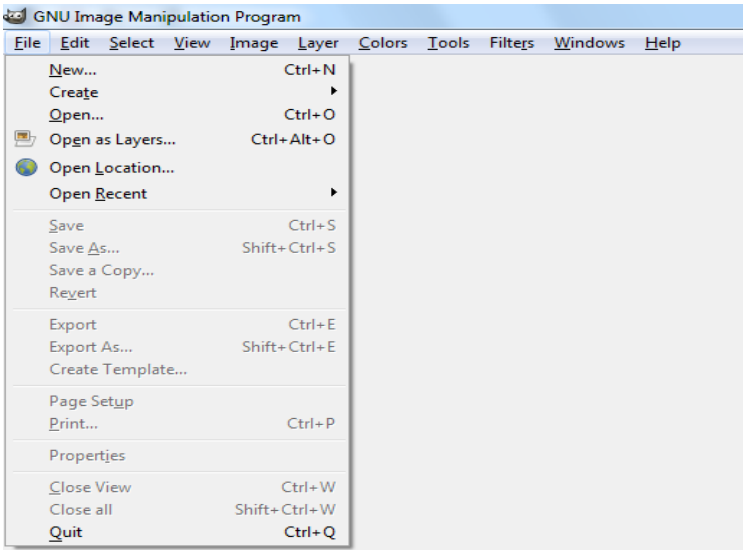


After intially designing the character in the paper, I needed an software to create my planned characters. So I searched in google to find the most suitable character designer that matches my art. Designing on the paper gave a brief idead on how my character will look. I didn't wanted to look realistic but more animated as my focus group would prefer animated images than realistic ones.

Samig Sherchan

So I found an inbrowser software/ game which allowed me to create characters with similar designs. It did have the most interesting options to choose from however it lacked a bit of flexibility while designing. I could only choose the given options, some designs I wanted weren't available there. However, it did let me edit compulosory fragments of my design so I could create any character with no hestitation.

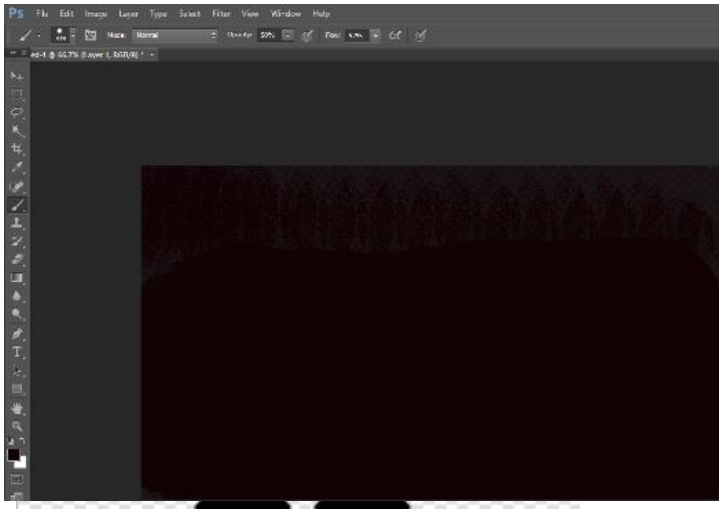
As you can see in the image next it, you can change the genetics/physique of the character, with different skin colours, nose types and etc. So all the characters can be unique rather than having majority of similar designs. The options also has misc where it can express the characters feeling however there were only 3 options given so that was a burden.

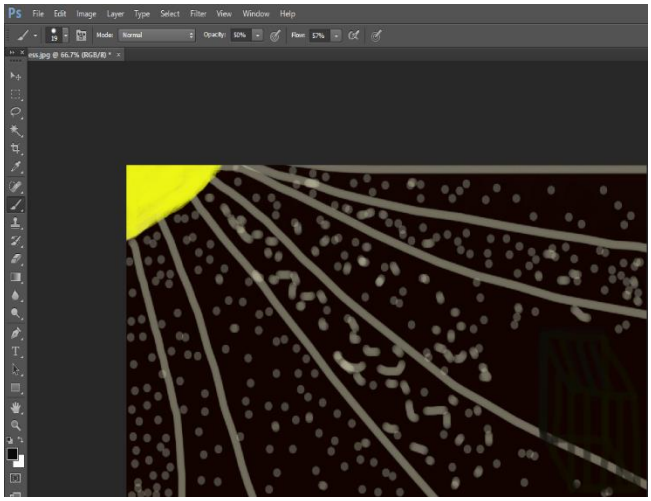


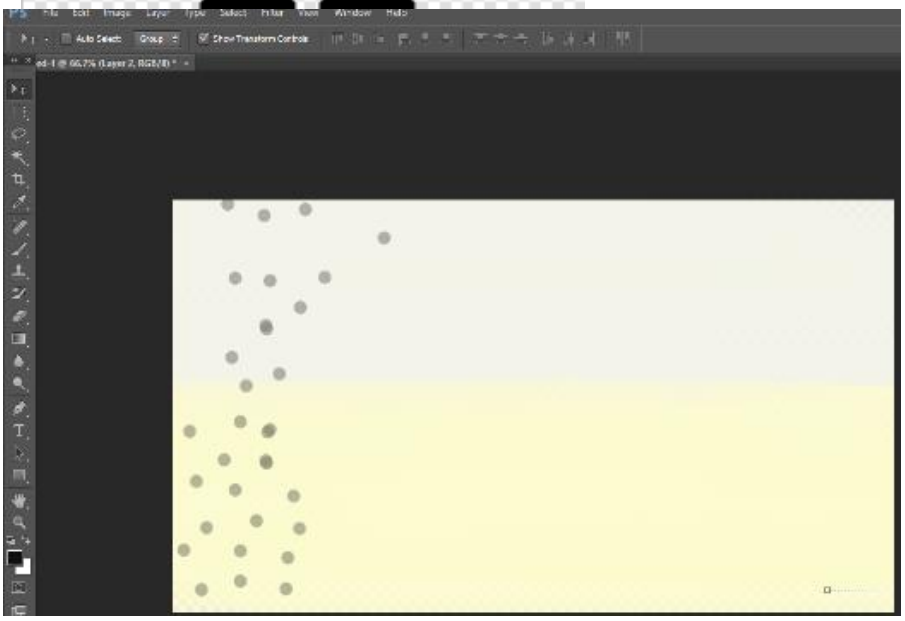
However, after saving the design, the file it was saved wasn't compatiple as image editing software Fireworks and Photoshop so I need to converted to pure jpeg or png file. So I used an pre-installed software in school computers called GNU Image Manipulation which read the file then I quickly exported the file in png format which later on let the photoshop read the file with no errors.

It was an easy process that actually designing the actual character. First I openned the image file created from the browser, without really using any features from the GNU, I just went File which listed all the options I could choose from, then I clicked Export, which directed the window to a different window where I could choose any file extension. So I converted this file to JPEG as its loseless and won't lose any data of my characters or become pixilated. I continously repeated the process while creating all others characters. It was an easy process however it did take some time.

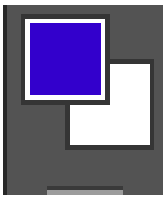
Samig Sherchan





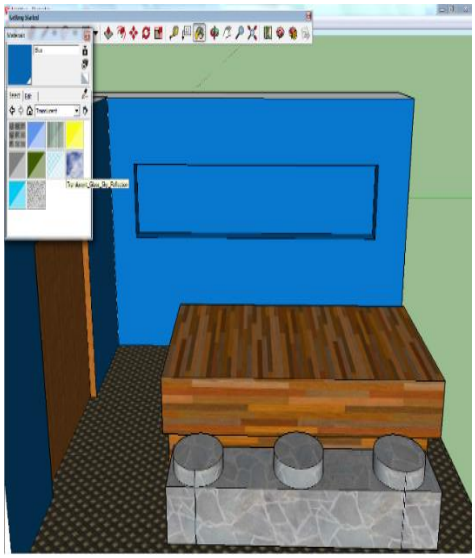
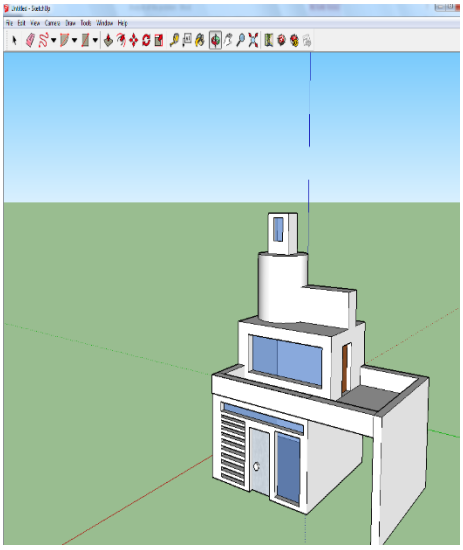
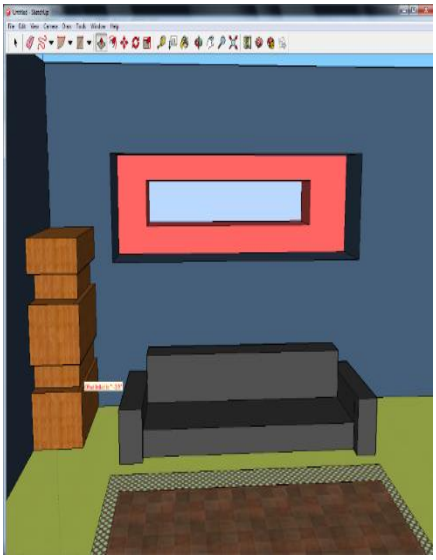


Then I continued to create other scenes built from scratch for my scenes in the game, I used tool such as brush, colour balancing and etc in Photoshop. I also had to make the background size 1280 x 720 pixels so it would perfectly fit at the game so there is no white space. As the software could cause problems if the size isn't perfect.



I used the colour change here to gain different colours and create different scenes. When overlapping the brush it managed to makes the background darker, so there is a contrast of the each layer.

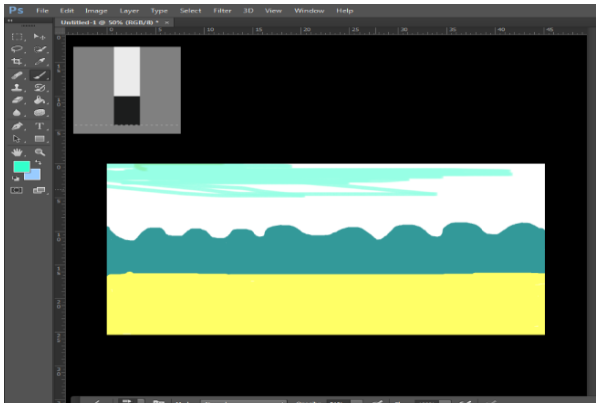
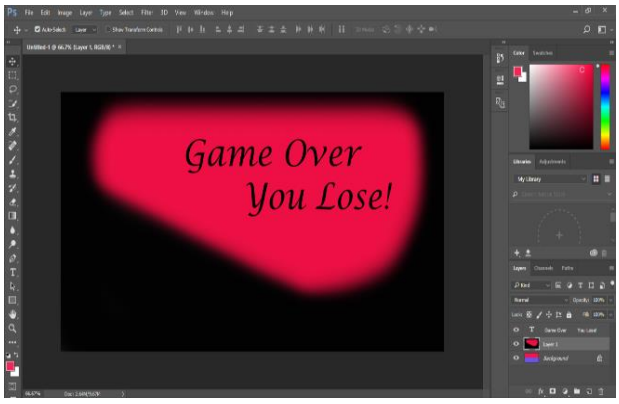
Samig Sherchan

For in-game rooms (scene) I use SketchUp to create 3D houses and objects, for example the middle image is the beach house which will be shown as a scene in the game. The other two images will also be used as scenes, using SketchUp made it easier to create rooms, It game me full flexibility to create anytype of shapes and change its depth to make it look realitistic and 3D as possible. Although some objects were hard to create and took some time to make it good.

The currently selected tool was used to make shapes, and it can be changed to other shapes as well, like square or rectangle. Other fuciton like the hand can move around the whole image. The 6th image is used to change the depth of the shape to make it look 3D.

I also had to create other scenes for the ending, once again Photoshop is easy to use so it was easy to create scenes of beach. Plus the ending credits of the game if the

user manages to lose. All my objects and resourcese were ready to be used in the game development. After understanding the tuturiol and examples, I decided to start coding in the software, of course the Ren'py is based on Python language, additionally it has its own statements.

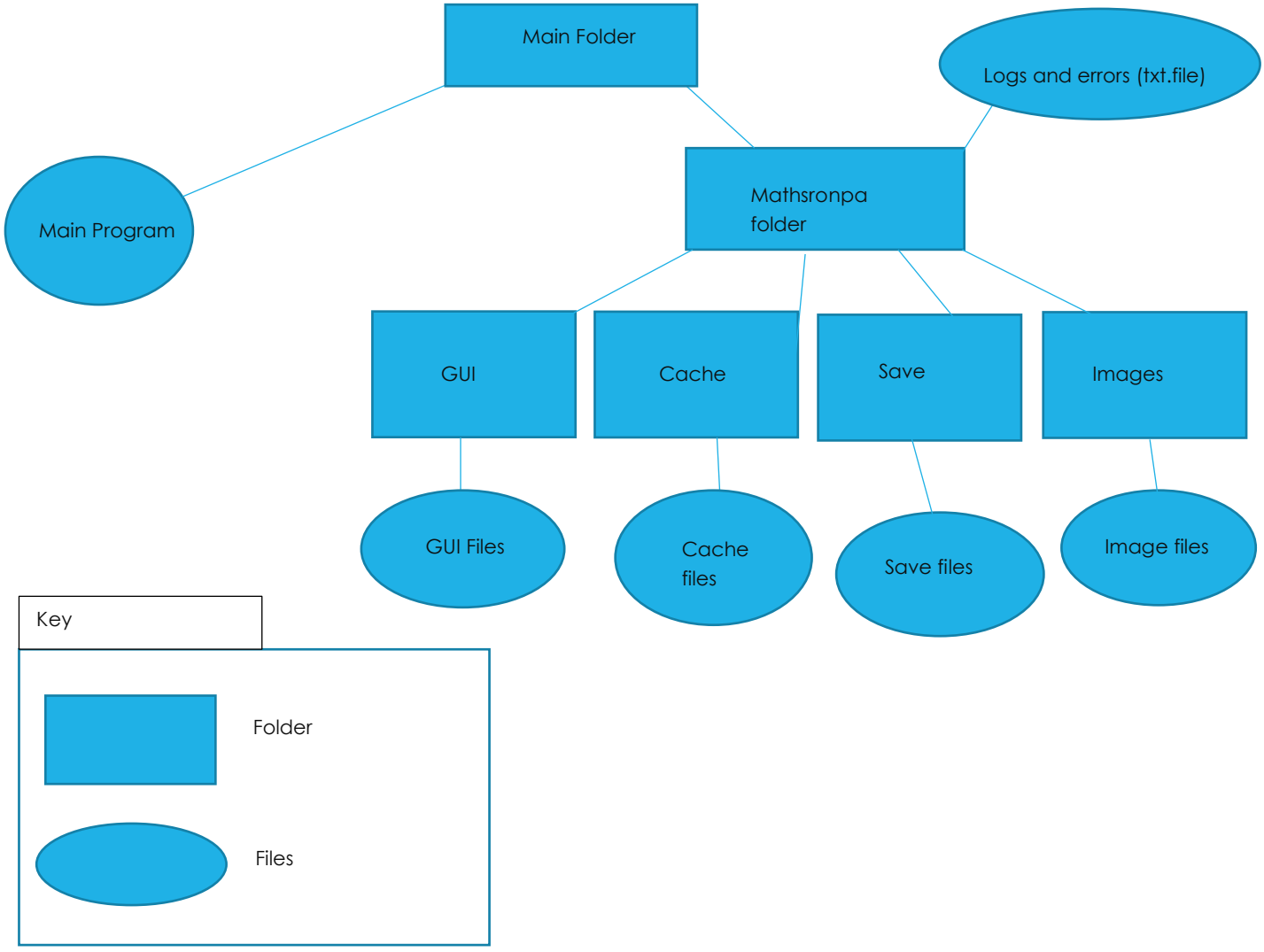
Samig Sherchan

Variable Name	Type	Size	Description	Sample Value	Validation
m	Char	8	This char defined as a character.	N/A	N/A
label start:	Long.	N/A	This the label for a scenario and dialogue.	N/A	N/A
scene1	Backg orund #1	N/A	The first background.	N/A	N/A
monokuma	Image	N/A	The main villan's image.	N/A	N/A
pov	string	N/A	The users input is defined as character.	N/A	N/A
povname = renpy.input	N/A	N/A	Used define the users input.	N/A	N/A
povname = povname.st rip()	N/A	N/A	Stores the data's input and uses it for the game.	N/A	N/A
menu start_from:	Boolea n	N/A	The first ingame menu for instructions.	N/A	N/A
$ question	N/A	N/A	One of the choices is defined here.	N/A	N/A
if question	IF Statem net	N/A	If statements for the choices made.	N/A	N/A
label looking:	string	N/A	A new label for the next scenario.	N/A	N/A
call start_from from _call_start_fr om	N/A	N/A	Calls to the game menu function.	N/A	N/A
pr	Backgr ound	N/A	The next scene for the game.	N/A	N/A
label new:	string	N/A	A new label for new dialogues and scenario.	N/A	N/A
dr	Backgr ound	N/A	New scene.	N/A	N/A
rd	Backgr ound	N/A	New scene.	N/A	N/A

lr	Backgr ound	N/A	New scene.	N/A	N/A
n	char	8	This is defined as a mysterious character.	N/A	N/A
beach	Backgr ound	N/A	New scene of a beach.	N/A	N/A
intro girl	Image	N/A	The image of character (blured) it will transit using disolve to 'girl'.	N/A	N/A
girl	Image.	N/A	The image of the female character after getting disolved intro girl.	N/A	N/A
p	char	8	The letter p is defined as character Lucy.	N/A	N/A
label searching:	String	N/A	New label for the scenario.	N/A	N/A
bh	Backgr ound	N/A	The new background of the beach house.	N/A	N/A
inside	Backgr ound	N/A	The scene from inside of the house.	N/A	N/A
fg	char	8	This is also defined as the mysterious character.	N/A	N/A
cha	char	8	This is the image of the character Alphonse.	N/A	N/A
label noturn:	string	N/A	New label for a new scenario.	N/A	N/A
f	char	8	This defined as the charcter alphonse.	N/A	N/A
menu rooms_from:	Boolea n	N/A	Menu for a new choices.	N/A	N/A
y	char	8	This char is also defined as mysterious character.	N/A	N/A
yr	char	8	This is defined as the charcter Yuri.	N/A	N/A

Samig Sherchan

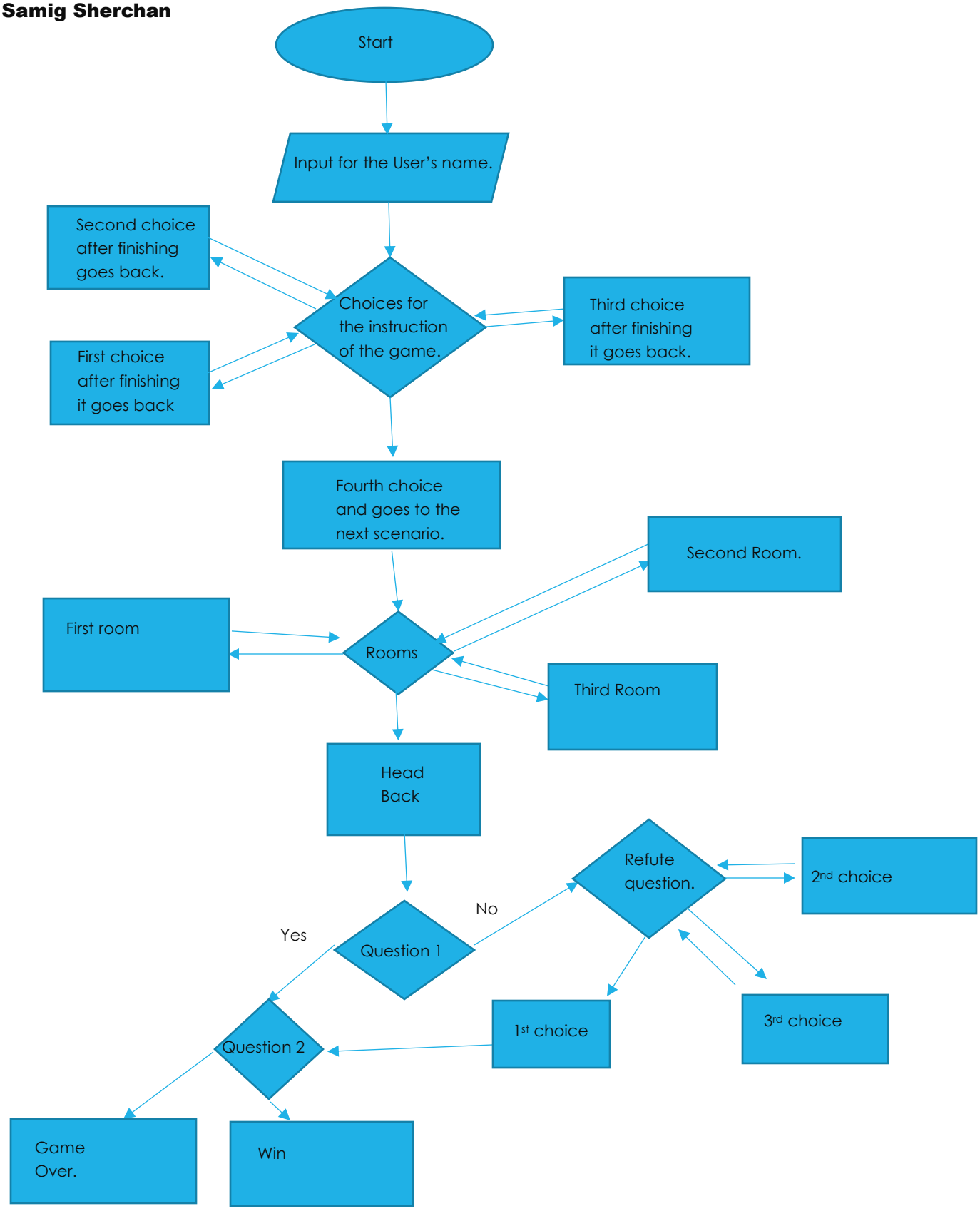
lving	Backgr ound	N/A	The scene of the living room.	N/A	N/A
sv	Backgr ound	N/A	Another scene of the rooms.	N/A	N/A
label again:	string	N/A	New lable for the start of questions.	N/A	N/A
pl	Backgr ound	N/A	This is the scene for the game room.	N/A	N/A
menu counter_qu estion:	Boolea n	N/A	The ingame menu for the refute question.	N/A	N/A
menu counter_die :	Boolea n	N/A	The second question which decides the game.	N/A	N/A
menu first_questio n:	Boolea n	N/A	This the first question choices asked.	N/A	N/A



Samig Sherchan

Test No	Description	Input Data	Expected Outcome.	Actual
1.	Is the window 1280x720?	Start Game	1280x720	
2.	Does the game scenes change and work?	Start Game	When running the game must change scene where its asked too.	
3.	Does the image of character appear alongside the dialogue?	Start game	The images must show alongside the dialogue.	
4.	Does the in-game menu allows the user to choose options?	Start game, and mouse click.	The user can choose the choices and it reads the statements.	
5.	Does the game allow iteration after finishing each choices?	Start game and mouse click?	Some of the in-game menu must iterate.	
6.	Does the buttons and keyboard plus mouse controll work?	Start game, keyboard and mouse click.	The buttons must work as well as the controlls.	

