



Prize Winner

Computer Programming, Apps & Robotics Year 7-8

Sam Weavers

Adelaide Botanic High School

Journal Index

Page No.

Summary written report	1
Journal	2
Background information	3
My experiments	4
Game draft sketches	16
My Scratch programme	23
Level 1 – Finding a new host	24
Level 2 – Invading a new host.....	29
Level 3 – Multiply numbers	32
Level 4 – Lung invasion and cytokine storm!	35
Level 5 – Handing over to bacteria and the ending	42
Project reflection	43
Bibliography	44
All of the code from my game.....	45

Summary Written Report

What is the aim of my entry?

The aim of my entry is to teach people about how a virus infects someone and what is involved when it enters the body. I have tried to do this through an interactive game with informative commentary.

What is the scientific purpose of my entry?

The scientific purpose of my entry is to teach people about how a virus enters a human's body, what it does once it is in the body and factors that affect the body's immune response.

What are the potential applications of my entry?

My entry will help other children who are learning about the human body's immune system. By playing my game they will learn how:

- viruses can enter the human body and remain undetected until they have multiplied in numbers.
- a virus can over stimulate the body's immune response which can cause a lot of damage.
- how the body sometimes loses the battle and unfortunately dies.

What type of computer is required to run my entry?

My programme will work on any computer with an internet connection.

How can you play my game?

In an internet browser go to <https://scratch.mit.edu/projects/398302334/> and follow the instructions next to the game to play. Good luck!

Who helped me with my entry?

Thanks to Mum and Dad. Dad helped me to research the human body's immune system and what books or internet pages I might like to look at or read to learn more. Dad helped me carry out the experiments but I did all the measuring, observations and recording of data. Sometimes Dad sat with me when I did the programming and asked me questions but I put the code together myself.

Project reflection

My reflection about this project is near the end of my journal (see index).

Bibliography

A bibliography is included at the end of my journal (see index).

All of the code for my game

I have provided the script from behind the Scratch coding blocks I used to create my programme.

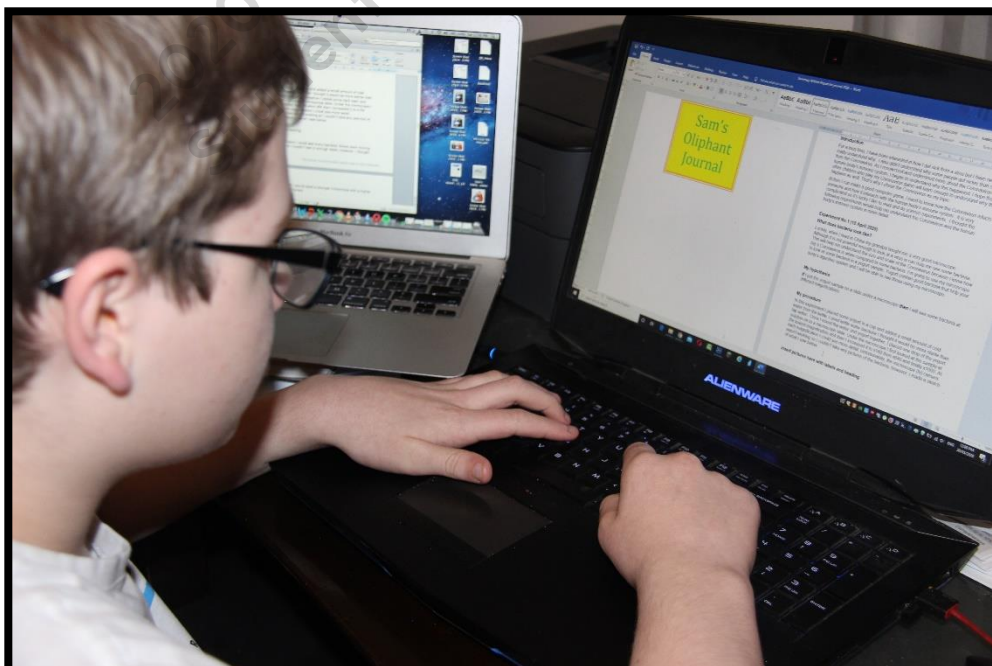
Sam's Oliphant Journal 2020

Introduction

For a long time, I have been very interested in how I get sick from a virus but I have never really understood why I got sick. I also didn't understand why some people got more sick than others from a virus. As I researched and understood more about viruses and the human body's immune system, I began to understand why this happened. I hope that other children who play my virus game will learn enough to understand why this happens as well. That's why I chose a virus as my topic.