Samig Sherchan



Samig Sherchan

In the the flow chart above explain the stages and steps of the game from the start. Firstly, the game begins providing some dialogue from the characters alongside their image. Then a decision with four choices appears. The three choices all iterate so you can access those choices again and again because the first three choices explain the instruction of the game, while the last choice moves on to the next scenario. The next scenarios will include a long dialogue about the main character and where the character is. Furthermore the surrounding will be described as well. New character will slowly appear until the next decision comes. The next decision also has 3 iterating choices so they can access it anytime, one of the choices also helps find a new character eitherway the new character will appear after making decision. The last choice will head back to the first question, the question will involve the mathematical quesiton which is the main purpose of this solution. The question will be answered already however it's the user's choice on answering the question. It's a Boolean question as there is two choices Yes or No. Depending on the choice it will cause different routes however it will connect. So if the user selects Yes, then the user proceeds to the next question along with some dialogues. The next question decides the outcome of the whole game. The user gets to choose two options again. Based on that options the user will either win or lose the game. The users job is to validate a characters answer to the question. Going back to the question 1, if no is chose as the option then they will get the answer wrong and the user need to answer the refute question so they can advance to the question 2. One of the option is correct it does not involve maths but involves logical think and remembering while the other option iterates until the user chooses the right option. Furthermore, the game will include buttons to access different pages, buttons for options where the users can edit some preferences like the music volume and other audio noises. There will also be a button for for help which explains all the controlls for the game. For example, the fuction of the enter button will skip to the next dialogue. This makes it easier for user as they don't require to click which can cause iratating noise constantly. So pressing the keyboard is much easier and use less noise. Also it will have a load button if the user saves their progress then load it anytime they like so they don't require to start all over again. Load button will also have loading slots so the users can load to particallar scene they want depending on wherever they saved the game.

Samig Sherchan	Design requirements	
Requirement No	**Requirement**	**Justification**
1.	The screen size must be 1280x720.	The basic screen size of this program is around 1280x720 so it must remain constant in order for it to work.
2.	The characters must have influence on the way the user will think.	User will have different impressions on characters. The characters must also assist the users decisions.
3.	The background of the first scenario and other mysterious must be dark.	Since the first scenario involves the character being trapped and randomly a character asks quesiton. Furthermore, the main character will also be unconcious.
4.	Each background must related to each scenarios.	As having the same scene for every scenario will be really boring and lack of creatifvity.
5.	The title must be big and visible.	Its important that the user can see the title clearly and visibally.
6.	Help must contain all the information about the controls.	The user must have flexibilty and all the controls to play the games, if they don't know how to then it will be a major problem.
7.	The art of the game must be suitable for the focus group.	Making sure the images are appropriate for the age group so there isn't any explicity image or it will it will cause anger from the user.
8.	Atleast two question are asked.	There is no point of user playing a game for just one question, its important to actually enjoy and think with their brains.
9.	The user should have access to rechoose some ingame-menu.	Since some of the choices needs to iterate as the first choice will include instruction to the game. For other choices, the players will get chances so they don't easily lose.
10.	The dialogue of game must not be too long.	Users don't like to read a long dialogue.

This is the basic design requirement for the game, they must be included inorder to make the game successful. As the game might cause some error or problem for the users if not followed. The aim of this requirement is to make sure the users are having fun and using the solution the best way possible without having any problems.

Samig Sherchan

Here I will cover the major algorithm and psuedo codes that will be used for software development. Some are python equivalent codes while other Ren'py variables since the task demands for both codes to be used in other to create this soltion. These codes also helps create easier code and requires less lines making easier to go throught code and edit it anytime.

In the game there will be lots of images used, in order to use it in an easy manner, I can use the Ren'py term:

show This allows me to display the image by typing show then writing the name of the image file, the extension won't be required. This will make any image display when required. To adjust the image since the image is always in the centre I am going to use at right or at left this positions the images depending on where I want.

hide This allows me to hide the image whenever its required, so when a character is set to leave due to the scene I can just type hide and the name of the image fill which will help hide the image.

```
init python:

    n = Character("Name of the character")
```

This is intialised python statements to define the characters name. First you select a letter then equal it to the charcter and the write the name of the character in the speech mark. This will help introduce new charcters easily for the dialogue of the game.

scene This will help introduce new backgrounds and new scenes for the game. So to add the background you type scene the image file of the background. This will help change depending on the scenario to making it looks interesting and dynamic.

```
define pov = Character("[povname]")

python:

    povname = renpy.input("Please enter your name")

    povname = povname.strip()

if not povname:

        povname = "Name of the character"
```

This will in another python fuction parameter where it allows the user to input their name, which will be stored, and displayed when used. Furthmore it includes if statement so when you input it it will store the name and use it if not then it will give a default name which the programmer can only edit.

```
menu question:

    "Choice 1":

        $ question = "C1"

        pov "This is my first choice."

    "Choice 2":

        $ question = "C2"

        pov "This is choice 2"
```

This how you make ingame choices, first you type the menu then the name of the menu with a colon. In the speech mark are choices which will be displayed while under it is the choices being defined. You can create many choices as possible depending on the game. After defining it you can just write the character's letter you defined. Since pov in the user's character, the speech will be displayed depeding on the what the speech mark reads. You can add many dialogues and speech as possible.

```
if question == "C1":

    pov "This is the effect of the choice 1 huh…"

if question == "C2":

    pov "Choice 2, why did I choose it?"
```

Then you also require if statements for the each choices, you type if question then equal equal to speech mark the name of the choice you defined, then under it you can create dialogue or add image music or anything. You must do it for all the choices as well.

call Call function allows the user to go back to the choice or statement, the call statement iterates ingame menu or words. First you type call then the name of the menu and it calls back to the menu after finishing that menu.

Samig Sherchan

Teacher Comments:

- Broken the problem down systematically into a series of smaller problems which suited for the computational solutions. Explained the process used to decompose this problem.
- Defined in detials the structure of the solution which developed.
- Described the solution fully using appropriate and accurate algorithms and explained how these algorithms helped form a completed solution to the problem.
- Described, explained choices made, alongside the usability features that was included in the solution.

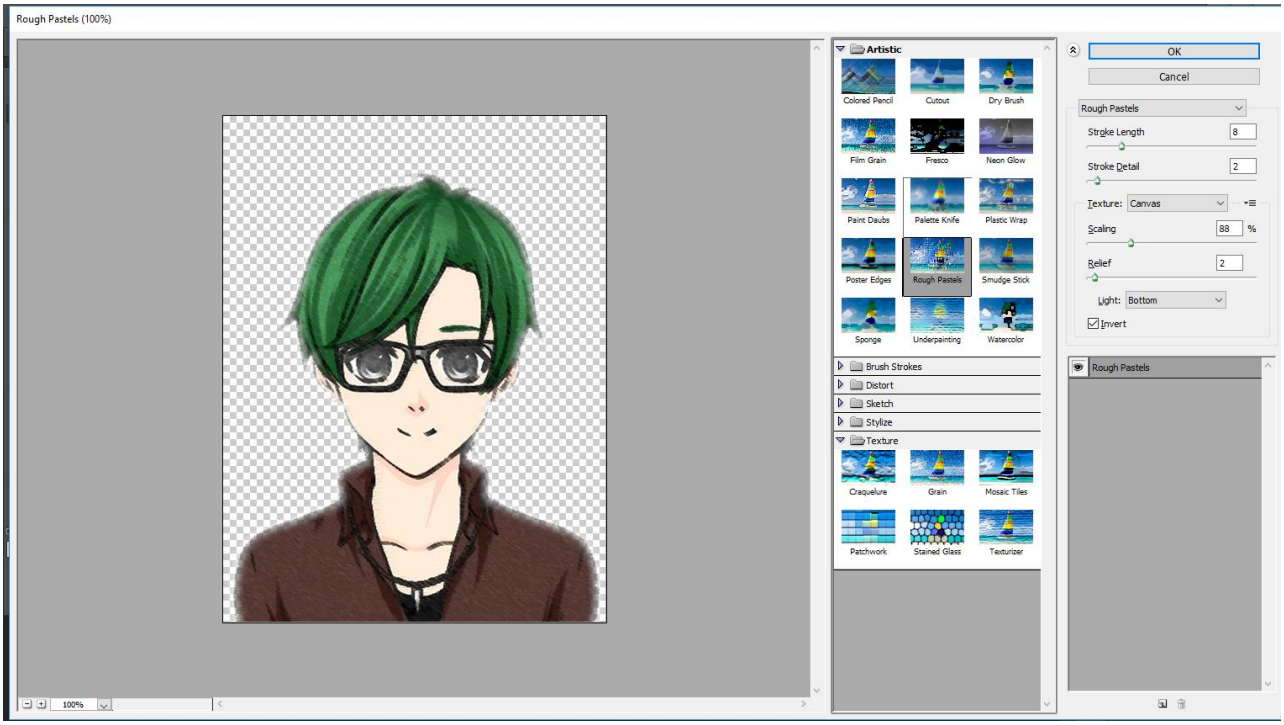
- Identified and justified the keyvariables / data structures / classes (as appropriate to the proposed solution)explained any necessary validation.

- Identified and justified the test data used on the iterative development of the solution.

- Identified and justified any further data used in the post development phase.

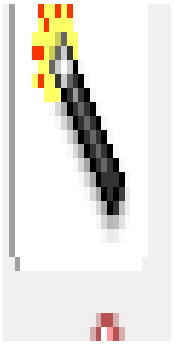
3.3 DEVELOPING THE SOLUTION (25 MARKS)

Next was to edit the image in Photoshop, to take out unnessary backgorund colour and the website link in the image. Furthermore, Photoshop allowed me to filter images, I wanted the characters to look more like a drawing, so I used the filter gallery using the feature rough pastels for all the characters. As you see on the right hand side, I have the ability to control on the



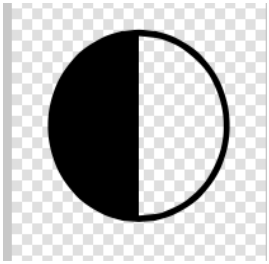
strenghth of the filtering, as I don't want to filter too much or my character won't be visible enough to see in the game. Which will cause the my target audience to dislike the game. The first impression usually depends on the images, I don't want my users to loose interest without even trying out the game. So after filtering I saved the file has JPEG, however I didn't realize that saving files in JPEG would make the background of the character all white. So whenever I insert this character in the game it'll cause problems as it will block out the game scene. So I had to use Fireworks, which is easy to use software specially used to make images transparent. This did cost some time, plus I hadn't saved my images in photoshop format so if I did use photoshop I would require to do everything again. So I had to use Fireworks where all I required to is used the magic want which was the easier than restarting everything all over again.

Samig Sherchan



To use Fireworks, all I required is to use the magic tool button, on the second image it's a icon of the magic want, then I needed to click on white background and click enter which easily erased the white background to make it transparent. Then I saved all the images in PNG extension so the transparency remains.

I repeatedly continue to do that with other images, and in the end it was a success although it did take some time. After character development I aslo required to add other features to my game, here is the the steady development of the villian made from scratch in fireworks, the art style may not be the best; it was quite long to build. Started from a semi-circle with a contrasting colours then step by step I added other shapes. At some point the face of the villian character changed.



First of all, I created a circle using the shape tool. I had to select a elipse tool from the small drop down arrow. I used black colours then tried to split the the circle into semi-circle. With knife and eraser tools. The knife tools allowed me to select the points and cut out the circle in half. So it has two contrasting colours black and white.Plus the background is transparent so the image doesn't block out the background.

Followed on by making the face, at first I made arrows for the mouth and ears. Alongside the elipse to make out the eyes. Also using triangles for the nose so it looks like a bear which is how I want it. All using the the shape tools with the drop down boxes.

Then I changed my whole concept of the bear design, I still used the the shape tools. I didn't replace the triangular ears, I managed to find the pointers that allows you to adjust the image like to increase the size of the shapes. There was also pointers that gave me the ability to change the expand certain pointers which allowed to create different shape like sharp-cut like edges for the villain.

Samig Sherchan

Options script:

```
 ## Main Menu
#############################################################

 define config.name = _("Mathsronpa")

## Determines if the title given above is shown on the main menu screen.

 define gui.show_name = True

## Setting it to True shows the title, if set the False the title hidden.

 define config.version = "1.0"

##Displays the value in speech marks (the version of the game). It can be changed anytime.

define gui.about = _("")

## Text that is placed on the game's about screen. To insert a blank line

define build.name = "Mathsronpa"

## A short name for the game used for executables and directories in the built.

 ## Sounds and music

 ############################################################
```

(My game doesn't include any sound, music or voice. However, its still important to have this code)

```
define config.has_sound = True

## Enables sound since its set on True, False disables sounds.

 define config.has_music = True

## Enables music since its set on True, False disables music.

 define config.has_voice = True

## Enables voice of the characters  since its set on True, False disables voice.
```

```
## Transitions
##############################################################################


 define config.enter_transition = dissolve

## When a new scene that I made enters it transits by disolving.

 define config.exit_transition = dissolve

## When a scene exits for a new scene it disolves as well.

 define config.after_load_transition = None

## A transition that is used after a game has been loaded. Since the value is 'None' no
transition occurs.

 define config.end_game_transition = None

## Used when entering the main menu after the game has ended. Since the value is 'None' no
transition occurs.

 ## Window management
##############################################################################

 define config.window = "auto"

 ## This controls when the dialogue window is displayed. "auto", the window is hidden before
scene statements and shown again once dialogue is displayed. You can type "hide" to the
opposite.

define config.window_show_transition = Dissolve(.2)

## Transitions used to shows dialogue window at the speed of 2 seconds.

 define config.window_hide_transition = Dissolve(.2)

## Transitions used to hide dialogue window at the speed of 2 seconds.
```

Samig Sherchan

Preference defaults
##

default preferences.text_cps = 0

Controls the default text speed. The default, 0, is infinite, while any other, number is the number of characters per second to type out.

default preferences.afm_time = 15

The default auto-forward delay. Larger text will take longer to display. It ranges from 0 to 30 seconds.

Save directory
##

define config.save_directory = "Mathsronpa-1474052795"

This generally should not be changed, and if it is, should always be a literal string, not an expression. The directory saves in the software's main folder with its own folder named after the game.

Icon

##

define config.window_icon = "gui/window_icon.png"

The icon displayed on the taskbar or dock. It can be changed anytime with anyother icons.

```
## Build configuration
##############################################################

## This section controls how Ren'Py turns my  project into distribution files.

init python:

    build.classify('**~', None)

## An example of a file in the folder that will not be included in the distribution.

    build.classify('**.bak', None)

## An example of a file in the folder that will not be included in the distribution.

    build.classify('**/.**', None)

## An example of a file in the folder that will not be included in the distribution.

    build.classify('**/#**', None)

## An example of a file in the folder that will not be included in the distribution.

    build.classify('**/thumbs.db', None)

## An example of a file in the folder that will not be included in the distribution.

      # build.classify('game/**.png', 'archive')

## PNG will be archived and be include in the distribution.

      # build.classify('game/**.jpg', 'archive')

## PNG will be archived and be include in the distribution

      build.documentation('*.html')

## The script can be accessed in html formate and distrubuted.

      build.documentation('*.txt')

## The script is written as text file.
```

Samig Sherchan

GUI Script:

```
## Initialization

################################################################################

init offset = -2

## The init offset statement causes the init code in this file to run before. It usually intialises Python statements.

init python:

    gui.init(1280, 720)

## width and height of the game. It is used by intialising the python statements.

## Colors
################################################################################

define gui.accent_color = '#0099cc'

## The colors of text in the interface. It will show the interface at this colour.

define gui.idle_color = '#555555'

## The color used for a text button when it is neither selected nor hovered.

define gui.idle_small_color = '#aaaaaa'

## The small color is used for small text, which needs to be brighter/darker to achieve the same effect.

define gui.hover_color = '#66c1e0'

## The color that is used for buttons and bars that are hovered.

define gui.selected_color = '#ffffff'

## The color used for a text button when it is selected but not focused. A button is selected if it is the current screen or preference value.

define gui.insensitive_color = '#5555557f'

## The color used for a text button when it cannot be selected.
```

```
define gui.muted_color = '#003d51'

define gui.hover_muted_color = '#005b7a'

## Colors used for the portions of bars that are not filled in. These are not used directly, but are used when re-generating bar image files.

define gui.text_color = '#ffffff'

## The colours used for the text.

define gui.interface_text_color = '#ffffff'

## The colors used for the interface.

## Fonts and Font Sizes
########################################################################

define gui.default_font = "DejaVuSans.ttf"

## The font used for in-game text.

define gui.name_font = "DejaVuSans.ttf"

## The font used for character names.

define gui.interface_font = "DejaVuSans.ttf"

## The font used for out-of-game text.

define gui.text_size = 22

## The size of normal dialogue text.

define gui.name_text_size = 30

## The size of character names.

define gui.interface_text_size = 24

## The size of text in the game's user interface.

define gui.label_text_size = 28

## The size of labels in the game's user interface.

define gui.notify_text_size = 16

## The size of text on the notify screen.
```

```
 define gui.title_text_size = 50

 ## The size of the game's title.

 ## Main and Game Menus
####################################################################

define gui.main_menu_background = "gui/main_menu.png"

##The image used for the main menu background

define gui.game_menu_background = "gui/game_menu.png"

## The image used for game menu background.

define gui.show_name = True

## Should we show the name and version of the game? True will show the name and version of the game, while False won't.

 ## Dialogue
####################################################################

 ## These variables control how dialogue is displayed on the screen one line at a time.

define gui.textbox_height = 185

 ## The height of the textbox containing dialogue.

define gui.textbox_yalign = 1.0

 ## The placement of the textbox vertically on the screen. 0.0 is the top, 0.5 is center, and 1.0 is the bottom.

 define gui.name_xpos = 240

 ## The placement of the speaking character's name, relative to the textbox.

 define gui.name_ypos = 0

 ## These can be a whole number of pixels from the left or top, or 0.5 to center.

define gui.name_xalign = 0.0

 ## The horizontal alignment of the character's name. This can be 0.0 for left- aligned, 0.5 for centered, and 1.0 for right-aligned.
```

```
 define gui.namebox_width = None

##The width of the border box cotaining the characters' name. None automatically sizes the width.

 define gui.namebox_height = None

##The height of the border box cotaining the characters' name. None automatically sizes the height.

define gui.namebox_borders = Borders(5, 5, 5, 5)

 ## The borders of the box containing the character's name, in left, top, right, bottom order.

define gui.namebox_tile = False

 ## If True, the background of the namebox will be tiled, if False, the background if the namebox will be scaled.

define gui.text_xpos = 268

##The text x-axis position.

define gui.text_ypos = 50

##The text y-axis position.

## The placement of dialogue relative to the textbox. These can be a whole number of pixels relative to the left or top side of the textbox, or 0.5 to center.

define gui.text_width = 744

## The maximum width of dialogue text, in pixels.

define gui.text_xalign = 0.0

## The horizontal alignment of the dialogue text. This can be 0.0 for left-aligned, 0.5 for centered, and 1.0 for right-aligned.
```

```
## Buttons
################################################################################

## These variables, along with the image files in gui/button, control aspects of how buttons are displayed.

define gui.button_width = None

##Since 'None', Ren'Py computes a size._height = 36 and uses a suitable width.

define gui.button

## The width and height of a button, in pixels.

define gui.button_borders = Borders(4, 4, 4, 4)

  ## The borders on each side of the button, in left, top, right, bottom order

define gui.button_tile = False

  ## If True, the background image will be tiled. If False, the background image will be linearly scaled.

define gui.button_text_font = gui.interface_font

  ## The font used by the button.

define gui.button_text_size = gui.interface_text_size

  ## The size of the text used by the button.

  define gui.button_text_idle_color = gui.idle_color

  ## The color of button text in idle state.

  define gui.button_text_hover_color = gui.hover_color

  ## The color of button text in hover state.

  define gui.button_text_selected_color = gui.selected_color

  ## The color of button text in selected state.

  define gui.button_text_insensitive_color = gui.insensitive_color

  ## The color of button text in insensitive state.

  define gui.button_text_xalign = 0.0

  ## The horizontal alignment of the button text. (0.0 is left, 0.5 is center, 1.0 is right).
```

```
## These customizations are used by the default interface:

define gui.radio_button_borders = Borders(25, 4, 4, 4)

##Radio borders button sizes.

define gui.check_button_borders = Borders(25, 4, 4, 4)

##Check button borders sizes.

define gui.confirm_button_text_xalign = 0.5

##Confirm button text xalign size.

define gui.page_button_borders = Borders(10, 4, 10, 4)

##Page button border sizes.

define gui.quick_button_borders = Borders(10, 4, 10, 0)

##Quick button border sizes.

define gui.quick_button_text_size = 14

##Quick button text sizes.

define gui.quick_button_text_idle_color = gui.idle_small_color

##Quick button text (idle state) colour.

define gui.quick_button_text_selected_color = gui.accent_color

##Quick button text (selected state) colour.

define gui.navigation_button_width = 250

 ##Width of a navigation button.
```

Samig Sherchan

```
## Choice Buttons
##############################################################

## Choice buttons are used in the in-game menus. They will appear in the game most of the
times. Where ever the IF statements are placed.

define gui.choice_button_width = 790

##Choice button width.

define gui.choice_button_height = None

##Choice button height is None so it resizes it self in a suitable size.

define gui.choice_button_tile = False

##Choice tile is False.

define gui.choice_button_borders = Borders(100, 5, 100, 5)

##Choice button borders' size.

define gui.choice_button_text_font = gui.default_font

##Choice button text font. Set in default.

define gui.choice_button_text_size = gui.text_size

##Choice button text size set in default size.

define gui.choice_button_text_xalign = 0.5

##Choice button text xalign size.

define gui.choice_button_text_idle_color = "#cccccc"

##Choice button text (idle state) color.

define gui.choice_button_text_hover_color = "#ffffff"

##Choice button text (hover state) color.
```

Samig Sherchan

```
## File Slot Buttons
##############################################################

## A file slot button is a special kind of button. It contains a thumbnail image, and text
describing the contents of the save slot. A save slot uses image files in gui/button, like the other
kinds of buttons.

 define gui.slot_button_width = 276

 ##Slot button width.

 define gui.slot_button_height = 206

 ##Slot button height.

 define gui.slot_button_borders = Borders(10, 10, 10, 10)

 ##Slot button borders' size.

 define gui.slot_button_text_size = 14

 ##Slot button text size.

 define gui.slot_button_text_xalign = 0.5

 ##Slot button text xalign size.

 define gui.slot_button_text_idle_color = gui.idle_small_color

 ## The slot button text (idle state) colour.

 define config.thumbnail_width = 256

 ##Thumnail width size of save slot.

 define config.thumbnail_height = 144

 ## The height of the thumnail size for save slot.

define gui.file_slot_cols = 3

##The number of columns in the grid of save slots.

define gui.file_slot_rows = 2

## The number of rows in the grid of save slots.
```

```
## Positioning and Spacing
##############################################################

 ## These variables control the positioning and spacing of various user interface elements.

define gui.navigation_xpos = 40

 ## The position of the left side of the navigation buttons, relative to the left side of the screen.

define gui.skip_ypos = 10

## The vertical position of the skip indicator.

define gui.notify_ypos = 45

## The vertical position of the notify screen.

define gui.choice_spacing = 22

## The spacing between menu choices.

define gui.navigation_spacing = 4

## Buttons in the navigation section of the main and game menus.

define gui.pref_spacing = 10

## Controls the amount of spacing between preferences.

define gui.pref_button_spacing = 0

## Controls the amount of spacing between preference buttons.

 define gui.page_spacing = 0

## The spacing between file page buttons.

 define gui.slot_spacing = 10

 ## The spacing between file slots.
```

```
## Frames
################################################################################

## These variables control the look of frames that can contain user interface components
when an overlay or window is not present.

define gui.frame_borders = Borders(4, 4, 4, 4)

## Generic frames that are introduced by player code.

define gui.confirm_frame_borders = Borders(40, 40, 40, 40)

## The frame that is used as part of the confirm screen.

define gui.skip_frame_borders = Borders(16, 5, 50, 5)

## The frame that is used as part of the skip screen.

define gui.notify_frame_borders = Borders(16, 5, 40, 5)

## The frame that is used as part of the notify screen.

define gui.frame_tile = False

## Should frame backgrounds be tiled?

## Bars, Scrollbars, and Sliders
################################################################################

## These control the look and size of bars, scrollbars, and sliders. The default GUI only uses sliders
and vertical scrollbars. All of the other bars are only used in creator-written code.

define gui.bar_size = 36

##The horizontal bar size.

define gui.scrollbar_size = 12

##The horizontal scrollbar size.

define gui.slider_size = 30

##The horizontal  slider size.

define gui.bar_tile = False

##The bar tile set false because its linearly scaled.
```

```
define gui.scrollbar_tile = False

##The scrollbar tile set false because its lineraly scaled.

define gui.slider_tile = False

## True if bar images should be tiled. False if they should be linearly scaled.

define gui.bar_borders = Borders(4, 4, 4, 4)

##Size of the horizontal bar borders.

define gui.scrollbar_borders = Borders(4, 4, 4, 4)

##Size of the horizontal scrollbar borders.

define gui.slider_borders = Borders(4, 4, 4, 4)

##Size of the horizontal slider borders.

define gui.vbar_borders = Borders(4, 4, 4, 4)

##Size of the vertical bar borders.

define gui.vscrollbar_borders = Borders(4, 4, 4, 4)

##Size of the vertical scrollbar borders.

define gui.vslider_borders = Borders(4, 4, 4, 4)

## Size of the vertical slider borders.

define gui.unscrollable = "hide"

## Unscrollable scrollbars in the gui. "hide" hides them, while None shows them.
```

Samig Sherchan

```
## History
########################################################################

## The history screen displays dialogue that the player has already dismissed.

define config.history_length = 250

##The length/number of blocks in the history screen dialogue.

define gui.history_height = 140

## The height of a history screen entry, or None to make the height variable at the cost of performance.

define gui.history_name_xpos = 150

##The history name x position.

define gui.history_name_ypos = 0

##The history name y position

define gui.history_name_width = 150

##The history name width.

define gui.history_name_xalign = 1.0

## The position, width, and alignment of the label giving the name of the speaking character.

define gui.history_text_xpos = 170

##The history text x position.

define gui.history_text_ypos = 5

##The history text y position.

define gui.history_text_width = 740

##The history text width.

define gui.history_text_xalign = 0.0

## The hisory text xalign size.
```

NVL-Mode

The NVL-mode screen displays the dialogue spoken by NVL-mode characters.

define gui.nvl_borders = Borders(0, 10, 0, 20)

The borders of the background of the NVL-mode background window.

define gui.nvl_height = 115

The height of an NVL-mode entry. Set this to None to have the entries dynamically adjust height.

define gui.nvl_spacing = 10

The spacing between NVL-mode entries when gui.nvl_height is None, and between NVL-mode entries and an NVL-mode menu.

define gui.nvl_name_xpos = 430

##The nvl mode entries x postion.

define gui.nvl_name_ypos = 0

##The nvl mode entries x postion.

define gui.nvl_name_width = 150

##The nvl mode name x width.

define gui.nvl_name_xalign = 1.0

##The nvl name x align size.

define gui.nvl_text_xpos = 450

##The nvl text x position.

define gui.nvl_text_ypos = 8

##The nvl text y position.

define gui.nvl_text_width = 590

##The nvl text width.

define gui.nvl_text_xalign = 0.0

The position, width, and alignment of the dialogue text.

```
define gui.nvl_thought_xpos = 240

##The nvl thought x position.

define gui.nvl_thought_ypos = 0

##The nvl thought y position.

define gui.nvl_thought_width = 780

##The nvl thought width.

define gui.nvl_thought_xalign = 0.0

##The nvl thought x xalign size.

define gui.nvl_button_xpos = 450

##The nvl button x positon.

define gui.nvl_button_xalign = 0.0

## The position of nvl button xalign size set to 0.

## Mobile devices
```

```
init python:

if renpy.variant("touch"):

## To make them easier to touch on tablets and phones.

gui.quick_button_borders = Borders(60, 14, 60, 0)

## This increases the size of the quick buttons and its borders.

if renpy.variant("small"):

 ## This changes the size and spacing of various GUI elements to ensure they are easily visible on phones.

gui.text_size = 30

##Text size is 30.

gui.name_text_size = 36

##Name text size is 36.
```

gui.notify_text_size = 25

##Notification text size is 25.

gui.interface_text_size = 36

##Interface text size is 36.

gui.button_text_size = 34

##Button text size is 34.

gui.label_text_size = 36

Label text size is 36.

gui.textbox_height = 240

##Text box height is 240.

gui.name_xpos = 80

##Name x position.

gui.text_xpos = 90

##Text x postion.

gui.text_width = 1100

Adjust the location of the textbox.

gui.choice_button_width = 1240

Change the size and spacing of items in the game menu.

gui.navigation_spacing = 20

##The spacing of navigation.

gui.pref_button_spacing = 10

##The preference button spacing.

gui.history_height = 190

##The history height.

gui.history_text_width = 690

##The history width.

```
gui.file_slot_cols = 2

##Slot grid columns.

gui.file_slot_rows = 2

## File button layout.

gui.nvl_height = 170

##Nvl name height.

gui.nvl_name_width = 305

##Nvl name width.

gui.nvl_name_xpos = 325

##Nvl name x position.

gui.nvl_text_width = 915

##Nvl text width.

gui.nvl_text_xpos = 345

##Nvl text x position.

gui.nvl_text_ypos = 5

##Nvl text y position.

gui.nvl_thought_width = 1240

##Nvl thought width.

gui.nvl_thought_xpos = 20

##Nvl thought x positon.

gui.nvl_button_width = 1240

##Nvl button width.

gui.nvl_button_xpos = 20

## NVL-mode.

gui.quick_button_text_size = 20

## Quick buttons text size.
```

Samig Sherchan

Screens Script:

```
## Initialization
################################################################################
init offset = -1
##Intialising the offset which is equivalent to xposition and yposition.
## Styles
################################################################################
style default:
    font gui.default_font
##Font is set on default to what Ren'py usually uses.
    size gui.text_size
##Default text size.
    color gui.text_color
##Default text color which is usually black.
style input:
    color gui.accent_color
##The accent color is used when inputing something like name.
style hyperlink_text:
    color gui.accent_color
##The accent color is used when hyperlinking a text.
    hover_color gui.hover_color
##This color focuses on the items of GUI. Like text.
    hover_underline True
##Since its True it will apply in the game.
```

```
 style gui_text:
     font gui.interface_font
##Defalt font for the text in the interface.
     color gui.interface_text_color
##Default text color in the interface.
     size gui.interface_text_size
##Default text size in the interface.
 style button:
     properties gui.button_properties("button")
##Default properties are applied to the button.
style button_text is gui_text:
     properties gui.button_text_properties("button")
##Default properties is also applied to button's text
     yalign 0.5
##The yalign size is 0.5.
style label_text is gui_text:
     color gui.accent_color
##Accent color in text label.
     size gui.label_text_size
##Default size of label text.
style prompt_text is gui_text:
     color gui.text_color
##Prompt text color set in default and defined.
     size gui.interface_text_size
##The prompt text size set in default and defined.
```

```
style bar:
    ysize gui.bar_size
##y-axis size for the gui bar in default.
    left_bar Frame("gui/bar/left.png", gui.bar_borders, tile=gui.bar_tile)
##Left frame bar is set as that specific png file that is being displayed.
    right_bar Frame("gui/bar/right.png", gui.bar_borders, tile=gui.bar_tile)
##Right frame bar is set as that specific png file that is being displayed.
style vbar:
    xsize gui.bar_size
##x-axis size of the gui vertical bar in default.
    top_bar Frame("gui/bar/top.png", gui.vbar_borders, tile=gui.bar_tile)
##The top bar frame is set as the png file that is being displayed.
    bottom_bar Frame("gui/bar/bottom.png", gui.vbar_borders, tile=gui.bar_tile)
 ##The bottom bar frame is set as the png file  that is being displayed.
 style scrollbar:
    ysize gui.scrollbar_size
 ##The y-axis of the scrollbar size in default.
    base_bar Frame("gui/scrollbar/horizontal_[prefix_]bar.png", gui.scrollbar_borders,
tile=gui.scrollbar_tile)
##The base of the bar frame has png file that is being displayed.
    thumb Frame("gui/scrollbar/horizontal_[prefix_]thumb.png", gui.scrollbar_borders,
tile=gui.scrollbar_tile)
##The thumb of the frame has png file that that is being displayed.
```

```
style vscrollbar:

    xsize gui.scrollbar_size

 ##The x-axis size of the vertical scroll bar size in default.

    base_bar Frame("gui/scrollbar/vertical_[prefix_]bar.png", gui.vscrollbar_borders,
tile=gui.scrollbar_tile)

##The vertical scrollbar base frame has png file that is being displayed.

    thumb Frame("gui/scrollbar/vertical_[prefix_]thumb.png", gui.vscrollbar_borders,
tile=gui.scrollbar_tile)

##The thumb frame of the vertical scrollbar has a png file that is being displayed.

style slider:

    ysize gui.slider_size

##y-axis size of the slider in default.

    base_bar Frame("gui/slider/horizontal_[prefix_]bar.png", gui.slider_borders, tile=gui.slider_tile)

##The base bar frame of the slider in png file that is being displayed.

    thumb "gui/slider/horizontal_[prefix_]thumb.png"

##The thumb of the slider has a png file that is being displayed.

style vslider:

    xsize gui.slider_size

##x-axis size of vertical slider size set in default.

    base_bar Frame("gui/slider/vertical_[prefix_]bar.png", gui.vslider_borders, tile=gui.slider_tile)

##Base bar frame of the vertical slider has png file that is being displayed.

    thumb "gui/slider/vertical_[prefix_]thumb.png"

##Thumb of the vertical slider is set as a png file which is being displayed.
```

```
style frame:
    padding gui.frame_borders.padding
 ##Spacings for the frame borders.
    background Frame("gui/frame.png", gui.frame_borders, tile=gui.frame_tile)
 ##The background frame is set as a png file which is being displayed.
## In-game screens
########################################################################
## Say screen
########################################################################
screen say(who, what):
    style_prefix "say"
 ## The say screen is called by the say statement, when displaying ADV-mode dialogue.
 window:
        id "window"
 ## A window or frame. This conventionally contains the who and what text. The character
object can be given arguments that style this displayable.
        text what id "what"
 ##The dialogue being said by the speaking character.
if who is not None:
##If the say screen is none and the text name of the character is none then.
window:
            style "namebox"
 ##Lets the users input the name. Or defines it with a default.
            text who id "who"
 ## A text displayable, displaying the name of the speaking character. The character object can be
given arguments that style this displayable.
```

```
    if not renpy.variant("small"):

        add SideImage() xalign 0.0 yalign 1.0

  # If there's a side image, display it above the text. Ay xalign of 0.0 and yalign of 1.0.

style window is default

##Style window is default if this occurs.

style say_label is default

##Style say_label is default if this occurs.

style say_dialogue is default

##Style say_dialogue is default if this occurs.

style say_thought is say_dialogue

##Style say_thoguht is same as say_dialogue.

style namebox is default

##Style namebox is set default if this occurs.

style namebox_label is say_label

##Style namebox_label is same as say_label.

style window:

    xalign 0.5

  ##The xalign is sized to 0.5.

    xfill True

  ##If true, the displayable will expand to fill all available horizontal space.

  yalign gui.textbox_yalign

  ##The yalign size of the text box is set in default.

    ysize gui.textbox_height

  ##The y-axis of text box height.

background Image("gui/textbox.png", xalign=0.5, yalign=1.0)

##The backgroudn image is set as this png file in size of the values in the xalign and yalign.
```

```
style namebox:

    xpos gui.name_xpos

##x-position of the style namebox in default.

    xanchor gui.name_xalign

##x-anchor of the style namebox in default.

    xsize gui.namebox_width

##x-size width of the style namebox in default.

    ypos gui.name_ypos

##y-position of the style namebox in default.

ysize gui.namebox_height

##y-size height of the style namebox in default.

background Frame("gui/namebox.png", gui.namebox_borders, tile=gui.namebox_tile,
xalign=gui.name_xalign)

##Background frame of the name box is set in this image file.

    padding gui.namebox_borders.padding

 ##Spacings for the namebox borders in default..

 style say_label:

    color gui.accent_color

##The accent color of the say_label in default.

    font gui.name_font

##The name font of the say_label in default.

 size gui.name_text_size

##The size of the name text size for the say_label in default.

    xalign gui.name_xalign

##xalign of the say_label in default.
```

```
    yalign 0.5

 ##The y-align of the say_label set in 0.5.

style say_dialogue:

    xpos gui.text_xpos

##x-position text of the say dialogue in default.

    xanchor gui.text_xalign

 ##x-anchor text size (xalign) of the say dialogue in default.

    xsize gui.text_width

 ##x-size width of the say_dialogue.

    ypos gui.text_ypos

 ##y-position text of the say_dialogue in default.

 text_align gui.text_xalign

 ##The text align of the say_dialogue in defualt.

    layout ("subtitle" if gui.text_xalign else "tex")

 ##The layout of the say_dialogue.

## Input screen
################################################################################

screen input(prompt):

    style_prefix "input"

 ##Input functon for the prefix.

window:

vbox:

        xpos gui.text_xpos

 ##The x-position of the vertical box in the window at default.

        xanchor gui.text_xalign

 ##The x-anchor of the vertical box in the window.
```

```
        ypos gui.text_ypos
  ##The y-position of the vertical box in the window at default.
text prompt style "input_prompt"
##The text that gives a input message in the window.
        input id "input"
  ##The input is required for the ID or the name of the character.
style input_prompt is default
##The input_promt is set default.
style input_prompt:
    xmaximum gui.text_width
  ##The x-maximum for the text width.
    xalign gui.text_xalign
  ##xalign of the input_prompt.
    text_align gui.text_xalign
  ##The text align for the input prompt set in default.
style input:
    xmaximum gui.text_width
  ##x-maximum text with for the input.
    xalign gui.text_xalign
##x-align text for the input in default.
    text_align gui.text_xalign
##Text align of the input.
```

Samig Sherchan

```
## Choice screen
################################################################################

screen choice(items):

    style_prefix "choice"

## This screen is used to display the in-game choices presented by the menu statement. The one parameter, items, is a list of objects, each with caption and action fields.

vbox:

        for i in items:

            textbutton i.caption action i.action

## An action that should be invoked when the menu choice is chosen.

define config.narrator_menu = True

##The menu captions will sopoken by the narrator since its true.

style choice_vbox is vbox

##The style choice_vbox will equal to vbox.

style choice_button is button

##Style choice_button is then same as button.

style choice_button_text is button_text

##The Style choice_button is the same as button_text.

style choice_vbox:

    xalign 0.5

##xalign size for the choice_vertical box.

    ypos 270

##The y-position for the choice_vertical box.

    yanchor 0.5

##The y-anchor for the choice_vertical box.
```

```
    spacing gui.choice_spacing

 ##The spacing for the choice_spacing.

style choice_button is default:

    properties gui.button_properties("choice_button")

##The default properties for the choice_button.

style choice_button_text is default:

    properties gui.button_text_properties("choice_button")

##The default properties for the choice_button text.

## Quick Menu screen
################################################################

 screen quick_menu():

 zorder 100

 ## Ensure this appears on top of other screens.

hbox:

        style_prefix "quick"

## Add an in-game quick menu for the hortizontal box.

xalign 0.5

##The xalign size for the horizontal box.

yalign 1.0

##The yalign size for the horizontal box.

textbutton _("Back") action Rollback()

##Alows the users to go back at the scene.

textbutton _("History") action ShowMenu('history')

##Allows the users to see the history of the game so far.

textbutton _("Skip") action Skip() alternate Skip(fast=True, confirm=True)

##The skip button skips once at a time if you click it or you can also do fast skip which is automatic.
```

```
textbutton _("Auto") action Preference("auto-forward", "toggle")

##The auto button plays the scene at specific time.

textbutton _("Save") action ShowMenu('save')

##The save button saves the current progress.

textbutton _("Q.Save") action QuickSave()

##The button quick saves the specific scene.

        textbutton _("Q.Load") action QuickLoad()

  ##The button quick loads the specific scene.

        textbutton _("Prefs") action ShowMenu('preferences')

  ##The user can edit their preferences here.

init python:

    config.overlay_screens.append("quick_menu")

  ## This code ensures that the quick_menu screen is displayed in-game, whenever the player
has not explicitly hidden the interface.

style quick_button is default

##The quick button is set in default.

style quick_button_text is button_text

##The quick button is set same as button_text.

style quick_button:

    properties gui.button_properties("quick_button")

  ##The properties of the quick button set in default.

style quick_button_text:

    properties gui.button_text_properties("quick_button")

  ##The properties of the quick button text set in default.
```

Samig Sherchan

```
## Main and Game Menu Screens

## Navigation screen
##############################################################

screen navigation():

 vbox:

        style_prefix "navigation"

  ## This screen is included in the main and game menus, and provides navigation to other
menus, and to start the game.

 xpos gui.navigation_xpos

##x-positions for the screen navigation in default.

        yalign 0.5

##The y-align size of the screen navigation.

spacing gui.navigation_spacing

##The spacing for the screen navigation.

  if main_menu:

 textbutton _("Start") action Start()

##If main menu then you get a Start button.

else:

  textbutton _("History") action ShowMenu("history")

  ##Else it will show the History button with other buttons.

  textbutton _("Save") action ShowMenu("save")

  ##Else it will show the Save button with other buttons.

  textbutton _("Load") action ShowMenu("load")

  ##Else it will show the Load button with other buttons.

  textbutton _("Preferences") action ShowMenu("preferences")

  ##Else it will show the Preferences button with other buttons.
```

```
 if _in_replay:
            textbutton _("End Replay") action EndReplay(confirm=True)
  ##End Replay button show if Replay is used.
        elif not main_menu:
            textbutton _("Main Menu") action MainMenu()
  ##Main Menu button show if not using Replays.
 textbutton _("About") action ShowMenu("about")
##About button show if not using Replays.
if renpy.variant("pc"):
  textbutton _("Help") action ShowMenu("help")
##Help button for those who don't understand the basics of this game.
textbutton _("Quit") action Quit(confirm=not main_menu)
##The Quit button to quit the game.
  style navigation_button is gui_button
  ##The style of the navigation_button is the same as gui_button.
  style navigation_button_text is gui_button_text
  ##The style of the navigation_button text is the same as gui_button text.
style navigation_button:
    size_group "navigation"
  ##The size_group of the navigation_button.
    properties gui.button_properties("navigation_button")
  ##The default properties of the navigation_button.
style navigation_button_text:
    properties gui.button_text_properties("navigation_button")
  ##The default properties of the navigation button text.
```