Before I can make a good computer game, I need to know how a virus infects someone and how it interacts with the human body's immune system. It is very complicated so it's lucky that I like to read and do science experiments. I thought the following experiments would help me understand viruses and the human body's immune system in more detail.



Experiment No. 1 (10 April 2020)

How does a virus spread?

In this experiment I hope to demonstrate how a virus can spread and how it makes its way onto and inside a human body. I hope this experiment demonstrates how a virus on food might be transferred from one location to another by people and how they can use this to method to contaminate different locations and people.

My hypothesis

If over a period of one hour I take M-n-M's from a bowl that also contains glitter **then** glitter will be transferred by my hand to other surfaces.

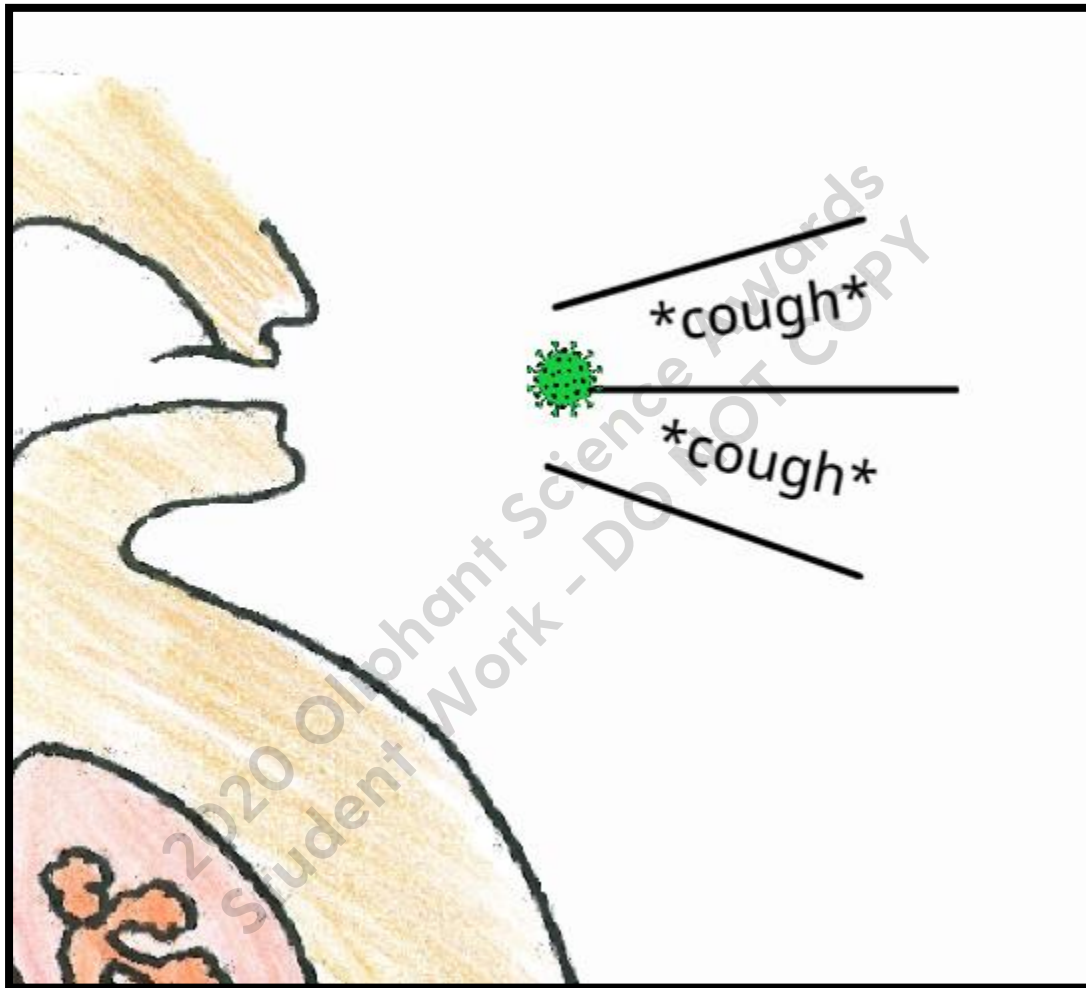
My procedure

I placed 180 grams of M-n-M's in a bowl with 20 grams of green glitter. Over one hour I ate some M-n-M's from the bowl then I looked around the house for glitter that had been transferred to other locations by my hand. I saw glitter had been transferred to door handles, light switches, kitchen bench and my face!



My reflection

The results were quite amazing. I didn't think that the glitter would be spread so far around the house (and make such a mess – sorry mum). However, if I was to do this experiment again, I would involve one or two other people to see how that affected the rate and area of virus distribution. I really liked this experiment because it demonstrated how pathogens could be easily spread from one area to another. This experiment demonstrated really well how efficiently a virus can spread and how they might infect different people.