Samig Sherchan

```
 ## Main Menu screen
##############################################################

screen main_menu():

tag menu

## This ensures that any other menu screen is replaced.

 style_prefix "main_menu"

##The prefered name for the button or fuction.

add gui.main_menu_background

##Adds a background for the main_menu.

frame:

        pass

  ## This empty frame darkens the main menu.

use navigation

## The use statement includes another screen inside this one. The actual contents of the main menu are in the navigation screen.

if gui.show_name:

vbox:

            text "[config.name!t]":

  ##The virtual vbox shows the config name.

style "main_menu_title"

##The main_menu_title in the vertical box.

text "[config.version]":

                style "main_menu_version"

  ##The text for the config.version.
```

```
  style main_menu_frame is empty
  ##The style main-menu_frame is empty for the vertical box.
  style main_menu_vbox is vbox
  ##The style main_menu_vbox is vbox.
  style main_menu_text is gui_text
  ##Style main_menu_text is the gui_text.
  style main_menu_title is main_menu_text
  ##The style main_menu title is the main_menu_text.
  style main_menu_version is main_menu_text
  ##The style main_menu+version is main_menu_text.
style main_menu_frame:
    xsize 280
  ##The style main_menue_frame x-size.
    yfill True
  ##Y-fill is true so it fills the background.
  background "gui/overlay/main_menu.png"
  ##Uses this image file to fill the background.
style main_menu_vbox:
    xalign 1.0
  ##The xalign of the style main_menu of the vertical box.
    xoffset -20
  ##The x-offset of the style main_menu of the vertical box.
    xmaximum 800
  ##The x-maximum size of the style main_menu of the vertical box.
yalign 1.0
   ##The yalign of the style main_menu of the vertical box.
```

```
    yoffset -20
  ##The yoffset of the style main_menu of the vertical box.
style main_menu_text:
      xalign 1.0
##The xalign size of the main_menu_text.
layout "subtitle"
##The layout for the subtitle.
      text_align 1.0
  ##The text align for the main_menu_text.
      color gui.accent_color
  ##The accent color for the main_menu text in default.
style main_menu_title:
      size gui.title_text_size
  ##The default main_menu_title size.
## Game Menu screen
################################################################
screen game_menu(title, scroll=None):
## The scroll parameter can be None, or one of "viewport" or "vpgrid". When this screen is
intended to be used with one or more children, which are transcluded (placed) inside it.
    if main_menu:
        add gui.main_menu_background
      ##Add the main menu background.
      else:
        add gui.game_menu_background
  ##Or add the game menu background.
```

```
    style_prefix "game_menu"
 ##Style prefix for the game menu.
 frame:
       style "game_menu_outer_frame"
 ##The outer frame of the game menu.
hbox:
         frame:
             style "game_menu_navigation_frame"
## Reserve space for the navigation section.
frame:
             style "game_menu_content_frame"
## Reserve space for the content section.
if scroll == "viewport":
viewport:
                 scrollbars "vertical"
 ##Defines scrollbars for vertical use.
                 mousewheel True
 ##The mousewheel is set True, it will scroll when you move use the mousewheel.
                 draggable True
 ##Makes the scrollball dragable to move the page.
                 side_yfill True
 ##Fills the side background since its set true.
```

```
 vbox:
                    transclude
  ## Hypertext reference for vbox.
elif scroll == "vpgrid":
vpgrid:
                    cols 1
  ##VP grid columns.
                    yinitial 1.0
  ##VP grid y-intial.
                    scrollbars "vertical"
  ##Scrollbars defined vertical.
                    mousewheel True
  ##Mousewheel is accessible since its true.
                    draggable True
  ##The mouse is draggable since its true.
side_yfill True
##The side y-fill is true so the background is set.
Transclude
##The hypertext refference for VP Grid.
else:
  transclude
  ##Alternative hypertext reference.
use navigation
##Uses navigation.
```

```
 textbutton _("Return"):
        style "return_button"
  ##Shows the return button.
action Return()
##The return button's actions.
label title
##Labels the title.
 if main_menu:
        key "game_menu" action ShowMenu("main_menu")
  ##Game menu key shows the main menu.
style game_menu_outer_frame is empty
##Game menu outer frame is set empty.
  style game_menu_navigation_frame is empty
  ##Game menu navigation frame is set empty.
  style game_menu_content_frame is empty
  ##Game menu content frame is set empty.
  style game_menu_viewport is gui_viewport
  ##Game menu viewport is set as gui viewport.
  style game_menu_side is gui_side
  ##Game menu side is set as gui-side.
  style game_menu_scrollbar is gui_vscrollbar
  ##Game menu scrollbar is set as vertical scrollbar.
style game_menu_label is gui_label
##Game menu label is set as gui label.
  style game_menu_label_text is gui_label_text
  ##Game menu label text is set as gui label text.
```

```
 style return_button is navigation_button
 ##Style return_button is set as navigation_button.
 style return_button_text is navigation_button_text
 ##Style return_button text is set as navigation button text.
style game_menu_outer_frame:
    bottom_padding 30
 ##The game_menu_outer_frame bottom padding distance.
    top_padding 120
 ##The game_menu_outer_frame top padding distance.
background "gui/overlay/game_menu.png"
##The image file used for the background of the game_menu_outer_frame.
style game_menu_navigation_frame:
    xsize 280
 ##The style game_menu_navigation_frame x-size is set to 280.
    yfill True
 ##The y-fill for the background of the navigation frame is set true it will automatically fill it.
style game_menu_content_frame:
    left_margin 40
 ##The game_menu_content_frame left margin size.
    right_margin 20
 ##The game_menu_content_frame right margin size.
    top_margin 10
 ##The game_menu_content_frame top_margin size.
```

```
style game_menu_viewport:
    xsize 920
  ##The style game_menu_viewport x-size set as 920.
style game_menu_vscrollbar:
    unscrollable gui.unscrollable
  ##Style game_menu_vertical scrollbar is set as unscrollable.
style game_menu_side:
    spacing 10
  ##Style game_menue_side spacing is set 10.
style game_menu_label:
    xpos 50
  ##The style game_menu_label x-position is set 50.
    ysize 120
  ##The style game_menu_label y-size is set 120.
style game_menu_label_text:
    size gui.title_text_size
  ##The style game_menu_label_text size in default.
    color gui.accent_color
  ##The style game_menu_label_text accent color in default.
    yalign 0.5
  ##The yalign of style game_menu_label_text set as 0.5.
style return_button:
    xpos gui.navigation_xpos
  ##The x position style return_button.
    yalign 1.0
  ##The y-align of style return_button.
    yoffset -30
  ##The y-offset of style return_button.
```

Samig Sherchan

```
## About screen
##############################################################################

## This screen gives credit and copyright information about the game and Ren'Py.

screen about():

 tag menu

## There's nothing special about this screen, and hence it also serves as an example of how to
make a custom screen.

use game_menu(_("About"), scroll="viewport"):

##This tells the infromation about the game and other related elements.

  style_prefix "about"

  ##The prefix for the about section.

vbox:

  label "[config.name!t]"

  ##The label of the config for the vertical box.

            text _("Version [config.version!t]\n")

  ##The text of the config version for the vertical box.

if gui.about:

                text "[gui.about!t]\n"

  ## gui.about is usually set in options.rpy.

  text _("Made with {a=https://www.renpy.org/}Ren'Py{/a}
[renpy.version_only].\n\n[renpy.license!t]")

  ##The license for renpy.

define gui.about = ""

## This is redefined in options.rpy to add text to the about screen.

style about_label is gui_label

##The style about_label is set as gui_label.
```

```
 style about_label_text is gui_label_text

 ##The style about_label_text is set as gui_label_text.

style about_text is gui_text

##The style about_text is set as gui_text.

style about_label_text:

    size gui.label_text_size

 ##The size of style about_label_text is set defaut.

## Load and Save screens
#####################################################

 ## These screens are responsible for letting the player save the game and load it again. Since
they share nearly everything in common, both are implemented in terms of a third screen,
file_slots.

screen save():

tag menu

##The tag menu of the screen save.

use file_slots(_("Save"))

##File slots to save the progress and for loading.

screen load():

tag menu

##Tag menu of the screen load.

use file_slots(_("Load"))

##The file slots that can be used to load the progress after saving it.

screen file_slots(title):

 default page_name_value = FilePageNameInputValue()

 ##The name of the file slot that can inputed.

use game_menu(title):

##The game_menu title in the load slot.
```

```
        fixed:
    order_reverse True
 ## This ensures the input will get the enter event before any of the buttons do.
            button:
                style "page_label"
## The page name, which can be edited by clicking on a button.
 key_events True
 ##If the button is pressed and its true then.
                xalign 0.5
 ##The xalign of the button is 0.5.
                action page_name_value.Toggle()
 ##The buttons value when using it.
 input:
style "page_label_text"
##Input style for the page label.
 value page_name_value
 ##Input value page name.
 grid gui.file_slot_cols gui.file_slot_rows:
                style_prefix "slot"
 ## The grid of file slots.
xalign 0.5
##The xalign of the grid of file slot.
 yalign 0.5
 ##The yalign of the grid of file slot.
```

Samig Sherchan

```
    spacing gui.slot_spacing

##Spacing of slot in the grid.

for i in range(gui.file_slot_cols * gui.file_slot_rows):

##FOR loop for save slots.

$ slot = i + 1

##Range increases as more slots are made.

button:

action FileAction(slot)

##The file action slot button.

has vbox

##Has vbox included.

add FileScreenshot(slot) xalign 0.5

##File screenshot slot and the xalign size for it.

text FileTime(slot, format=_("{#file_time}%A, %B %d %Y, %H:%M"), empty=_("empty slot")):

##Format for the filetime slots.

style "slot_time_text"

##The style for the slot time text.

text FileSaveName(slot):

##Text for the file save name slot.

style "slot_name_text"

##The style slot name text.

key "save_delete" action FileDelete(slot)

##The key to save or delete slot actions.
```

```
hbox:

style_prefix "page"

## Buttons to access other pages.

xalign 0.5

##The xalign of horizontal box.

yalign 1.0

##The yalign of the horizontal box.

spacing gui.page_spacing

##The page spacing for the horizontal box.

textbutton _("<") action FilePagePrevious()

##Goes back to the previous text.

textbutton _("{#auto_page}A") action FilePage("auto")

##Does every transitions between page automatically.

textbutton _("{#quick_page}Q") action FilePage("quick")

##Makes the page change quick.

for page in range(1, 10):

        textbutton "[page]" action FilePage(page)

##The range(1, 10) gives the numbers from 1 to 9. So that amount of aciton are in a page.

textbutton _(">") action FilePageNext()

##Transist to the next page.

style page_label is gui_label

##The style page_label is set as gui_label.

style page_label_text is gui_label_text

##The style page_label_text is set as gui_label_text

style page_button is gui_button

##The style page_button is set as gui_button.
```

```
style page_button_text is gui_button_text
##The style page_button_text is set as gui_button_text.
style slot_button is gui_button
##The style slot_button is set asgui_button.
style slot_button_text is gui_button_text
##The style slot_button_text is set as gui_button_text.
style slot_time_text is slot_button_text
##The style slot_time_text is set as slot_button_text.
style slot_name_text is slot_button_text
##The style slot_name_text is set as slot_button_text.
style page_label:
    xpadding 50
##The xpadding for the style page_label.
    ypadding 3
##The ypadding for the style page_label.
style page_label_text:
    text_align 0.5
##The text align size for style page_label_text.
    layout "subtitle"
##The text layout subtitle for style page_label_text.
    hover_color gui.hover_color
##The hover color for style page_label_text.
style page_button:
    properties gui.button_properties("page_button")
##The default properties for style page_button.
```

```
style page_button_text:
    properties gui.button_text_properties("page_button")
 ##The default properties for style page_button_text.
style slot_button:
    properties gui.button_properties("slot_button")
 ##The default properties for style slot_button.
 style slot_button_text:
    properties gui.button_text_properties("slot_button")
 ##The default properties for style slot_button_text.


 ## Preferences screen
########################################################################
 ## The preferences screen allows the player to configure the game to better suit themselves.
screen preferences():
    tag menu
 ##The tag menu for the screen preferences.
if renpy.mobile:
        $ cols = 2
 ##If renpy is in mobile then column is 2.
    else:
        $ cols = 4
 ##Else the column is 4 instead of 2.
```

```
use game_menu(_("Preferences"), scroll="viewport"):

 vbox:

 ##Verical box for the game menu (Preferences).

 hbox:

 box_wrap True

 ##The box wrap is set as true for the horizontal box.

 if renpy.variant("pc"):

 ##IF the Renpy variant is the pc then.

 vbox:

 style_prefix "radio"

 ##The style prefix of the radio.

 label _("Display")

 ##The label for the display setting.

 textbutton _("Window") action Preference("display", "window")

 ##The text button that allows the users to use the game window in windowed state.

  textbutton _("Fullscreen") action Preference("display", "fullscreen")

 ##The text button allows the user to use the fullscreen mode.

 vbox:

 style_prefix "radio"

 ##The style prefix for the radio.

 label _("Rollback Side")

 ##The rollback side of the the game.

 textbutton _("Disable") action Preference("rollback side", "disable")

 ##Allows rollback to be disabled.

 textbutton _("Left") action Preference("rollback side", "left")

 ##Rollback action for left key.
```

```
textbutton _("Right") action Preference("rollback side", "right")

##The rollback action for right key.

vbox:

style_prefix "check"

##The style prefix check.

label _("Skip")

##The label for the skip.

textbutton _("Unseen Text") action Preference("skip", "toggle")

##Preferencence for the unseen text when skip.

textbutton _("After Choices") action Preference("after choices", "toggle")

##This controls the skip, only skips when choices are made if there are choices in the game.

textbutton _("Transitions") action InvertSelected(Preference("transitions", "toggle"))

##The transtion action preference can adjust the transitions.

null height (4 * gui.pref_spacing)

##The hieght for the spacing.

hbox:

style_prefix "slider"

##The style prefix for slider.

box_wrap True

##Box wrap set true of the horizontal box.

vbox:

label _("Text Speed")

##The label for the text speed.

bar value Preference("text speed")

##The value set for the text speed.
```

```
label _("Auto-Forward Time")

##Auto forwards to other scenes.

bar value Preference("auto-forward time")

##The speed or rate of auto forwarding.

vbox:

if config.has_music:

##If config includes music then.

label _("Music Volume")

##Allows the user to configure the volume.

hbox:

bar value Preference("music volume")

##The value the users wants for the music volume.

if config.has_sound:

##If config includes sound.

label _("Sound Volume")

##The sound volume label.

hbox:

bar value Preference("sound volume")

##The value preference for the sound volume.

if config.sample_sound:

##The config for sample sound.

textbutton _("Test") action Play("sound", config.sample_sound)

##The test allows if the sample sound is working making sure the other sounds will work too.
```

```
 if config.has_voice:

 label _("Voice Volume")

 ##The voice volume of the character label.

 hbox:

 bar value Preference("voice volume")

 ##The preference of the value for the voice volume.

 if config.sample_voice:

 textbutton _("Test") action Play("voice", config.sample_voice)

 ##The text button to test the sample voice.

 if config.has_music or config.has_sound or config.has_voice:

 null height gui.pref_spacing

 ##If the config includes sound or voice or music then spacing are made.

 textbutton _("Mute All"):

 ##Allows the users to mute each audio.

 action Preference("all mute", "toggle")

 ##Action preferences that allows to mute audio.

 style "mute_all_button"

 ##The action that allows to mute all the button.

style pref_label is gui_label

##The style pref_label is  set as gui_label.

style pref_label_text is gui_label_text

##The style pref_label_text is set as gui_label_text.

style pref_vbox is vbox

##The style pref_vbox is set as vbox.

style radio_label is pref_label

##The style radio_label is set as pref_label.
```

```
style radio_label_text is pref_label_text

##The style radio_label_text is set as pref_label_text.

 style radio_button is gui_button

 ##The style radio_button is set as gui_button.

 style radio_button_text is gui_button_text

 ##The style radio_button_text is set as gui_button_text.

 style radio_vbox is pref_vbox

 ##The style radio_vbox is set as pref_vbox.

 style check_label is pref_label

 ##The style check_label is set pref_label.

 style check_label_text is pref_label_text

 ##The style check_label_text is set as pref_label_text.

style check_button is gui_button

##The style check_button is set as gui_button.

 style check_button_text is gui_button_text

 ##The style check_button_text is set as gui_button_text.

style check_vbox is pref_vbox

##The style check_vbox is set as pref_vbox.

style slider_label is pref_label

##The style slider_label is set as pref_label.

style slider_label_text is pref_label_text

##The style slider_label_text is set as pref_label_text.

style slider_slider is gui_slider

##The style slider_slider is set as gui_slider.
```

```
style slider_button is gui_button

##The style slider_button is set as gui_button.

style slider_button_text is gui_button_text

##The style slider_button_text is set as gui_button_text.

style slider_pref_vbox is pref_vbox

##The style slider_pref_vbox is set as pref_vbox.

 style mute_all_button is check_button

 ##The style mute_all_button is set ascheck_button.

style mute_all_button_text is check_button_text

##The style mute_all_button_text is set as check_button_text.

style pref_label:

    top_margin gui.pref_spacing

 ##The style pref_label top margin spacing.

    bottom_margin 2

 ##The style pref_label bottom margin.

style pref_label_text:

    yalign 1.0

 ##The yalign for style pref_label_text.

style pref_vbox:

    xsize 225

 ##The x size for the preference box.

 style radio_vbox:

    spacing gui.pref_button_spacing

 ##The spacing for style radio_vbox.
```

```
style radio_button:
    properties gui.button_properties("radio_button")
 ##The default properties of style radio_button.
    foreground "gui/button/check_[prefix_]foreground.png"
 ##The image file for foreground of style radio_button.
style radio_button_text:
    properties gui.button_text_properties("radio_button")
 ##The default properties for style radio_button_text.
style check_vbox:
    spacing gui.pref_button_spacing
 ##The spacing for style check_vbox.
style check_button:
    properties gui.button_properties("check_button")
 ##The default properties of style check_button.
    foreground "gui/button/check_[prefix_]foreground.png"
 ##The image used for foreground of style check_button.
style check_button_text:
    properties gui.button_text_properties("check_button")
 ##Properties of style check_button_text.
style slider_slider:
    xsize 350
 ##The xsize for style slider_slider.
style slider_button:
    properties gui.button_properties("slider_button")
 ##The default properties for style slider_button.
```

```
    yalign 0.5
 ##The yalign of style slider_button.
    left_margin 10
 ##The left margin size of style slider_button.
style slider_button_text:
    properties gui.button_text_properties("slider_button")
 ##The default properties of style slider_button_text.
style slider_vbox:
    xsize 450
 ##The xsize for the slider_vbox.
## History screen
##############################################################################
 ## This is a screen that displays the dialogue history to the player. While there isn't anything
special about this screen, it does have to access the dialogue history stored in _history_list.
 screen history():
tag menu
##The tag menu of screen history.
    predict False
 ## Avoid predicting this screen, as it can be very large.
use game_menu(_("History"), scroll=("vpgrid" if gui.history_height else "viewport")):
##Game menu for history.
        style_prefix "history"
 ##Style prefix for history.
 for h in _history_list:
 ##History list loop.
```

Samig Sherchan

```
            Window:
                has fixed:
  ## This lays things out properly if history_height is None.
                    yfit True
  ##Yfit is automatically sized since its set to True.
if h.who:
  label h.who:
  ##Labels h who the character is who.
                        style "history_name"
  ##The style for history's name.
                        if "color" in h.who_args:
  ## Take the color of the who text from the Character, if set.
  text_color h.who_args["color"]
  ##the text colour for who.
  text h.what
  ##The text color for what.
          if not _history_list:
              label _("The dialogue history is empty.")
  ##If there is no history then a dialogue will state history is empty.
style history_window is empty
##The style history_window is set as empty
style history_name is gui_label
##The  style history_name is set as gui_label.
  style history_name_text is gui_label_text
  ##The style history_name_text is set as gui_label_text.
```

```
style history_text is gui_text
##The style history_text is set as gui_text.
 style history_label is gui_label
 ##The style history_label is set as gui_label.
style history_label_text is gui_label_text
##The style history_label_text is gui_label_text.
style history_window:
    xfill True
 ##The style history x fill is set True.
    ysize gui.history_height
 ##The ysize for the height of history window.
style history_name:
    xpos gui.history_name_xpos
 ##The xposiiton for history_name.
    xanchor gui.history_name_xalign
 ##The history_name xanchor align.
    ypos gui.history_name_ypos
 ##The yposition for history_name.
    xsize gui.history_name_width
 ##The xsize width for history_name.
 style history_name_text:
    min_width gui.history_name_width
 ##The minimum width for style history_name_text.
    text_align gui.history_name_xalign
 ##The text xalign for style history_name_text.
```

```
style history_text:
    xpos gui.history_text_xpos
##The xposition for style history_text.
    ypos gui.history_text_ypos
##The yposition for style history_text.
    xanchor gui.history_text_xalign
##The xanchor for style history_text.
    xsize gui.history_text_width
##The xsize width for style history_text.
    min_width gui.history_text_width
##The minimum text width for style history_text.
    text_align gui.history_text_xalign
##The text xalign for style history_text.
    layout ("subtitle" if gui.history_text_xalign else "tex")
##The layout subtitle for style history_text.
style history_label:
    xfill True
##The xfill set try for style history_label which automatically fills the background.
style history_label_text:
    xalign 0.5
##The xalign size for history_label_text.
```