Impact of this experiment on my game design

When I design my game, I would like to replicate how viruses are easily spread and how they use humans to distribute themselves and move from one infected person to another.

Experiment No. 2 (21 April 2020)

How do white blood cells work?

In this experiment I wanted to make a simple model to investigate how the immune system defends the human body from bacteria and viruses. In a jar I added some salt and iron filings then used a magnet to capture the iron filings. In this experiment the salt and jar simulate the human body, the iron filings simulate the pathogens and the magnet simulated the work of white blood cells in a human body immune response.

My hypothesis

If I move the magnetic wand around in the salt/iron filings mixture **then** it will capture more iron filings than a magnetic wand that is not moved around in the mixture




My procedure

First, I added one cup of salt to a jar so that it was about half full. Then I added 5 grams of iron filings to the jar.



After shaking the jar to mix the two ingredients I removed the jar lid and pushed a magnet wand into the mixture and pulled it out straight away. I weighed any iron filings that were attached to the magnet and recorded the results in the table below. I replaced the captured iron filings to the jar and shook it again. Inserting the wand again I move it around in the mixture in the jar, twisting as I moved it. I did this for 15 seconds and then again for 30 seconds. This was to simulate a white blood cell tracking down a virus in a human body. I removed the wand and weighed the iron filings attached to the wand and recorded the results in the table below.

Iron Filings experiment results table

Method	Description	Amount of iron filings captured by the magnet
1	<p>Insertion and removal without moving around the jar</p> 	1 gram
2	<p>Magnet insertion and moving around the jar for <u>15 seconds</u></p> 	2 grams
3	<p>Magnet insertion and moving around the jar for <u>30 seconds</u></p> 	4 grams

It is clear that the second and third methods of wand insertion captured more iron filings than the first static method of insertion. I think the movement of the second and third wand insertion methods better simulated the life of a white blood cell constantly searching and moving around the body searching for pathogens. This is why the second and third methods captured more iron filings than the first method.



My reflection

If I was to do this experiment again, I would use one small magnet initially, then two and three magnets. I think this would better simulate the body's immune response. The immune system uses many white blood cells to defend the body from invading pathogens so using multiple magnets would provide an interesting change to this experiment. This experiment was really interesting because it explained how white blood cells moving around the body make them more efficient at capturing pathogens.

Impact of this experiment on my game design

It was fantastic to see how the simulated white blood cells moving around the body would encounter more pathogens. When I am designing the white blood cell portion of my program, I would like to incorporate white blood cell movement and demonstrate how that is an efficient way to capture invading pathogens. A snapshot of the white blood cells from my game is below.



Experiment No.3 (01 May 2020)

Making model of human blood?

In this experiment I wanted to simulate the contents of a drop of blood. I used glucose as the nutrient rich plasma, red tic-tacs as red blood cells, small marshmallows as White blood cells and hundreds and thousands as platelets. From my research I know that blood generally consists of:

- 54% plasma
- 45% red blood cells
- <1% white blood cells
- <1% platelets



My procedure

Although it was difficult, I tried to keep the percentage of ingredients the same as a what is contained in a drop of blood.

First, I added 250 grams of glucose to a glass jar.



Then I added 48 grams of red tic tacs



Then I added 3 grams of white marshmallows.



Finally, I added 1 gram of hundreds and thousands.



I stirred the ingredients together using a teaspoon and put the lid back on before I shook the jar to mix the contents further. The pictures below show the final results.



My reflection

If I was to do build this model again, I would try to use ingredients that were not affected by the liquid. After about fifteen minutes the solids in the liquid started to discolour and this affected the look of the experiment. Also, because a virus is so small, I couldn't incorporate it to my model. Although the jar with all the ingredients in it looked great and gave me a good idea of what blood might look like if it was magnified.

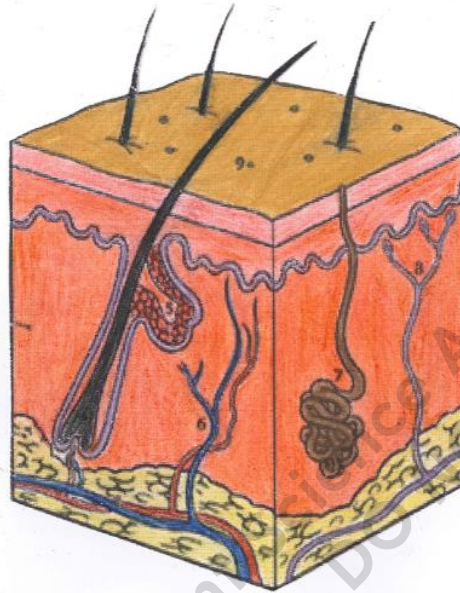
Impact of this experiment on my game design

I am very happy with how the model looked immediately after all the ingredients were added but using different ingredients would give me more time to observe the model before the liquid affected its contents. This experiment helped me better visualise the size of what is contained in a drop blood and I think this will help me design a better game.

Experiment No. 4 (16 May 2020)

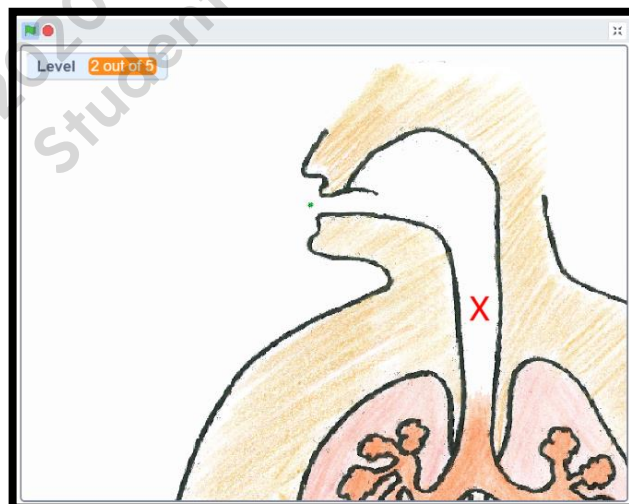
How does skin protect our body?

I read in a book called How the Human Body Works that the skin provides a terrific barrier to stop a virus entering the human body. As well as acting as a barrier it is helped in different areas of the body by combinations of sweat, oil, acid, mucous and enzymes (for example defensins on the skin). Of course, a virus can gain access to the body through the mouth, eyes, nose, etc but the skin helps to dramatically reduce the entry points. I printed a cross-section of skin from education.com/worksheets and added some colour to reinforce what I had learned about the skin.



My reflection

Although this was a simple experiment, I liked doing it because I now have a good picture in my head of the different layers and components of skin.



Impact of this experiment on my game design

Understanding the different components of the skin will help me design a program that reflects how the skin helps keep a virus out of the body. It will also help me incorporate where a virus is more likely to enter a human's body.

Experiment No.5 (17 June 2020)

What does bacteria look like?

Luckily, when I lived in China my grandpa bought me a very good microscope. Although it is not powerful enough to see a virus it can help me see some bacteria. This will help me understand the size and scale of a virus because I have read about how big a virus is when compared to some bacteria. I'm going to use my microscope to look at some bacteria in a yogurt sample. Some types of yogurt contain live bacteria that help your body's digestive system. I hope that I will be able to see those bacteria using my microscope at different magnifications.

My hypothesis

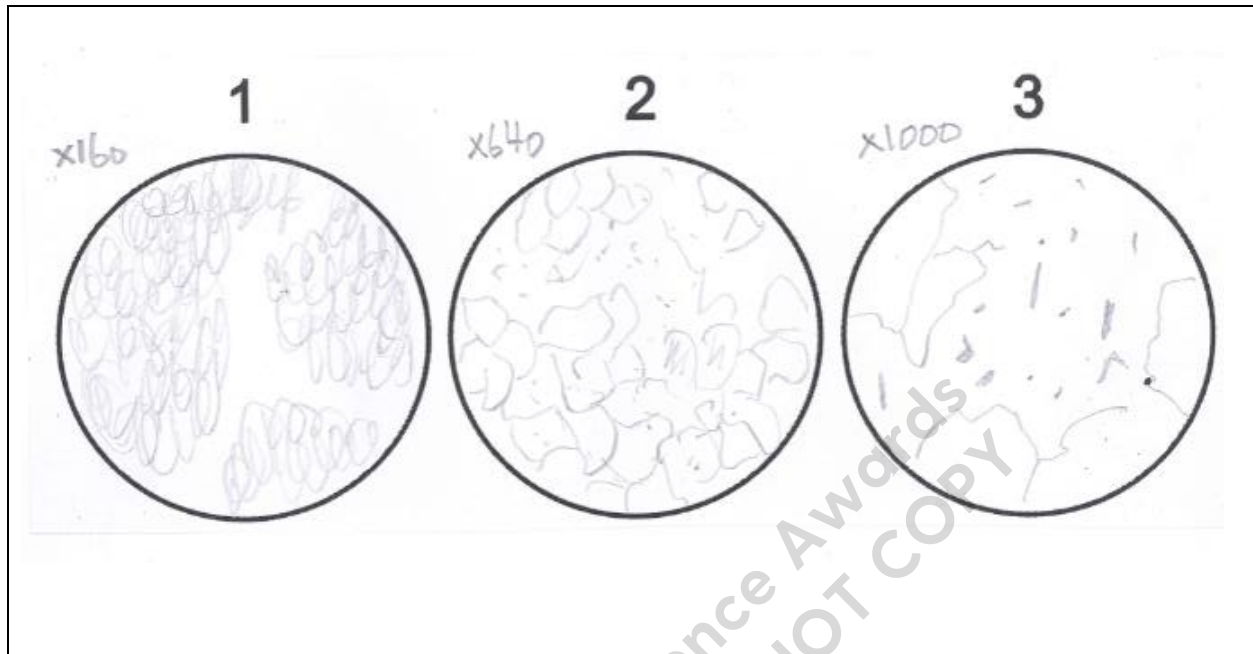
If I put the yogurt sample containing some live bacteria on a slide under a microscope **then** I will see some bacteria at different magnifications.