Samig Sherchan

```
## Help screen
########################################################################

## A screen that gives information about key and mouse bindings. It uses other screens
(keyboard_help, mouse_help, and gamepad_help) to display the actual help.

screen help():

    tag menu

    ##The tag menu for screen help.

    default device = "keyboard"

    ##The default device is keyboard.

    use game_menu(_("Help"), scroll="viewport"):

    ##Game menu includes help function.

    style_prefix "help"

    ##The style_prefix for help.

    vbox:

        spacing 15

    ##The spacing distance for the vbox in help.

    hbox:

    textbutton _("Keyboard") action SetScreenVariable("device", "keyboard")

    ##Horizontal box with the textbutton of keyboards action.

    textbutton _("Mouse") action SetScreenVariable("device", "mouse")

    ##Horizontal box with the textbutton of mouse's action.

    if GamepadExists():

            textbutton _("Gamepad") action SetScreenVariable("device", "gamepad")

    ##Text button for gamepad's actions if it exist.
```

```
if device == "keyboard":

use keyboard_help

##Uses keyboard help.

        elif device == "mouse":

            use mouse_help

 ##Uses mouse help.

        elif device == "gamepad":

            use gamepad_help

 ##Uses gamepad help.

screen keyboard_help():

 hbox:

      label _("Enter")

 ##Help for enter fuction.

      text _("Advances dialogue and activates the interface.")

 ##Dialogue explaining what enter does.

hbox:

      label _("Space")

 ##Help for space function.

      text _("Advances dialogue without selecting choices.")

 ##The dialogue explains the fuction of spce bar.

 hbox:

      label _("Arrow Keys")

 ##Help for function of arrow keys.

      text _("Navigate the interface.")

 ##Dialogue explaining the fuction of arrow keys.
```

```
    hbox:
        label _("Escape")
 ##Help label for escape fuction.
        text _("Accesses the game menu.")
 ##Dialogue explaining the fuction of escape.
    hbox:
        label _("Ctrl")
 ##The help label for ctrl function.
        text _("Skips dialogue while held down.")
 ##Dialogue explaining the ctrl function.
hbox:
        label _("Tab")
 ##The help label for tab function.
        text _("Toggles dialogue skipping.")
 ##Dialogue explaining the tab function.
 hbox:
        label _("Page Up")
 ##The help label for page up function.
        text _("Rolls back to earlier dialogue.")
 ##Dialogue explaining the page up function.
 hbox:
        label _("Page Down")
 ##The help label for pagedown function.
        text _("Rolls forward to later dialogue.")
 ##Dialogue explaining the pagedown function.
```

```
    hbox:
        label "H"
 ##The help label for H key function.
        text _("Hides the user interface.")
 ##Dialogue explaining the H key function.
 hbox:
        label "S"
 ##The help label for S key function.
        text _("Takes a screenshot.")
 ##Dialogue explaining the S key function.
hbox:
        label "V"
 #The help label for V key function.
        text _("Toggles assistive {a=https://www.renpy.org/l/voicing}self-voicing{/a}.")
 ##Dialogue explaining the V key function.
screen mouse_help():
 hbox:
        label _("Left Click")
 ##The help label for left click function.
        text _("Advances dialogue and activates the interface.")
 ##Dialogue explaining the left click function.
 hbox:
        label _("Middle Click")
 ##The help label for middle click function.
        text _("Hides the user interface.")
 ##Dialogue explaining the middle click function.
```

```
hbox:
        label _("Right Click")
##The help label for right click function.
        text _("Accesses the game menu.")
##Dialogue explaining the right click function.
    hbox:
        label _("Mouse Wheel Up\nClick Rollback Side")
##The help label for mouse wheel up function.
        text _("Rolls back to earlier dialogue.")
##Dialogue explaining the mouse wheel up function.
hbox:
        label _("Mouse Wheel Down")
##The help label for Mouse Wheel Down function.
        text _("Rolls forward to later dialogue.")
##Dialogue explaining the Mouse Wheel Down function.
screen gamepad_help():
    hbox:
        label _("Right Trigger\nA/Bottom Button")
##The help label for Right Trigger\nA/Bottom Button function.
        text _("Advance dialogue and activates the interface.")
##Dialogue explaining the Right Trigger\nA/Bottom Button function.
hbox:
        label ("Left Trigger\nLeft Shoulder")
##The help label for Left Trigger\nLeft Shoulder fuction.
        text _("Roll back to earlier dialogue.")
##Dialogue explaining the Left Trigger\nLeft Shoulder function.
```

```
    hbox:
        label _("Right Shoulder")
##The help label for Right Shoulder function.
        text _("Roll forward to later dialogue.")
##Dialogue explaining the Right Shoulder  function.
    hbox:
        label _("D-Pad, Sticks")
##The help label for D-Pad, Sticks function.
        text _("Navigate the interface.")
##Dialogue explaining the D-Pad, Sticks function.
    hbox:
        label _("Start, Guide")
##The help label for Start, Guide function.
        text _("Access the game menu.")
##Dialogue explaining the Start, Guide function.
hbox:
        label _("Y/Top Button")
##The help label for Y/Top Button function.
text _("Hides the user interface.")
##Dialogue explaining the Y/Top Button function.
    textbutton _("Calibrate") action GamepadCalibrate()
##Dialogue explaining the Calibrate function.
```

```
 style help_button is gui_button

 ##The style help_button is set as gui_button.

 style help_button_text is gui_button_text

 ##The style help_button_text is set as gui_button_text.

style help_label is gui_label

##The style help_label is set as gui_label.

 style help_label_text is gui_label_text

 ##The style help_label_text is set as gui_label_text.

style help_text is gui_text

##The style help_text is set as gui_text.

 style help_button:

     properties gui.button_properties("help_button")

 ##The default properties for style help_button.

     xmargin 8

 ##The x margin size for style help_button.

style help_button_text:

     properties gui.button_text_properties("help_button")

 ##The default properties to style help_button_text.

 style help_label:

     xsize 250

 ##The xsize for the style help_label.

     right_padding 20

 ##The right padding for the style help_label.

style help_label_text:

     size gui.text_size

 ##The size of style help_label_text.
```

```
    xalign 1.0

 ##The xalign size for style help_label_text.

    text_align 1.0

 ##The text align size for style help_label_text.

## Additional screens

################################################################################

 ## Confirm screen
################################################################################

 ## The confirm screen is called when Ren'Py wants to ask the player a yes or no question.

 screen confirm(message, yes_action, no_action):

    modal True

    ## Ensure other screens do not get input while this screen is displayed.

 zorder 200

 ##zorder is set 200.

 style_prefix "confirm"

 ##Style prefix for confirm.

 add "gui/overlay/confirm.png"

 ##Image for the confirm.

 frame:

 vbox:

 xalign .5

 ##xalign is set 0.5.

 yalign .5

 ##yalign is set 0.5.

 spacing 30

 ##Spacing is set 30.
```

```
label _(message):
        style "confirm_prompt"
 ##Labeled message confrim prompt.
        xalign 0.5
 ##The xalign is set 0.5 for this message.
 hbox:
        xalign 0.5
 ##The xalign is set 0.5.
        spacing 100
 ##The spacing is 100.
 textbutton _("Yes") action yes_action
 ##The text button for yes actions.
 textbutton _("No") action no_action
 ##The text button for no actions.
 key "game_menu" action no_action
 ## Right-click and escape answer "no".
 style confirm_frame is gui_frame
 ##The style confirm_frame is set as gui_frame.
 style confirm_prompt is gui_prompt
 ##The style confirm_prompt is set as gui_prompt.
 style confirm_prompt_text is gui_prompt_text
 ##The style confirm_prompt_text is set as gui_prompt_text
style confirm_button is gui_medium_button
##The style confirm_button is set as gui_medium_button.
style confirm_button_text is gui_medium_button_text
##The style confirm_button_text is set as gui_medium_button_text.
```

```
 style confirm_frame:
     background Frame([ "gui/confirm_frame.png", "gui/frame.png"], gui.confirm_frame_borders,
tile=gui.frame_tile)
 ##The background frame has an image file.
     padding gui.confirm_frame_borders.padding
 ##The padding for the borders.
     xalign .5
 ##The xalign is set 0.5 for the borders.
     yalign .5
 ##The yalign is set 0.5 for the borders.
 style confirm_prompt_text:
     text_align 0.5
 ##The text align size style confirm_prompt_text.
     layout "subtitle"
 ##The layout subtitle for style confirm_prompt_text.
style confirm_button:
     properties gui.button_properties("confirm_button")
 ##The default properties for style confirm_button.
 style confirm_button_text:
     properties gui.button_text_ style confirm_button_text properties("confirm_button")
 ##The default properties for the style confirm_button_text.
```

Samig Sherchan

```
## Skip indicator screen
##############################################################

## The skip_indicator screen is displayed to indicate that skipping is in progress.

screen skip_indicator():

    zorder 100

##zorder is set 100 for skip_indicator.

    style_prefix "skip"

##The style prefix for skip.

frame:

 hbox:

 spacing 6

 ##Frame spacing is set 6.

 text _("Skipping")

 ##Text showing the skipping action.

 text "▸" at delayed_blink(0.0, 1.0) style "skip_triangle"

 ##Text delayed blink set at different timing.

 text "▸" at delayed_blink(0.2, 1.0) style "skip_triangle"

 ##Text delayed blink set at different timing.

 text "▸" at delayed_blink(0.4, 1.0) style "skip_triangle"

 ##Text delayed blink set at different timing.

 transform delayed_blink(delay, cycle):

    alpha .5

 ## This transform is used to blink the arrows one after another.

pause delay

##Pause delay set.
```

```
    block:
        linear .2 alpha 1.0
##Linear set .2 for block.
        pause .2
##Pause timing set .2.
        linear .2 alpha 0.5
##Liner ser .2 for pause.
        pause (cycle - .4)
##The cycle for the pause fuction.
        Repeat
##Repeats the action when used.
style skip_frame is empty
##The style skip_frame is set empty.
style skip_text is gui_text
##The style skip_text is set gui_text.
style skip_triangle is skip_text
##The style skip_triangle is set as skip_text.
style skip_frame:
    ypos gui.skip_ypos
##The yposition for style skip_frame.
    background Frame("gui/skip.png", gui.skip_frame_borders, tile=gui.frame_tile)
##The image file for the background frame of style skip_frame.
    padding gui.skip_frame_borders.padding
##The padding for the frame of style skip_frame.
```

```
style skip_text:

    size gui.notify_text_size

##The size of the style skip_text in default.

style skip_triangle:

 ## We have to use a font that has the BLACK RIGHT-POINTING SMALL TRIANGLE glyph in it.

    font "DejaVuSans.ttf"

##The font of the style skip_triangle.

## Notify screen
##############################################################################

## The notify screen is used to show the player a message. (For example, when the game is
quicksaved or a screenshot has been taken.)

screen notify(message):

zorder 100

##zorder set for screen notify messages.

    style_prefix "notify"

##The style prefix for notify.

frame at notify_appear:

        text message

##Text message frame for notify_appear.

timer 3.25 action Hide('notify')

##Timer for the action hide to be notified.

transform notify_appear:

    on show:

        alpha 0

##alpha for notify_appear set to 0.
```

```
        linear .25 alpha 1.0

  ##The linear set for notify_appear.

on hide:

        linear .5 alpha 0.0

  ##The linear set for on hide.

style notify_frame is empty

##The style notify_frame is set empty.

  style notify_text is gui_text

  ##The style notify_text is set as gui_text.

style notify_frame:

     ypos gui.notify_ypos

  ##The yposition for style notify_frame.

  background Frame("gui/notify.png", gui.notify_frame_borders, tile=gui.frame_tile)

  ##Background frame for style notify_frame.

     padding gui.notify_frame_borders.padding

  ##The padding for style notify_frame.

style notify_text:

     size gui.notify_text_size

  ##The size for style notify_text
```

Samig Sherchan

```
## NVL screen
##############################################################################

## This screen is used for NVL-mode dialogue and menus.

screen nvl(dialogue, items=None):

window:

        style "nvl_window"

##The style for the nvl window.

has vbox:

            spacing gui.nvl_spacing

##spacing for the vbox.

if gui.nvl_height:

## Displays dialogue in either a vpgrid or the vbox.

vpgrid:

                cols 1

##Column for the vpgrid.

                yinitial 1.0

##The y intial for the vpgrid.

use nvl_dialogue(dialogue)

##Use of nvl dialogue.

else:

use nvl_dialogue(dialogue)

##Use of nvl dialogue for a different statement.

for i in items:

## Displays the menu, if given. The menu may be displayed incorrectly if config.narrator_menu
is set to True, as it is above.
```

```
textbutton i.caption:

        action i.action

##Text button for items.

style "nvl_button"

##The style for the nvl button.

add SideImage() xalign 0.0 yalign 1.0

##adds side image at these size of xalign and yalign.

screen nvl_dialogue(dialogue):

for d in dialogue:

##Loop for the dialogue.

window:

    id d.window_id

##window id.

fixed:

        yfit gui.nvl_height is None

##The yfit height of the fixed window.

if d.who is not None:

##If d.who is set none then.

text d.who:

  id d.who_id

##The text d who id.

text d.what:

id d.what_id

##The text for d what id.
```

```
define config.nvl_list_length = 6

## This controls the maximum number of NVL-mode entries that can be displayed at once.

style nvl_window is default

##The style nvl_window is set default.

style nvl_entry is default

##The style nvl_entry is set default.

style nvl_dialogue is say_dialogue

##The style nvl_dialogue is set as say_dialogue.

style nvl_button is button

##The style nvl_button is  set as button.

style nvl_button_text is button_text

##The style nvl_button_text is set as button_text.

style nvl_window:

    xfill True

##The xfill set true for style nvl_window.

    yfill True

##The yfill set true for style nvl_window.

background "gui/nvl.png"

##The image file used for the fill.

    padding gui.nvl_borders.padding

##The padding for the nvl borders.

style nvl_entry:

    xfill True

##The xfill set true for style nvl_entry.

    ysize gui.nvl_height

##The ysize for style nvl_entry.
```

```
style nvl_label:
    xpos gui.nvl_name_xpos
##The xposition for the style nvl_label.
    xanchor gui.nvl_name_xalign
##The xanchor for style nvl_label.
    ypos gui.nvl_name_ypos
##The yposition for the style nvl_label.
    yanchor 0.0
##The y anchor for style nvl_label.
    xsize gui.nvl_name_width
##The xsize for style nvl_label.
    min_width gui.nvl_name_width
##The minimum width for style nvl_label.
    text_align gui.nvl_name_xalign
##The text align for style nvl_label.
style nvl_dialogue:
    xpos gui.nvl_text_xpos
##The xposition for style nvl_dialogue.
    xanchor gui.nvl_text_xalign
##The xanchor for style nvl_dialogue.
    ypos gui.nvl_text_ypos
##The yposition for style nvl_dialogue.
    xsize gui.nvl_text_width
##The xsize for style nvl_dialogue.
```

```
    min_width gui.nvl_text_width

##The minimum width for style nvl_dialogue.

text_align gui.nvl_text_xalign

##The text align for style nvl_dialogue.

    layout ("subtitle" if gui.nvl_text_xalign else "tex")

##The layout for style nvl_dialogue.

style nvl_thought:

    xpos gui.nvl_thought_xpos

##The xposition for the style nvl_thought.

    xanchor gui.nvl_thought_xalign

##The xanchor for style nvl_thought.

    ypos gui.nvl_thought_ypos

##The yposition for style nvl_thought.

    xsize gui.nvl_thought_width

##The xsize for style nvl_thought.

    min_width gui.nvl_thought_width

##The minimum for style nvl_thought.

    text_align gui.nvl_thought_xalign

##The text align for style nvl_thought.

    layout ("subtitle" if gui.nvl_text_xalign else "tex")

##The layout for style nvl_thought.

style nvl_button:

    properties gui.button_properties("nvl_button")

##The default properties style nvl_button.

    xpos gui.nvl_button_xpos

##The xposition style nvl_button.
```

```
    xanchor gui.nvl_button_xalign

  ##The xanchor for style nvl_button.

  style nvl_button_text:

    properties gui.button_text_properties("nvl_button")

  ##The default properties of style nvl_button_text.

## Mobile Variants

################################################################################

  style pref_vbox:

    variant "medium"

  ##The variant for style pref_vbox.

    xsize 450

  ##The xsize of style pref_vbox.

screen quick_menu():

    variant "touch"

  ## Since a mouse may not be present, we replace the quick menu with a version that uses
fewer and bigger buttons that are easier to touch.

  zorder 100

  ##The zorder is set 100.

    hbox:

      style_prefix "quick"

  ##The style prefix of quick.

  xalign 0.5

  ##The xalign of the hbox.

  yalign 1.0

  ##The yalign of the hbox.
```

```
    textbutton _("Skip") action Skip() alternate Skip(fast=True, confirm=True)
```

##The text button for skip.

```
    textbutton _("Menu") action ShowMenu()
```

##The text button for menu.

```
    textbutton _("Auto") action Preference("auto-forward", "toggle")
```

##The text button for preference.

```
style window:
    variant "small"
```

##The variant for the window.

```
    background "gui/phone/textbox.png"
```

##The background for this style window.

```
style nvl_window:
    variant "small"
```

##The variant for the style nvl_window.

```
    background "gui/phone/nvl.png"
```

##The background of style nvl_window.

```
style main_menu_frame:
    variant "small"
```

##The variant for style main_menu_frame.

```
    background "gui/phone/overlay/main_menu.png"
```

##The background for style main_menu_frame.

```
style game_menu_outer_frame:
    variant "small"
```

##The variant for style game_menu_outer_frame.

```
    background "gui/phone/overlay/game_menu.png"
```

##The background for style game_menu_outer_frame.

```
  style game_menu_navigation_frame:
      variant "small"
  ##The variant style game_menu_navigation_frame.
      xsize 340
  ##The xsize style game_menu_navigation_frame.
  style game_menu_content_frame:
      variant "small"
  ##The variant style game_menu_content_frame.
      top_margin 0
  ##The top margin style game_menu_content_frame.
  style pref_vbox:
      variant "small"
  ##The variant style pref_vbox.
      xsize 400
  ##The xsize style pref_vbox.
  style slider_pref_vbox:
      variant "small"
  ##Variant style slider_pref_vbox.
      xsize None
  ##xsize for style slider_pref_vbox
style slider_pref_slider:
      variant "small"
  ##Variant of style slider_pref_slider.
      xsize 600
  ##The xsize for style slider_pref_slider.
```

Samig Sherchan

Game code:

```
label start:

##Label to start the game.

init python:

##Initialises python statement to define something.

    m = Character("Monokuma")

##m is declared as the character Monokuma, so in the game screen, when the character is talking the m will be shown as Monokuma.

scene scene1

##This is the image file scene1 which is used as the background scene.

show monokuma

##Shows the image file monokuma in the middle.

m "I am Monokuma. Upupupupupu... Don't take it wrong you weren't kidnapped or something, but you surely stink, go take a shower."

##Monokuma's dialogue for the first page which is the introduction.

m "What I mean to say was, you have been selected to investigate a murder and find the culprit! But the question is, can you really solve it ?"

##The dialogue in a different page questioning the user.

m "No need to dwell too much, by the way I haven't heard anything from you, what's your name?"

##A question is asked which is making the user input its name.

define pov = Character("[povname]")

##This defines pov and it also has a parameter which lets the user input their name, and wherever there is pov, the name inputed will be shown instead.

python:

    povname = renpy.input("Please enter your name")

##The input dialogue that tells the user what to do.
```

```
    povname = povname.strip()

  ##Povname defined.

    if not povname:

  ##If the name isn't inputed then they can press enter.

        povname = "Makoto Naegi"

  ##Pressing enter will give the user this default name.

  hide monokuma

  ##Hides the image file monokuma.

  pov "My name is [povname], I don't where I am but let's settle this with some questions!"

  ##This shows the input of the user for the name, and in the dialogue [povname] will display the
output the sentence.

  show monokuma

  ##Shows monokuma image again.

  m "Hmmmmm... So what do you want to ask me, [povname]?"

  ##Monokuma's dialogue with the input from the user (name).

  hide monokuma

  ##hides monokuma image again.

  menu start_from:

  ##Choices menu given to the player with list of options below.

        "How to Play":

  ##The choice will appear as this.

          $ question = "Game Knowledge"

  ##Defining the choice.

          pov "I'll would like to understand how to escape!!"

  ##The main character will speak when choising the statement.
```

```
        "What happens if I get an answer wrong?":
##Second choice the user can choose.
            $ question = "Consequences"
 ##The choices is defined.
            pov "There must be a catch behind this"
 ##The main character with a different statement if chosen.
        "Despairful Secret!":
 ##Third choice.
            $ question = "Key to escape!"
 ##The choice is defined as this.
            pov "There must be another way to escape this despair!"
 ##The dialogue from the main character.
        "Lets start!":
 ##The last choice which begins the game.
            $ question = "Hope!"
 ##Choice defined as this.
            pov "I shall overcome despair!"
 ##The statement from the main character.
show monokuma
##Shows the monokuma image file.
 if question == "Game Knowledge":
 ##If statement for the first choice made, the statement continues here.
    m "Now, that's the spirit! I see you are interested in the game"
 ##Monokuma's dialogue.
    pov "No thats not it! But go ahead!"
 ##User's character or main character's dialogue.
```

m "Well there is culprit behind each murders, and you are given a question with false answers, , your job is to work the real answer out and counter the culprit. There are multiple choices"

##Monokuma's dialogue explaining the game

m "However, some question from other characters might be true, meaning they aren't necessarly the culprit, so do be careful!"

##Monokuma explaining the consequences of the game.

call start_from from _call_start_from

##Goes back to the menu statement which was called start_from. The call function makes the in-game menu iterate so the user can go each choices much as they like.

if question == "Consequences":

##The second choice's if statement which continues the main characters dialogue.

m "Upupupupupu, if you get a True or False statement wrong, you'll slowly lose trust from other characters."

##The affect of the wrong choices having an effect on the surrounding.

m "Furthermore, if you get all answers wrong then..."