My procedure

In this experiment I placed some yogurt (that contained some live bacteria) in a cup and added a small amount of cold water from the kettle. I used kettle water because I thought the water would be more sterile than tap water. Once I mixed the water and yogurt together, I placed one drop of the yogurt solution on to a microscope slide. Under the microscope I first looked at the sample at the lowest magnification x64 then I increased it to x160 then x640 and finally x1000. At each magnification I could see more detail. Unfortunately, the microscope (tv) camera wasn't working so I couldn't take any pictures of the bacteria, however, I made a sketch of what I saw below.



At the highest microscope magnification, I could see many bacteria moving. I wasn't sure as I couldn't see in enough detail, however, I thought some of the bacteria may have been dead. I would need a stronger microscope with a higher magnification to determine if that was correct.



My reflection

If I was to do this experiment again, I would use a stain or I would use more than one type of stain to see if it changed the number of bacteria I could see. I would also try to look at the yogurt sample through a stronger microscope to see if that changed the detail or number of bacteria I could see. I really liked seeing the bacteria in this experiment.

Impact of this experiment on my game design

Although this experiment gave me some great information about the size and shape of the bacteria in yogurt, I couldn't see a virus. Viruses are too small. This is important because when I am designing my program, I can try to incorporate this information to make it more interesting and realistic.

Game draft sketches

Introduction

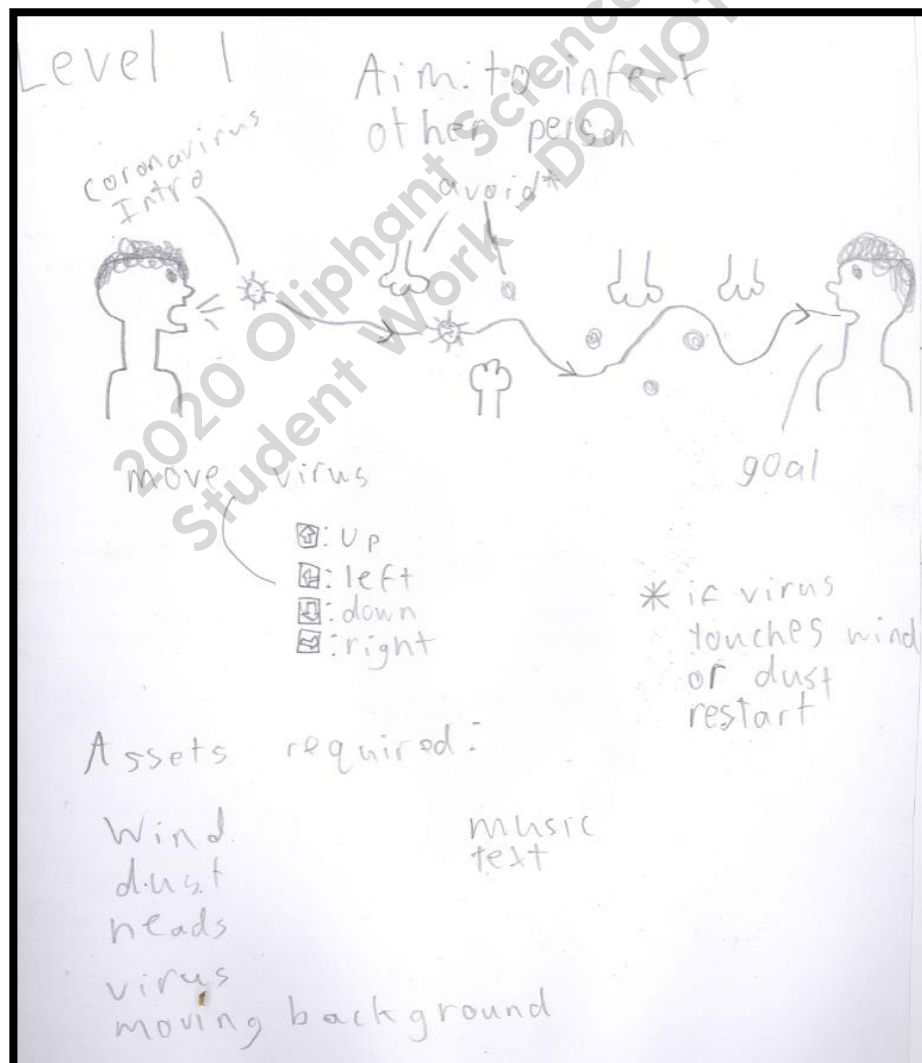
Before I begin coding and writing text for my game, I wanted to make a rough draft of what I thought each level might look like. First, I imagined what the overall game might look like in my head and broke that down into levels.

The following pages are what I drew before I started to build my game and I used as a guide during my game development. I changed a few things as I developed my game but these sketches were very effective in keeping my ideas on track and the information in the different levels separated.

Level 1

The first level my program closely reflects my initial draft thoughts. I was pleased that I was able to incorporate using the arrow keys for movement, most of the assets and most key concepts. It was quite difficult to incorporate the wind so that it had an effect on the dust/virus and not impact so much that it made it too difficult for the player. I think I found an appropriate level of interaction.

Initial draft



Actual level in my game

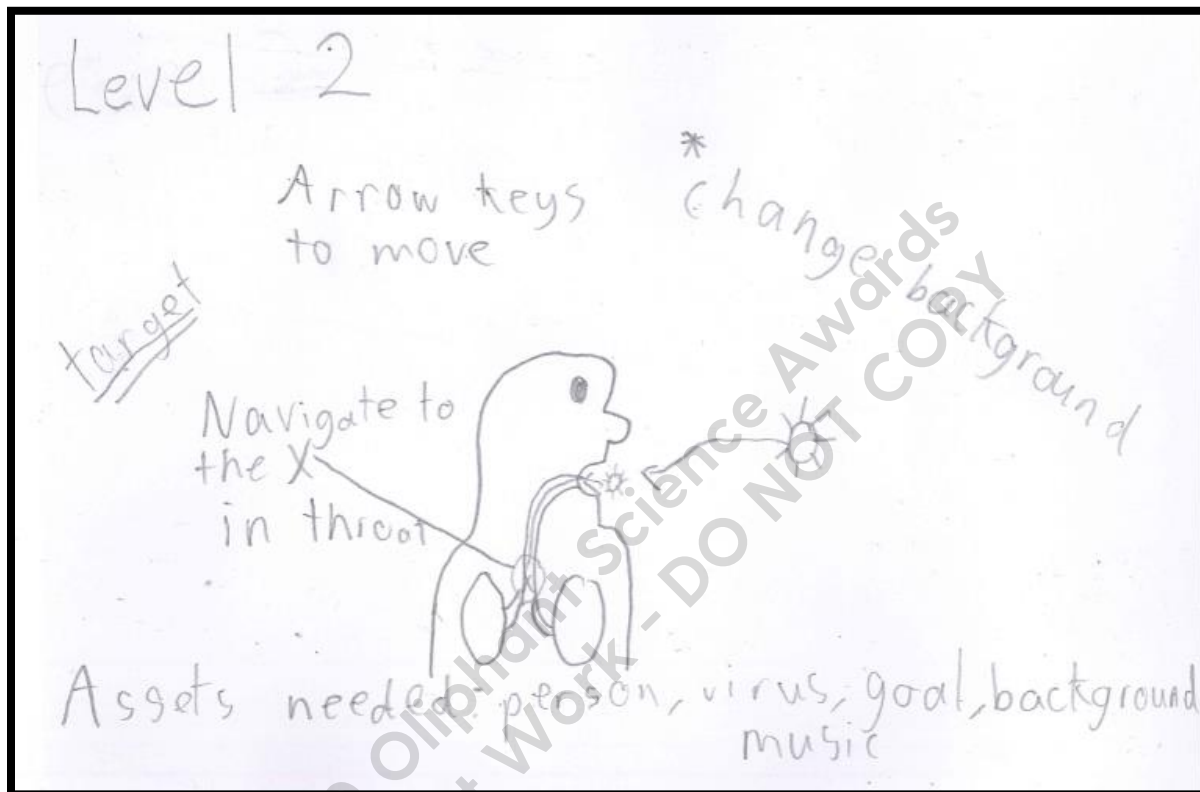


I added a moving back ground that I hadn't thought of in the early draft. When I was programming this level, it felt like it needed the moving background to help the player feel like the virus was moving from the infected person to someone else. I think it also made this level feel more interesting.