##More information about the game.

m "You won't be just suspected as a culprit, you will die too, plus all the other innocent memebers. While the culprit gracefully escapes all alone. Upupupupu!"

##The final consequences when you lose the whole game.

m "Plus you won't be able to continue, so a restart or you should save your progress all the time, if you do lose then in the credits you'll get the real answer and on how to answer the question"

##Explaining how to prevent the lose of data, since losing will cause restarts.

call start_from from _call_start_from_1

##call function which will redirect the user back to the in-game choices.

if question == "Key to escape!":

##If statement for the third choice.

```
  m "Aww, you caught me!"

  ##The Monokuma character's dialogue.

      m "Psyched! There is nothing to see here, now hurry up and get those brain cells running! "

  ##Another statement for Monokuma's humour.

      call start_from from _call_start_from_2

  ##Call statement that goes back to the ingame menu.

if question == "Hope!":

##The last choice and the if statement.

  m "Upupuppu! This is what I am talking about! Let's see if you fall in to my sweet sweet despair!"

##Statement made by monokuma.

   m "With no further ado, I will have unconscious, in 5 secs. Upupupupupu."

  ##Monokuma explaining the current situation.

   pov "Wait what do mean lossing my consciousness. I never even agreed to this in the first
place!"

  ##The user's character is questioning monokuma.

   hide monokuma

  ##Hides the image file of monokuma.

  label looking:

  ##A new label which changes the situation and changes the scene and enables narration.

  scene pr

  ##Scene for the new label, a new background.

  "Looking around, [povname] tried to find an escape route planning to escape from this horror.
It was like an underground cell, almost as in [povname] had been imprisioned like the culprit."

  ##Narration about the vision of the main character.

  "There were no windows, the only place left to escape was the ventalation system and a
metallic door which seemed unbreakable"

  ##Naration about the scene and escaping ideas.
```

Samig Sherchan

"Without no hesitation [povname] ran to the vents and started to climb the ladder which connected to the vents."

##The main character climbs the ventalation in desperation.

"This looked awfully well prepared almost like [povname] was carrying out the exact stages from Monokuma's plan."

##The room was awfully made out which seemed like a trap.

"Hisss... Sudden noise alerted [povname], it seemed to be coming from the vents, just when was about to reach for the vents, a smoke deployed all over the room."

##Narartion about a noise allerting the main character.

"Causing [povname] to choke, so he decided to return back down, but he slowly started to grow dozzy... Just like that he felt asleep in the floor."

##The effect it had the main character.

label new:

##New label starting a new scenario.

scene dr

##A new scene/image which matches the naration below.

"Everything was pitch black, no sense of hope but just despair. [povname] was asleep via Monokuma's sleeping gas which unexpectedly came through [povname] escape route."

##The naration describing the field vision of the the main character's unconciousness.

"To Monokuma it was all part of the play. Plently occured through out the past hour."

##The monokuma's plan.

"Had [povname] really disappeared in the depth of abbys? [povname] mentality snapped, with no understanding the cause and finding the solution of the problem."

##The main character failing to gain conciousness.

"Almost as in he was trapped by the tainted darkness hidden in his heart."

##The feelings of the main character explained.

scene rd

##A new scene changing the the story slowly.

Samig Sherchan

"Suddenly a noise gives hope to [povname], Almost like a ray of light coming to save [povname]. The louder the noise got the bigger the ray of light shined mercifully demolishing all the darkness."

##The naration for the scene changing, as the main character is slowly regaining it's conciousness.

scene lr

##The scene changes.

"The darkness comepletly vanished, then the noise converted to a voice.

##Intialising python.

n = Character("???")

##n is defined as an character with no name to make the character mysterious.

n "You ok? Someone help! A person is lying on the floor here! Ahhhh! I don't want people think I killed him! I'm innocent!"

##Statement of the mysterious character, the user won't know who it is, because it is labeled as '???'.

label death:

##New label to create a new scene.

"Speaking of the floor, it was rather warm and the floor wasn't solid hard, he could manipulate it or carry the part of the floor with his palm, [povname] felt relaxed and comfortable."

##The main characters senses being displayed in the from of dialogue.

"However, thinking back at the statement from a feminine voice she stated the word 'kill' & 'innocent'. This straight away made [povname] have flashbacks of the horror he been through ages ago"

##A voice alerts the main character.

"With no hesistation he force himself to woke up"

##The character is awoken due to the femine voice.

"Front of him was feminine figure, which was hard to adjust her image as his eyes had difficulties getting used to the sudden light."

##The main character slowly regains their vision.

Samig Sherchan

scene beach

##The scene for the surrounding described on earlier.

show intro girl at right

##The blurred image girl when the main character is regaining their vision adjusted at right.

"It was beach given, that explains the warm ground & the ability for [povname] to feel the sand."

##A dialogue explaining what the main character felt early on.

"The image grew more clear and clear, standing front of [povname] was..."

##The image of the feminine figure growing more clearer and clearer.

hide intro girl

##Hides the blurred image of the girl.

show girl at right

##The clear image of the girl set at right.

with dissolve

##Using the dissolve transtition to show the change.

"A beautiful woman, she seemed to be around the same age as [povname]"

##Dialogue introducing the new character.

"The only problem surrounded [povname] was she had a royal background with her crown and dress she isn't someone you should casually talk to."

##The dialogue shows the background of the feminine character based on the appearance.

"Without no hesitation, he spoke"

##The main character speaks in desperation.

pov "Uhm, where am I?"

##Main character questioning the female character.

n "(Sighs in relief), I am glad your fine. Otherwise the public would suspect me for killing, it is a great damage towards my prestige and reputation."

##The girl (labeled '???') speaking about the consequences she could face and the hysteria.

pov "(Trying to speak at the same level as a royal) Uhm. Miss mind if I ask your name?"

##Main character trying to speak at the level due to her appearance.

n "You don't know me, I feel heartbroken, you idiot! Idiot! Idiot! But fine since you insist. I am the great princess of Geneva. Lucy Akiho Bella Elizabeth Bee'l VIII. Hope you remember that!"

##Full introduction to the female character, with angry emotions.

pov "Is it fine if I just call you Lucy?"

##The main charcter asking if he should pronounce by that name.

init python:

##Intialising python for defining.

p = Character("Lucy")

##p is defined as Lucy, so when the p is used for dialogue it will appear as Lucy.

p "Sure. Wait a minute, how rude can you be! Thats Princess Lucy to you! You idiot! Idiot!"

##Lucy anger directed towards the main charcter due to lack of respect.

pov "(Teases the princess with a smug face) You sound more of tsundere than a princess..."

##The main character's reaction.

p "Hmmmph can't stand rude people like you. I'm leaving you here all alone!"

##The reply from the princess.

pov "Where are you going you don't even know anything about this place...?"

##The question from the main character.

p "Ahhhh! Fine I'll walk with you"

##The question startles the Lucy.

pov "Ahahahaha! Your funny lets check the beach house front of us."

##The main character thinking about the plan.

p "Your so mean to me. But ok"

##Despite the anger from the princess, she agrees with the plan.

label searching:

##New label to start the journey throught the beach.

scene bh

##New scene showing the beach house.

"Just behind the beach there was a beach house which looked new & refurbished, maybe going there will be the key of [povname] & Lucy of solving the real truth behind this place."

##Describing the beach house.

"Another problem about the beach was it had been bothering [povname] too much was it was awfully isolated despite the beauty of the beach surely people would come to relax here."

##However, the beach seemed awfull well made but so isolated, which is growing concern to the main character.

"Without further ado after reaching the beach house, they opened the door & front of them was..."

##The mystery getting tense.

"And inside was..."

##Making the user curios about what's going to occur.

scene inside

##A new scene to show the inside area of the beach house.

init python:

##Initialising python to define a character.

fg = Character ("???")

##fg is defined as character '???' to make it mysterious since the user doesn't know the person's name.

show cha at right

##Shows the image of the new character.

fg "Hi, how are you. My name is Alphonse!"

##The mysterious character introducing themselves.

fg "Sorry if I startled you, the rest are in the other rooms. By the whats your name and the lovely princess next to you?"

##The mysterious character questioning the main character and Lucy about their names.

label noturn:

##New label to start a new scene.

show girl at left

##The image of the character Lucy is shown in the left.

p "It's Princess Lucy Akiho Bella Elizabeth Bee'l VIII. See [povname] atleast he has respect why can't you treat me the same way!"

##Introducing the character Lucy to the mysterious character.

pov "Just Lucy, sounds better."

##Main character reaction to the long name.

p "Hmmmph!"

##Reaction from the princess.

init python:

##Initialising the character

    f = Character ("Alphonse")

##f is defined as Alphonse since the character isn't mysterious anymore.

f "Ah so you are [povname], thats a nice name."

##The new character complementing the user's name/main character's name.

pov "Nice to meet you Alphonse."

##Main character greeting back.

f "I don't know much how we got here, but all I remember was toy bear"

##Alphonse talking about something that will cause trouble to the characters.

pov "That b*stard! He's the reason"

##The reaction of the main character due to what occurred at the start.

p "[povname]... Well I also understand how you feel, if I remember I was riding my carriage and all of the sudden fog suddenly appeared then I fell asleep."

##Lucy explaining the similar situation how she happened to be here.

pov "Wait a minute when you said there was fog! It must be the sleeping gas, its quite strong, I feel you too Lucy!"

##The explaination of the sleeping gas.

p "Princess Lucy to you!"

##Reaction from the princess.

f "Now, now no need to rage, raging won't get us anywhere, oh [povname] can I ask you a favor?"

##Alphonse asking the main character a favor.

pov "Uhmmm sure..."

##The user consents in a confused manner.

f "Could you check some rooms and call two people back in the dining hall? We could discuss about how we get here and understand how to escape from this place."

##The male character is requesting to bring people back.

p "I'll help!"

##Lucy is eager to help.

pov "Sure! Hmmmm, Lucy helping me..."

##The user character questioning the realiability of the princess.

p "Whats wrong with me helping!!!!!!!"

##Lucy raging due to that comment.

f "Thank you, it nice to see you guys get along well, like lovers!"

##Alphonse trying to tease and cause more tension between the current relationship between the two characters.

p "Who would even go out with a rude guy like him!!!"

##The sudden reaction to that comment.

```
pov "That hurts my feeling... Princess like you are to arrogant!"

##The user reaction and reply to the princess.

p "Hmmph"

##The princess's emotion.

pov "Hmmph!"

##The users reaction.

hide girl

##Hides image of Lucy.

hide cha

##Hides image of Alphonse.

menu rooms_from:

##New in-game menu/choices for the users to interact.

    "Living Room":

##First choice displayed.

        $ question = "Person 1"

##The choice is defined as this.

        pov "I guess a gamer or a hardcore soap opera person is going to be there."

##A short statement made by the main character.

        p "Maybe"

##Lucy's short statement.

    "Balcony":

##Second choice displayed.

        $ question = "Person 2"

##The second choice defined as this.

        pov "Hmmmm, the balcony will surely have a nice view."

##The reaction of the main character at this second choice.
```

p "I know right... I really used to love looking at the sunset when my father used to take us to a beach"

##Lucy's reply to the view.

pov "Private beach"

##User character teasing about her wealth.

p "Still the same!"

##Princess denying the user's statement.

"Kitchen":

##Third the choice displayed.

$ question = "No Person!"

##The third choice is definied as this.

pov "I wonder who's eating at the kitchen but boy I am sure hungry."

##User character hunger strikes.

p "You call yourself a man! (Growl)"

##The princess tries to embarress the user, however her stomach growls making her statement invalid.

pov "(Smug face) Hmmmm..."

##The user character finds it humurous.

p "I mean nothing!"

##To embarresed to continue to talk.

"Head back to the Dinning Hall":

##The display of the last choice.

$ question = "Head back"

##The last choice is defined as this.

pov "Time to head back don't you think. We called everyone."

##The main character's consideration.

```
        p "I guess so."

##The princess agrees.

init python:

##The python is intialised for the defining the character.

    y = Character ("???")

##The character is appeared mysterious untill the introduction of the character.

init python:

##The python is intialised to define the character.

    yr = Character ("Yuri")

##The yr is defined as Yuri.

if question == "Person 1":

##If statement of the first choice if its chosen.

    scene lving

##Shows the scene of the living room.

    pov "Lets see who's here?"

##The main character searches for people.

    show girl at left

##The image of Lucy shown in the left.

    p "It looks rather comfy. I could fall asleep."

##Lucy's reaction to the comfort of the room.

    pov "Then I'll leave you here"

##The main character teasing the female character.

    p "Don't be mean!"

##Reaction to that statement.

    y "Uhmm excuse me?"

##A mysterious character questions the two.
```

Samig Sherchan

show pm at right

##The image of mysterious character shows up.

hide girl

##Hiding the image of the Lucy.

y "May I help you?"

##The mysterious asking if they need assistance.

pov "Ah Alphonse is looking for you waiting for you at the living room."

##The main character explaining the situation.

y "So sorry you had to waste your precious time to come and get me."

##The mysterious character appologising for the trouble.

pov "Don't worry its fine, by the way the name is [povname] nice to meet you."

##The main character introducing to the mysterious character.

y "I'm Yuri nice to meet you too. I just love books and found so many limited edition book here."

##The mysterious character introducing himself, now the the mysterious character name is converted to Yuri.

yr "I shall head back"

##Yuri heads back.

hide pm

##Image of Yuri is hidden to make it look like he left.

pov "One done, how many to go? Hmmmmm"

##The main character questioning about the favor.

show girl

##The image of Lucy is shown.

p "Hey! Hey! I been ignored for the whole time!"

##The anger of Lucy being ignored as she didn't even speak for one second.

```
    pov "(Ignoring Lucy) Maybe I should take a book too."

##The main character ignoring Lucy for their humour.

    p "Don't ignore me!"

##The reaction to the action.

    hide girl

##Hides the image of Lucy.

    call rooms_from from _call_rooms_from

##Goes/calls back to in game menu so the user can view other choices.

if question == "Person 2":

##The if statement of the second choice.

    scene sv

##Scene of the current room.

    pov "I'm sure Alphonse said there were two people"

##Due to the empty room the main charcter is concerned.

    p "Then where is that person?"

##Lucy also questioning.

    pov "We should check somewhere else."

##The main character requesting to go to another room.

    call rooms_from from _call_rooms_from_1

##The call back to the ingame menu.

if question == "No Person!":

##The if statement for the third choice.

    pov "Lets come here sometimes"

##The main character reaction to the beach view has attracted the main character.

    p "Uhmm sure ahaaha..."

##Lucy agree to the sudden reaction.
```

```
    call rooms_from from _call_rooms_from_2

  ##The ingame menu is called again.

  if question == "Head Back":

  ##The last choice if statement.

      scene inside

  ##Showing the image of the first room they went.

      show cha at right

  ##Showing image of Alphonse.

      f "(Depressed tone) Welcome back..."

  ##A sad voice made by Alphonse.

      show pm

  ##Shows Yuri image.

      yr "I'm afraid someone died again..."

  ##Explaining the situation.

  label again:

  ##New label to start the new stage of the story.

  "The moment Yuri stated again, that reminds [povname] something Monokuma said about murder"

  ##Dialogue explaining to the main character about the murder.

  "Maybe it already happened and it is continuing even here. [povname] look around and on the floor of the dining table was..."

  ##The main character looked at their field of vision.

  "A fresh corspe that died recently with internal stabbing which caused loss of blood. This could mean there is a serial killer hiding somewhere."

  ##The view of the corspe.

  "Observing the body, the victim had 3 stabs"

  ##The detail about the murder.
```

Samig Sherchan

pov "Damn! Someone died again! There must be someone lurking around here!"

##The main character's reaction to the death.

show girl at left

##Shows the image of Lucy.

p "No...No!!! (It would be a shock for a royal to see someone to die in front of them."

##Lucy reaction to the death as everyone wen't hysteric.

hide pm

##Hiding the image Yuri.

"Then a familiar voice appeared, it did annoy [povname]"

##A voice that was heard before iriating the main character.

m "Not bad [povname] you guessed it right there is a serial killer but..."

##Monokuma appears.

pov "I knew you were here! Get us out of here. And you got any more to say."

##The main character wants to escape.

show monokuma

##Shows the image of monokuma

m "The serial killer is one of you!"

##Monokuma giving hints about the serial killer.

pov "Wait what!!!!!!!"

##The main character is failing to believe the words of monokuma.

p "No this can't be!"

##Lucy is also denying the statement.

yr "This is impossible!"

##Yuri's reaction.

f "Stop making everyone paranoid monokuma."

##Alphonse trying to calm the situation down.

pov "Maybe your lying monkuma."

##The main character still questioning monokuma.

m "No I'm not..."

##Monokuma with no hestiation says no.

m "Anyway lets get to the questions!"

##Monokuma begins the main game.

hide girl

##Hides Lucy image.

hide cha

##Hides Alphonse's image.

"Suddenly everyone is teleported to a random room"

##A dialogue explaining the sudden change.

scene pl

##The new scene of the gameroom.

m "Right the first question"

##The first question is asked.

pov "Wait we didn't even agree to this!?"

##The main character can't tolerate this.

m "Didn't I tell you well I told everyone in the begining!"

##Monokuma trying to remind what was told in the beginning.

m "Right no more distrubance"

##Monokuma showing full authority.

"[povname] had no choice but to get on with it. Everyone felt the same way. Everyone was hoping monokuma was lying. But who knows."

##Without hesitation Monokuma started to ask question.

m "First question! 8^-2/3 (8 to the power of -2/3)"

Samig Sherchan

```
##The first question about indicies is asked.

p "Isn't that 1/4 Monokuma? What do you think [povname]?"

##Lucy answers but questions the main character for agreement.

menu first_question:

##The new ingame menu for the first question.

        "Yes":

##First choice for the agreement.

            $ question = "A1"

##The first choice is defined.

            pov "That is certainly the right answer, the 8 becomes a reciprocal, then you cube root the dominator (8) and square the it which equals to 4."

##Short statement made by the main character.

        "No":

##Second choice for disagreement.

            $ question = "A2"

##Second choice is defined.

            pov "Uhmm wait really. It thought it was 4."

##Short statement about the disagreement.

if question == "A1":

##The if statement for the first choice.

    p "Yup, I'm surprised you know it."

##Lucy is content with the answer.

    show girl at left

##Image of Lucy shown.

    pov "Well it isn't much ahaha."

##The main character showing his why reaction.
```

Samig Sherchan

```
    show monokuma

##Shows the image of monokuma.

    m "I'll lend you a hint, the letter 'ESN'! 'Well on to the next quesiton"

##The hint to the killer due to the right answer.

    m "Differentiation!"

##Next topic of the question.

    m "Which one of the following is the differential of this equation y = 2x^5+4x^2+5 (2x to the power of 4 + 4x to the power of 2 + 5)?"

##The equation to be differentiated.

    show cha at right

##Alphonse image shown.

label next_one:

##A new label for the second question.

f "Its dy/dx = 10x^4x^2+5"

##Alphonse answers with no hestiation.

m "Do we all agree?"

##Monokuma asking about everyone's opinion.

f "Of course its right!"

##Alphonse declaring it.

menu counter_die:

##Another choices for the answer of Alphonse.

  "Yes":

##First choice in agreement.

      $ question = "CS1"

##The first choice is defined this.

      pov "It must be right"
```

Samig Sherchan

```
##A short statement of the user.
   "No":
##The second choice.
      $ question = "CS2"
##Second choice is defined as this.
      pov "No thats wrong!"
##Denying the question.
if question == "CS1":
##If the first choice is chosen.
    show cha at right
##Shows the image of Alphonse.
    f "Your all wrong! I win!"
##Alphonse denies his own answer.
    show girl at left
##Shows the image of Lucy.
    p "What do you mean?"
##Questioning in confusion.
    show pm
##Shows the image of Yuri.
    yr "You mean your the killer!"
##Yuri suspects Alphonse.
    f "Thats right! I killed her! I am the serial killer!"
##Alphonse reveals himself.
    pov "Why? How could you!"
##The main character questioning.
    f "Killing is esctasy!"
```

```
##Alphonse's reply.

    hide pm

##Hides Yuri.

    show monokuma

##Shows the image of monokuma

    m "Oh well you all die and the culprit escapes!"

##The serial killer escapes.

    m "You lose!"

##Monokuma declares this game lose.

    scene gmver

##Scene of game over.

    m "Die in Despair!"

##With a short sentence.

    Return

##Ends the game back to to home screen.

if question == "CS2":

##The second choice of the menue.

    f "What do you mean?!?!?"

##Alphonse confused with anger.

    pov "The five disappears because there is no x or a power. When you differentiate a integer it is removed. So the real answer is dy/dx = 10x^+8x!"

##The main character corrects Alphonse's answer.

    show monokuma

##Shows image of monokuma.

    m "Pom, pom pom! Your all right! Awww the hope side wins... Well now you know the killer is ESNOHPLA"
```

Samig Sherchan

##Monokuma declares the game as win. With the full hint of the killer.

show girl at left

##Shows the image of Lucy on the left.

p "Its Alphonse!"

##She figures out.

f "Wait what!!! You got no proof!"

##Alphonse denies.

p "Esnohpla reversed spells out Alphonse!"

##Lucy gives the answer.

pov "Not only that, I was at the beach with Lucy, while Yuri was reading the book! While you were in the dining hall you had all the time needed to kill someone!"

##The main character explains the possibilities.

m "Your all right!"

##Monokuma agrees.

m "You all escape while the killer dies!"

##The consequence of the killer.

f "No please its not me its not me!"

##Alphonse grieves in fear.

m "Hush! It won't be painful! Upupupupu!"

##Monokuma declaring Alphonse to quiet, with happiness.

hide monokuma

##Hides monokuma image.

show pm

##Shows image of Yuri.

yr "Now we can escape!"

##He states about escaping.

pov "Yeah!"

##The main character agrees.

Return

##Ends the game with a win.

if question == "A2":

##The second choice if statement from the first question (indicies) This for the 'No' Choice before differentiation came up.

p "How can you not know that answer!!!"

##Lucy is shocked about the answer.

show girl at left

##The image of Lucy on the left.

pov "Wait I thought..."