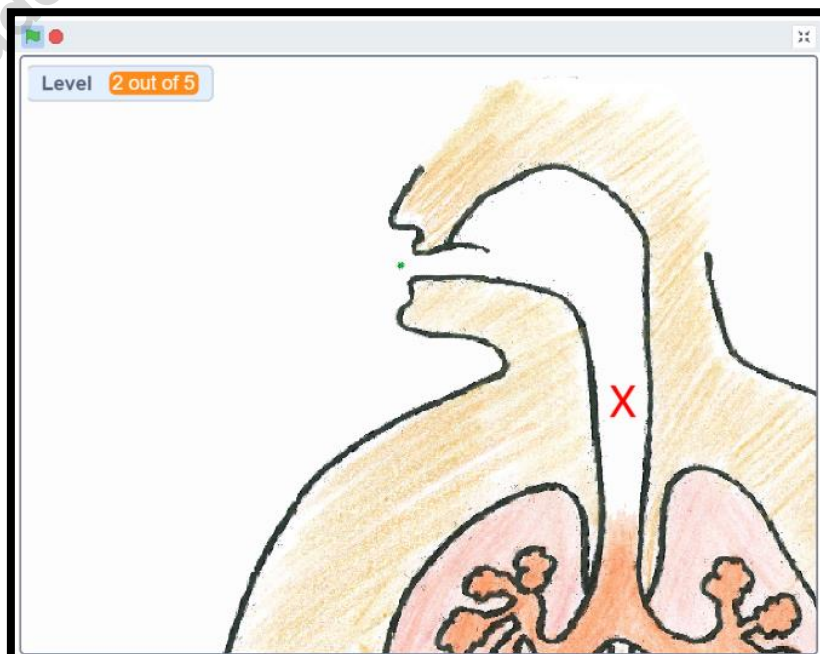
Level 2

In my draft I wanted this level to simply reflect the need for the virus to make it to a part of the body where it can safely invade a host cell and replicate. When programming this level, I decided to shrink the virus so it was easier to show the navigation from outside the body to the throat area. It also made it easier to use the navigation arrows to maneuver the virus to the throat area (marked by an 'X'). This is a simple level but contains a really important message for the player. I'm really pleased that I added this level and the transition between the first level and this one is really smooth.