##The main character confused about the answer.

show pm at right

##Yuri shows up.

yr "Uhm you, first cube root the 8 which equals to 2 then you square it which equals the answer to 4"

##Yuri explains the answer.

show cha

##Shows Alphonse.

f "Oh my [povname] could it be you killer of this victim?"

##Alphonse suspects the main character as he is starting to lose the trust.

pov "Wait I was with Lucy all day long!"

##The main character trying to deny it.

f "Well what were you doing before you went to search for Yuri?"

##Alphonse trying to question even more.

```
    pov "Well I was in the beach unconcious."

##The main character explains the situation.

    f "Hmmm suspicious! You have no alibi!..."

##Alphonse still not sure about his answer.

label thinking:

##New label.

"Thinking back, [povname] needed some evidence. [povname] must surely have an alibi. But where?"

##The main character needs a counter question.

menu counter_question:

##Ingame menu for the counter.

        "Beach":

##The first choice for the ocunter.

            $ question = "C1"

##Defined for the first choice.

            pov "Wait I do have an alibi! Lucy met in the beach!"

##The main character explains about the alibi.

        "Monokuma":

##Second choice.

            $ question = "C2"

##Defined for the second choice.

            pov "Well Monokuma brought me here!"

##The short sentence from the main character.

        "Yuri":

##Third choice.

            $ question = "C3"
```

```
##Third choice is defined.
        pov "Wait I met Yuri!"
##The dialogue about meeting Yuri.
if question == "C1":
##The if statement for first choice.
    p "Wait I found him in the beach laying, then I woke him up!"
##Lucy explaining the situation.
    show girl at left
##Shows image of Lucy.
    show cha at right
##Shows image of Alphonse.
    f "Hmmmm... Still doesn't mean your not supicious."
##Alphonse still concerned.
    pov "Huh...?"
##The main character confused.
    p "Aren't you just accusing him randomly!"
##Lucy questioning Alphonse.
    show monokuma
##Image of Monokuma.
    m "Alright lets go the next question."
##Goes to the next quesiton.
    call next_one from _call_next_one
##Goes back to the the differentiation question.
if question == "C2":
##The second if statement.
```

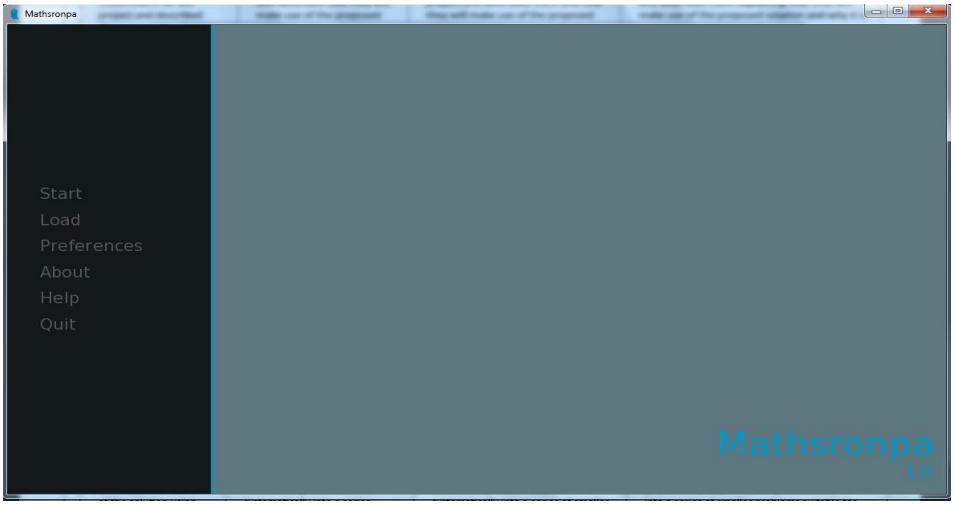
```
        show monokuma

    ##Shows image of monokuma.

        m "Hmmm that doesn't prove anything"

    ##Monokuma explains its invalid.

        show cha

    ##Shows the image of Alphonse.

        f "Exactly, you must be the murder!"

    ##Alphonse backing up monokuma.

        pov "Wait thats not it!"

    ##The main character denying it.

        call counter_question from _call_counter_question

    ##Goes back to counter ingame menu.

if question == "C3"

    ##Last choice if statement.

        show pm at right

    ##Shows Yuri at the right.

        yr "I never met you till the living room encounter!"

    ##Yuri denying it.

        pov "Oh your right..."

    ##The main character slowly agrees and diagrees with his own owrds.

        show cha

    ##Shows Alphonse.

        f "Aren't you randomly trying to get an alibi! Give it up your the killer!"

    ##Alphonse suspects the main character.

        call counter_question from _call_counter_question_1

    ##Goes back to the question.

    return

    ## This ends the game.
```

Samig Sherchan

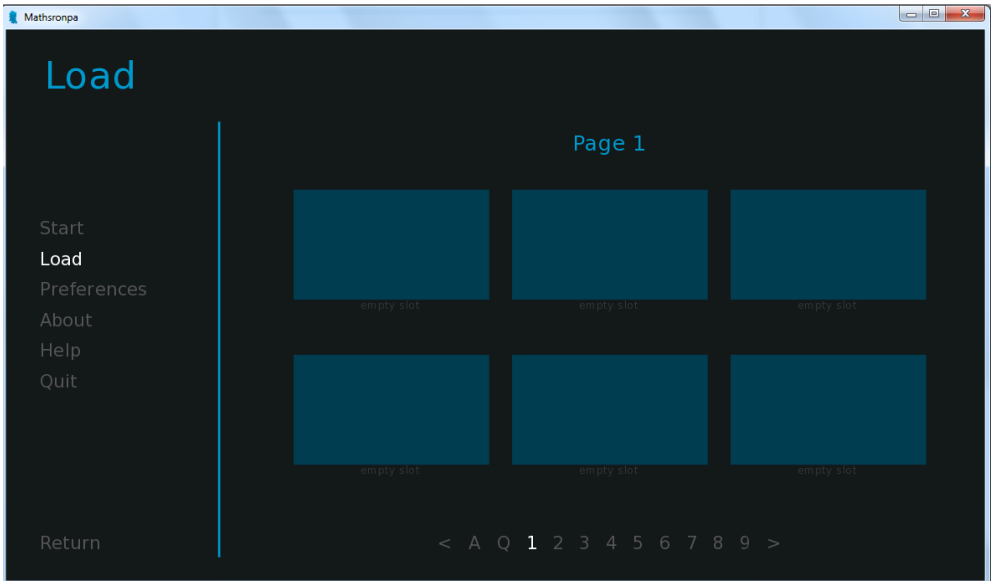
The GUI script and Screen script are prebuilt so the code is reusable, it can edited any time, it also helps the creation of the game easier. GUI script helps create the interface where the programmer and the user can communicate. It also gives the basic features of regular visual novel. Meanwhile the screens script helps display the output in the window and the screen, without displaying the code and just displaying whats required like the image of character. The Game script is mostly dialogues which is said by the characters in the game alongside it include ingame menus which are ingame choices which the user can choose, depending on what they choose they follow a unique root. Most of the in-game menu are called so they iterate giving the users chances, or the access to view the choice again for more information. This gives them the better decision making idea. Furthermore, NVL mode is a type of presentation that Visual Novels use. The purpose of NVL is to present multiple lines screen at a time in a window that takes all of the screens. Its makes effiecent use of the memory and uses less screens at a time. NVL also helped defined the layout like the color of an object or border making it look attractive. Furthermore, the IF Statement in the game scripts had to be labled and defined as well, so when using the IF Statement the word with speech mark is printed as one of the choices, depending on the choices you will go to the particular statements and continue. Additionally, the code also included lots of spacing and paddings ensuring that no frames and borders overlap which would've affected the layer and cause problems to the game. All of the scripts contained some Ren'Py statements especially because it was required to access certain parts of the code. It was required to create some features and buttons that the users can click and interact to, for example the preferences as different users have different preferences. Also images and scene required those statements to display alongside when the code is executing. This also allowed the images to adjust and position where they are needed. Most objects and variables we're set in default properties because this makes the creation of game easy and more effiecent giving a better enviroment to code the game. Furthermore, the option script included many buttons that the user will require for help, to save files or to load files. They'll never lose their progress during the game, plus the access to the option is available in the game and in the menu as well. Finally the flow chart also demonstrates the flow of data and what choices can lead, most choices are iterating depending on the situation. If the first question is No then you require to answer the refute question which also iterates giving the user chances. Then after answering it, it goes to the 2nd question. However the second choice has two choice. The users only get one chance, based on the choice they made they will either win or lose the game. Losing the game will result a Game Over screen, while winning will just show other characters happiness and joy for the win.

Samig Sherchan

After developing the game and the code, I had to test the game on whether the code was working or not. Of course at first the code won't completely work it will have errors along the way. I had to conduct some form of testing, so I did alpha testing, it was likely to have bugs. It manage to work as it displayed the image the menu with buttons, which was good so far. It also include the name of the game on bottom left with the version number.

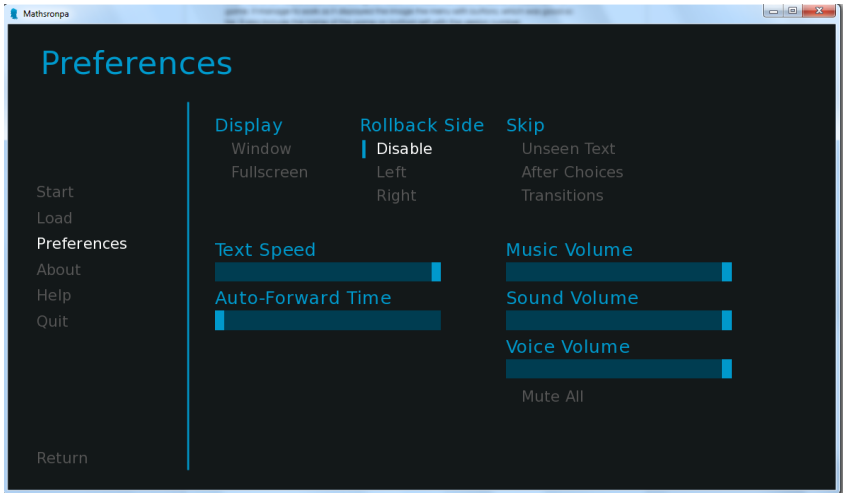


Furthermore, the game window also was in the size of what was required so it was 1280x720 which exactly what was needed for the game to work. The other need to be checked is ensuring all the options was working. So I start from Load.

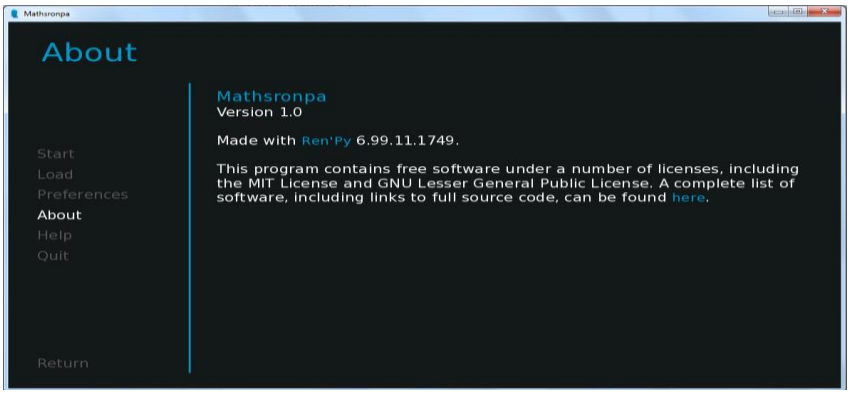


The load option was also working however it was empty since I haven't saved any progress of the game or yet to start. However, it includes all the pages that the users could save the file. Alll the file slots can be accessed anytime.

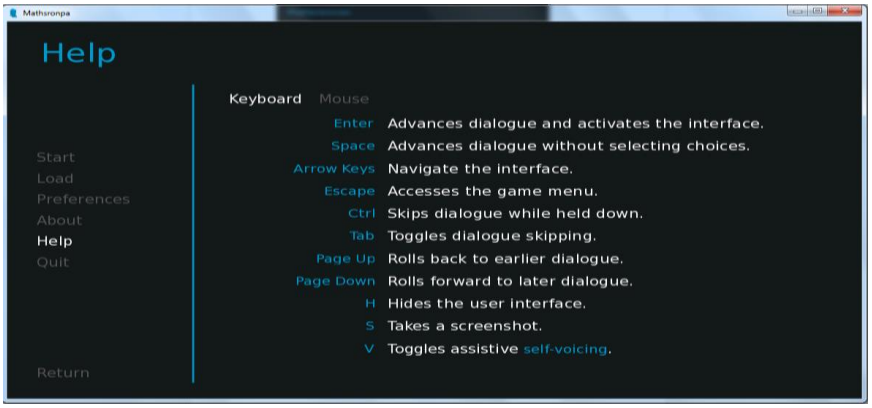
Samig Sherchan



Furthermore, the preferences button also worked, it also allowed me to select options and adjust the text speed, and adjust the audio plus I can also use mute system. So it was working fine.

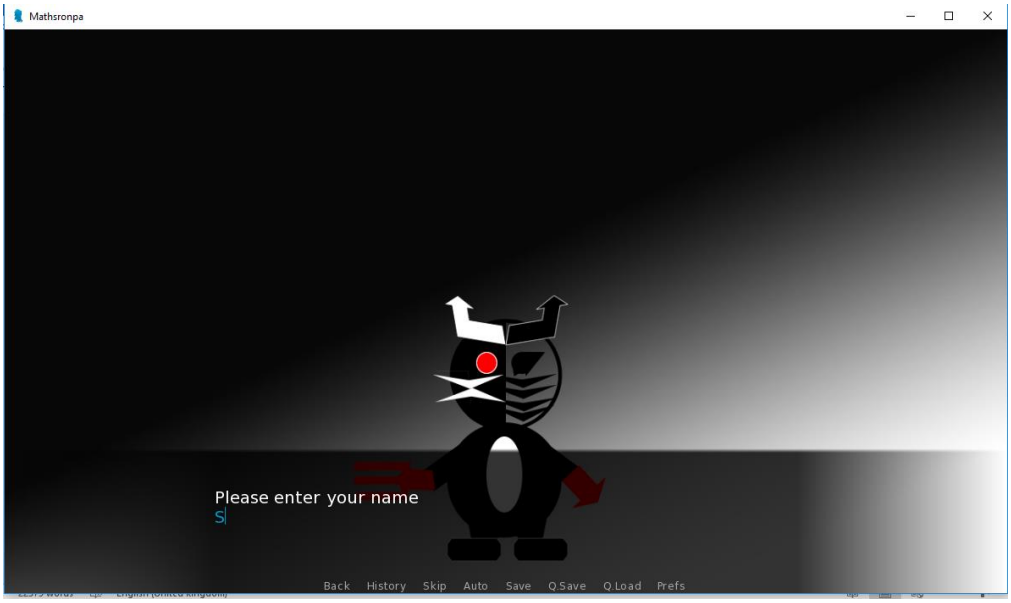


The About option contains the license for the game, plus the includes the source codes, this isn't really improtant for the user.

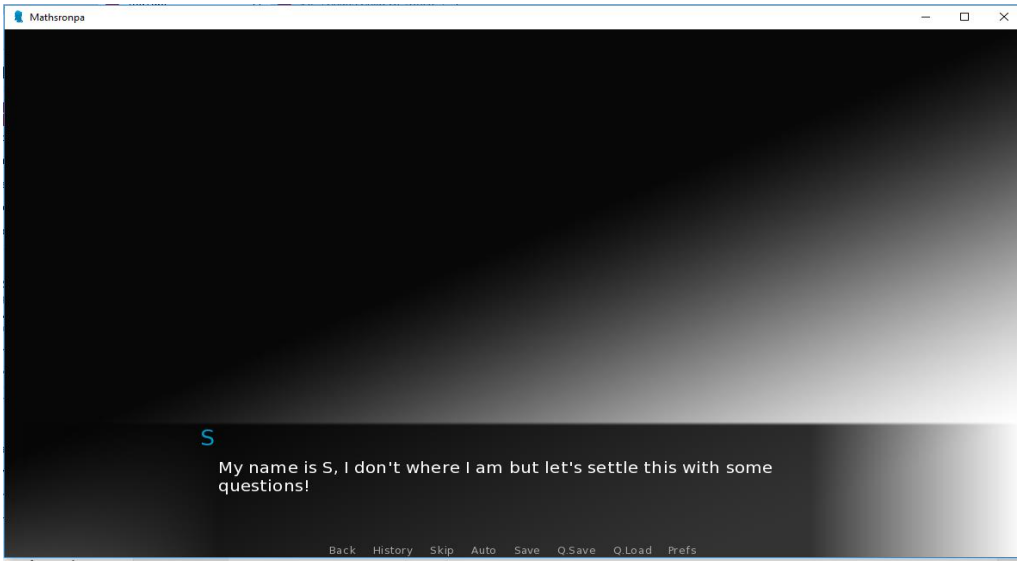


The help option is also working plus it informs the user all about the keyboard and mouse controls.

Samig Sherchan

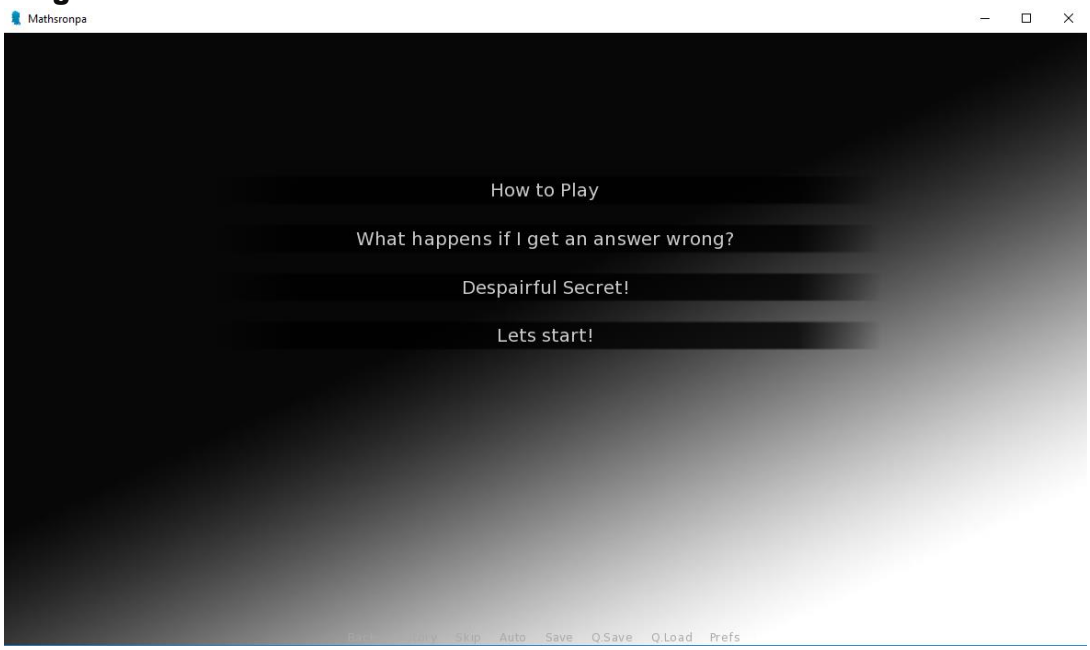


Another thing I wanted to test was if the input statement was working, this input parameter will allow the user to write their name, and the input name will be used throughout the game where ever the statement involves the character's name. Furthermore, one of the testing requirement was also if the image is alongside the text, based on the image it seems to have worked.

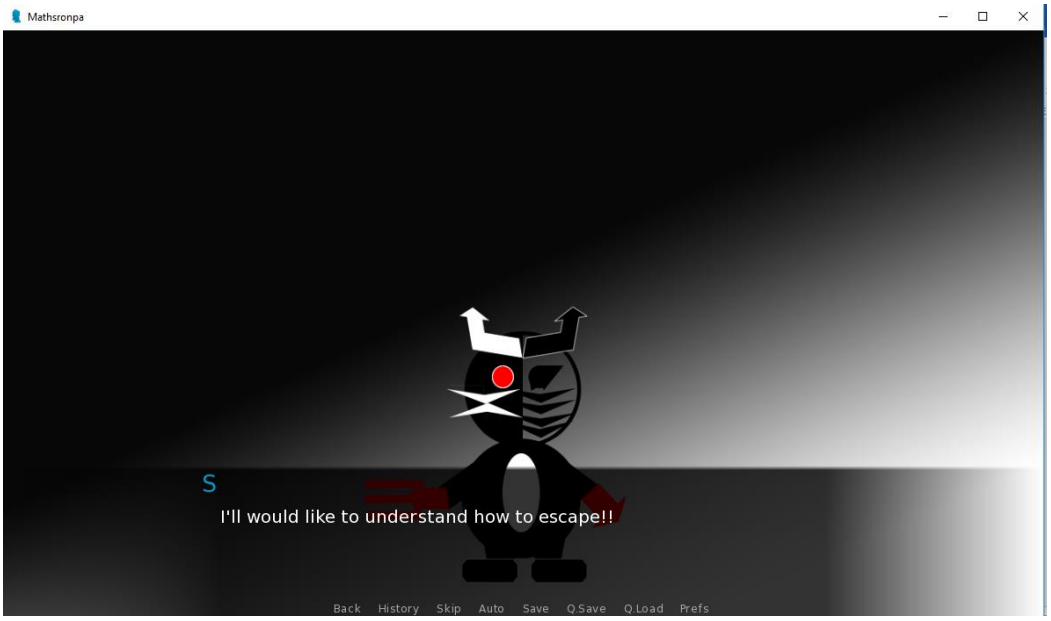


To confirm it was working, I wrote a letter as a name, there is no limit for the name. Furthermore, the next page displayed the input name where it was required so it seem it was working.

Samig Sherchan



Here is one of the in-game menu or choices which the user have to select the purpose of this is making sure the first three choice iterates after finishing each choices. However, the last choice allows the user to go the next scenario.

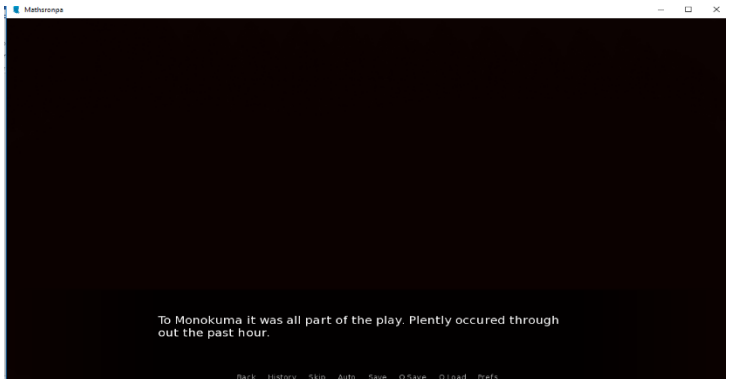


So when choosing the first three choices, dialogue will show up, and it must iterate and go back to the 4 choices. Which successfully worked. Although at first it never iterated because I didn't include the call function, call function is used to recall a menu and use it again continiously.

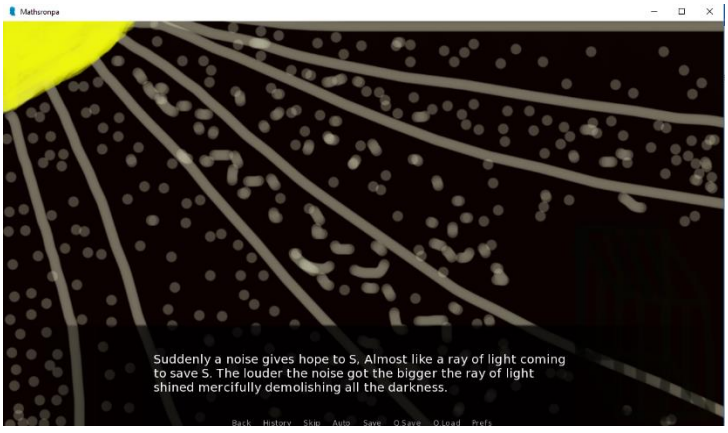
```
58  call start_from from _call_start_from
```

Samig Sherchan

This is what happens when choosing the last option it heads to the next scenario which is exactly what I want, so the choices work perfectly.

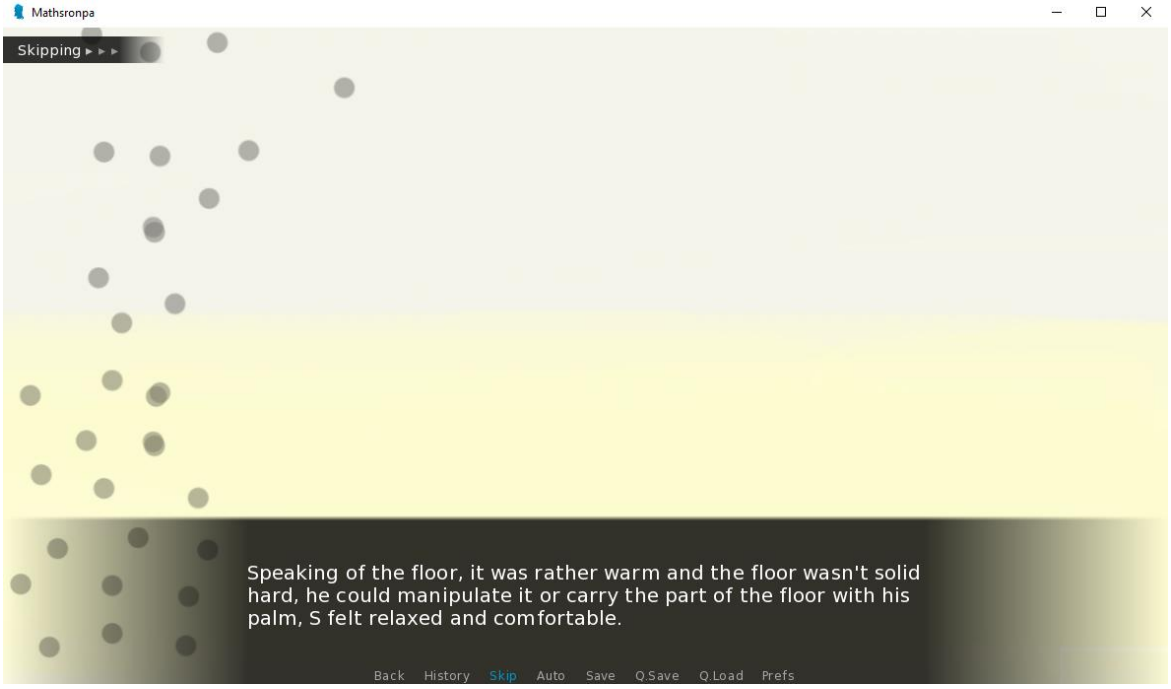


Another testing requirement was that the game changing scenes as dialogue lead the way. This is the first scene, slowly as dialogue reads it the scene must change. The second image shows the scene change after couple of pages.



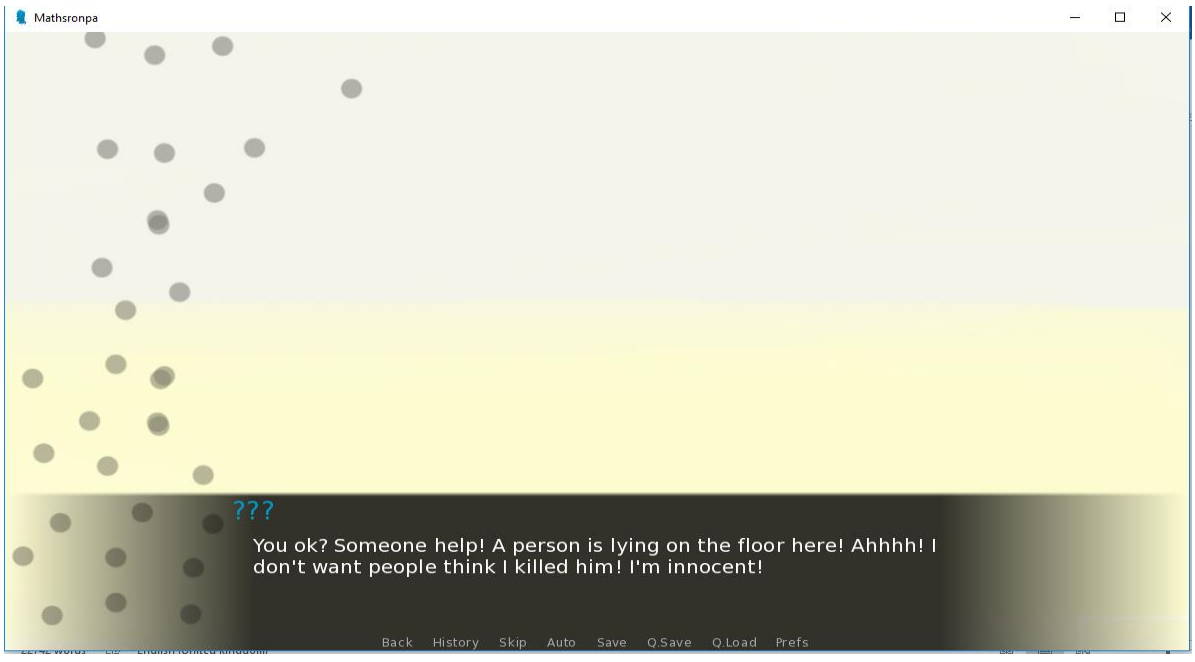
This is the second scene relating to the scenario. The aim of the background was to show the change and contrast on the character's conciousness. So this test requirment was working as well.

Samig Sherchan



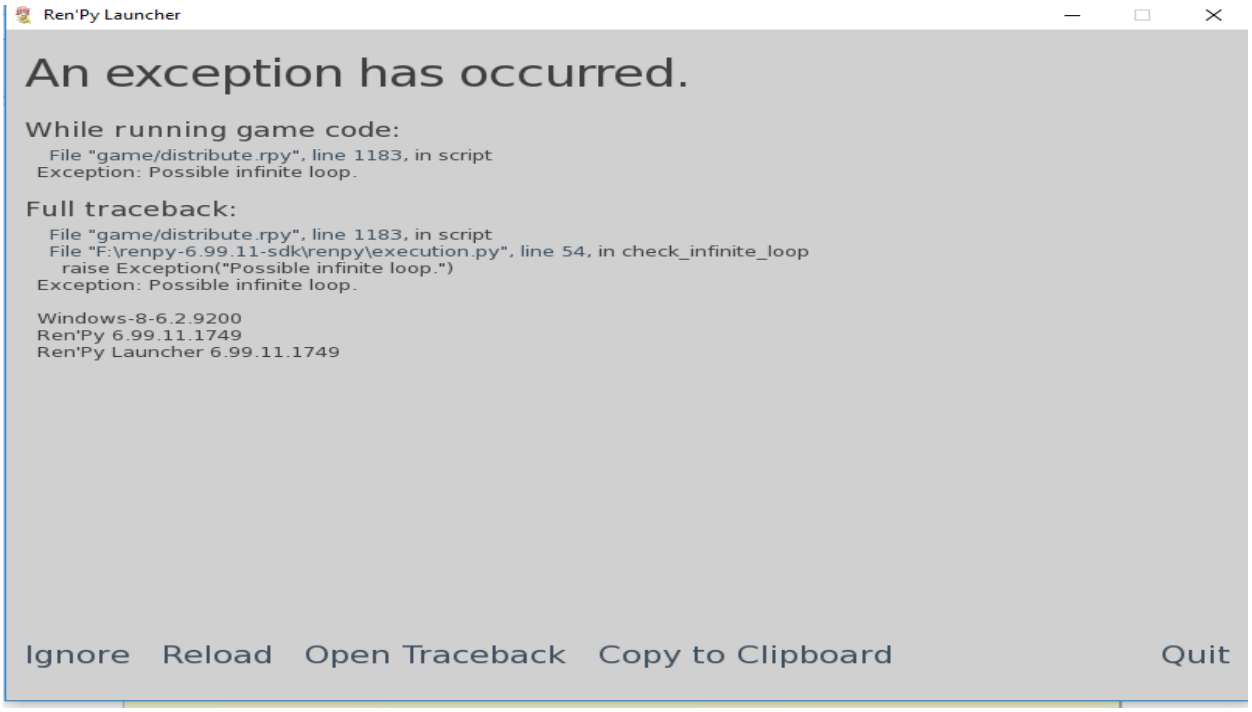
Another test requirement was to see if the buttons, keys and mouse were working, for example checking if the skip button skips the dialogues. Which it did really fast, also saving the file worked as well. The buttons also responded to the mouse clicks, otherwise you wouldn't even have been able to use the skip button. Furthermore, the key must also work.

So when I press the 'Enter' Key. It moves on to the next page.

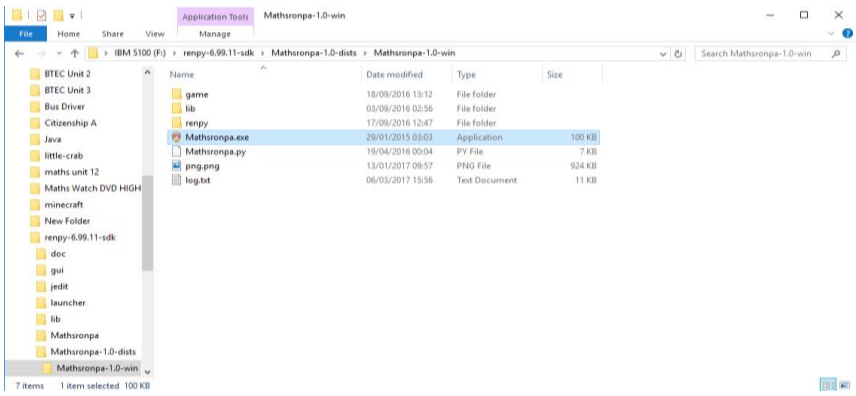


Samig Sherchan

However, trying to export the game or distribute the game, it had list of errors which required to be fixed.




```
if time.time() > il_time:
    il_time = time.time() + 60
    raise Exception("Possible infinite loop.")
```



All I require to do was remove the exception of possible infinite loop as this affected my distribution at first. Causing errors. While removing made it successful. Soon distributing it I had Mathsronpa and other components. The application ran successfully. Now I can also do a beta-testing for the game, so recalled the students I interviewed to test the game, and give their feedback and whether it's suitable. This will really help me with the user evaluation.

Samig Sherchan

Teacher Comments:

• Provided evidence of each stage of the iterative development process for a coded solution related this to the breakdown of the problem from the analysis stage and explained what they did at each stage.

• Provided evidence of some prototype versions of their solution.

• The solution is modular in nature.

• Code is annotated which explained all key components.

• Most variables and structures appropriately named.

• There is evidence of validation for most key elements of the solution.

• The development showed review at most key stages in the process

Teacher Comments:

• Provided evidence of testing at most stages of the iterative development process.

• Provided evidence of some failed tests and the remedial actions taken with some explanation of the actions taken.

Samig Sherchan

AO 3.3 EVALUATION (MAXIMUM 20 MARKS)

After testing, comes evaluation where you can evaluate and analyse the game. The game took some time to create, with list of errors and problems with in-game menu. The game creation was fairly successful, didn't really cause too much errors and problems but there are many things the game could've included. First of all looking back at the test requirments all of it was successful made.

Test No	Description	Input Data	Expected Outcome.	Actual
1.	Is the window 1280x720?	Start Game	1280x720	1280x720
2.	Does the game scenes change and work?	Start Game	When running the game must change scene where its asked too.	The game scene changes when asked to.
3.	Does the image of character appear alongside the dialogue?	Start game	The images must show alongside the dialogue.	Yes, the image of the character does appear with the dialogue side by side.
4.	Does the in-game menu allows the user to choose options?	Start game, and mouse click.	The user can choose the choices and it reads the statements.	Yes the user can choose any options they like.
5.	Does the game allow iteration after finishing each choices?	Start game and mouse click?	Some of the in-game menu must iterate.	After reading each choices, the ones to iterate, iterates.
6.	Does the buttons and keyboard plus mouse	Start game, keyboard and	The buttons must work as well as the controlls.	All buttons and key work. The mouse perfectly

	controll work?	mouse click.		communicates with the buttons.

To continue to the user evaluations I asked again to the people who contributed to my survey to test the game, and give brief description on the result of the game. Plus their own comments towards the game, as this will help understand what went wrong and possible elements I could've included. Here are the question I asked, with the user's reply and comments.

Question	Answer	Comments
Was the screen size suitable?	Yes (7)	Keith: The screen size is not bad, the school computer can run it easily. Abdi: I think its good, because it's better if the screen is big and people can see it.
Was the menu clear and easy to use?	Yes(7)	Chisa: It was actually easier that I thought, your given options and you can return and go through it again. Ciel: Simple and easy to use, you don't even need to learn some random things to use it.
Was the Maths question used suitable?	Yes(4) No (1) Maybe (2)	Kate: I kinda dislike the fact the answer is already given so you can't answer the question but you can only prove it. John: I think it was not bad all you really need to do is just say yes or no.

Samig Sherchan

Was the speed of the game flow perfect?	Yes (6) Maybe (1)	
What could made the game better?	N/A It ask for the user's opinion.	Most of the users stated that music and audio noises would've made the game better, since there is audio preferences but the game has no audio. But that is what I would like to improve on. Some users said make the dialogue and story short and ask just maths instead.

The next thing I am going to talk about is the result of the game, the good points and the bad points and even improvement.

Good Points:

- The game exported/distrubuted successfully. As the launcher wouldn't work if the error was ignored. Plus some files might've corrupt and lots of bugs and errors would've happened.
- The chosen choices iterated after finishing the choices, as the users we're required to read throught each choices and they don't miss any except for one or two choices which leads to different routes.
- All the buttons we're working fine, the users we're able to navigate through pages and dialogues. They we're able to save their progress and load their progress. You can also view other options as well.
- The game also doesn't require lot of storage space also making efficent use of the memory and the CPU. So the game will run smoothly since it doesn't involve complex machine processing.
- The scenes perfectly matched the scenario, the scenes weren't irrevallant, for the the sensation of beach showed the image of the beach, for a prison showed the image of prison.

Bad Points:

- The code was really complex and had lots of recursion, with the similar statements used again and again.
- The game doesn't' include music or audio like the user evalutation stated, this kinda ruins the mood for the game. Plus the preference button for audio feelss useless since there is no audio.

Samig Sherchan

- The game includes more story and dialogue than maths question which is doesn't really make up a good solution.
- The background design could've better and designed better, since the coloring isn't that good, it has amatuerish designs it could've have been better.

Maintaining the code is rather easy, as all I require to do is open up the game code and edit the source code. For example, if I wanted to change and correct my spelling mistakes, or add more to the code like for the in-game menu. The codes can be reused anytime, so it won't really be causing problems. Furthermore, the function of this game isn't long so most of the codes fit in one pages, saving other pages. However, the maintainability of the code isn't perfect as lots of code or reoccurs. Furthermore, the user can also have maintainability while playing this game, as they can maintain their preferences making it suitable for the users to play the game. For example, if the user had eyesight problems they could just use full screen to make the text appear a bit larger than than the windowed mode.

There is a lot of improvements that could've been made for this game. If I had more knowledge and time I could've added copyright-free music or create my own music using softwares that include instrumental audio. Since music will help make the game more fun and comfortable to play. It also creates a good atmosphere as the user uses nearly all the senses. It would also be better if I added more question than questions the user about maths, rather just validating a characters answer and having a long dialogue about the story.

Samig Sherchan

Teacher Comments:

• Provided annotated evidence of post development testing for function.

• Provided annotated evidence for usability testing.

Teacher Comments:

• Used the test evidence to cross reference with the success criteria, evaluated the solution which identified whether the criteria have been met, partially met or unmet.

• Provided comments on how any partially or not met criteria is addressed in further development.

• Provided evidence of the usability features.

• Considered maintenance issues and limitations of the solution.

• There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence.

Samig Sherchan

Problem Identification

Teacher Comments:

Identified the problem as students had difficulties learning A Level Maths since it is an huge jump. The style of learning will really affect the students, as that is the source of the whole problems occuring. Also found external and internal factors that really affects the students time studying A Level Maths. Factors like GCSE grades and their experience which influences A Level. This problem was amened as varaties of method on solving this problem.

Stakeholders

Teacher Comments:

Identified stakeholders and they were happy to contribute with their feedback in the pre-development stage and in evaluation stage. Gave interview question which had similar results and correlation. Question like "Which exam board do you think has the hardest question for maths?" showed the similar results.

Research the problem

Teacher Comments:

Researched multiple prexisting solutions which can be used to solve the problem, through different websites. Identified different approaches to thinking of the solutions, justified why certain solutions were suitable for the task. Described the essential features for this computational solution with explained choices for each of them. Like the use of the software to create the solution, and how it benefited the creation of the solution. Explained the limitation of the proposed solutions that will likely occur during the development of the solutions and pre-development.

Specify the proposed solution

Teacher Comments:

Specified the requirements required and justified why it was required to prevent problems occuring for the stakeholders. Created a table to list all the information of the requirments. Identified success criteria for the the creation of the solution. Justified why it should be met and how it effect the proposed solution.

Samig Sherchan

Decompose the problem

Teachers Comments:

Broken down the problems into samller parts, from a huge problems with the scenario to much easier problem to solve. Justified decisions on stage of decompositions making sure the problems are effectively broken down.

Describe the solution

Teacher Comments:

Explained the structure of the solution and how it will be presented and work. Described parts of the solution which will include major algorithms and how they will impact how the solution will work. Described the proposed usuablitity features to be included and how it will help out make the solutions and help the stakeholders. Identified vairables to be used in the solution with a list in the table. Identified data structures of the solution and how the solution will be implemented using it.

Describe the approach to testing

Teacher Comments:

Identified the test data to be use on the iterative development with justification on type of testing that will occur on the testing stages including reasons to perform these testings for the solution.

Iterative development process

Teacher Comments:

Provided annotated evidence of each stage with iterative process in the creation of design and it uses in the solution with constant changes and improvements. Provided protypes solutions to go through each one and select the best one.

Testing to inform development

Teacher Comments:

Provided annotated evidences for each testing which justification on the testing. With the success of the testing. Provided evidence of any remedial actions taken justifying the decision made after each testings to make sure the solutions is more robust.

Samig Sherchan

Testing to inform evaluation

Teacher Comments:

Provided annoated evidence of testing solution of robustness with a success criteria if testing went good. The testing were robust and had no problems. Provided annotated evidence of usability testing with comments from each users feedbacks with an feedback interview, each users had their own sayings.

Success of the solution

Teachers Comments:

Used the test evidence from the development and post development using it to evaluate the solution which met the success criteria. With in-depth analysis on the success criteria and testings.

Describe the final product

Teacher Comments:

Provided annotation on the usability of the final product from the design describing its effectiveness and how it influences the users and assist them. With in-depth infromation about the features and how the user can make full use of it.

Maintenance and development

Teacher Comments:

Discussed the maintainability of the solution for the users so no problems are caused. Discussed the potential further development with list of them through the feedbacks of the users and to meet their needs.

References:

- Daniweb, used to find suggesstions for progamming language, 2002, https://www.daniweb.com/programming/game-development/threads/286909/best-programming-language-for-video-games, 19/08/16.
- Scratch, used to find ideas for making type of games, 2013, https://scratch.mit.edu/discuss/topic/16229/, 24/11/2016.
- Visual Novel Random Idea Generator, to get ideas for the storyboard of the game and find the easiest one that relates to the storyboard, unknown, http://chris.charabaruk.com/random/vngen/?tags=AF4CQgQVBeAIWQ, Date: 15/12/2016.
- Wikipedia, to find forms of testing to be used to test the solution, 2001, https://en.wikipedia.org/wiki/Game_testing, 04/03/2017.

Samig Sherchan