Initial draft



Actual Level in my game

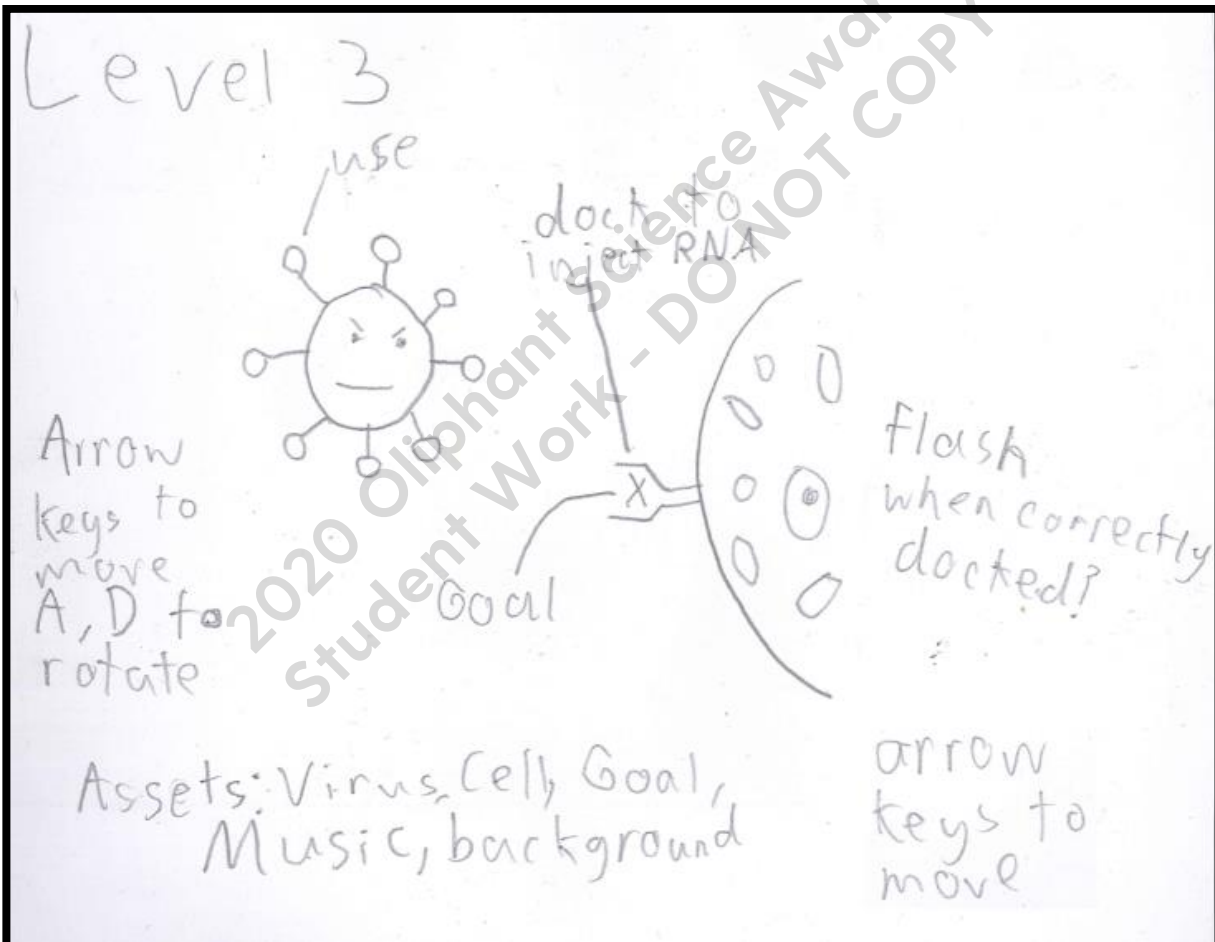


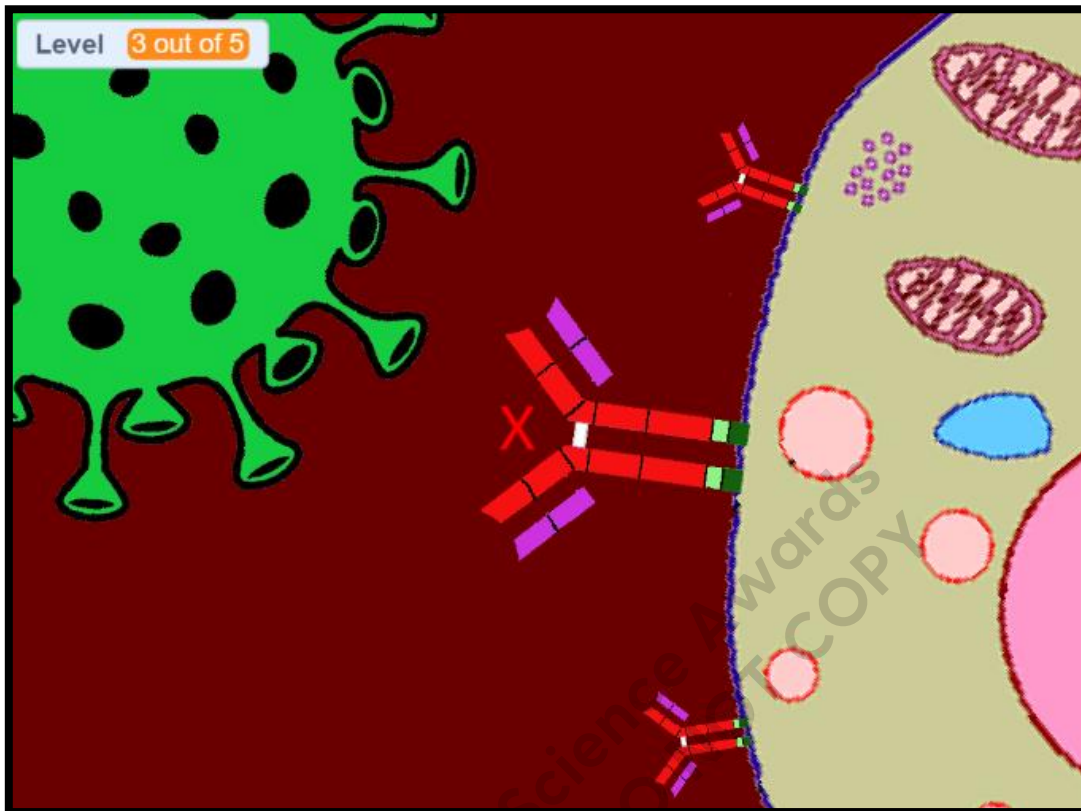
Level 3

When I was designing this level, I wanted to show how a virus must inject its RNA into a cell so that it can be replicated. The way that a virus can do this is by binding its protein spike with a receptor on the cell surface tricking the cell so that it thinks that it is ok to bind with. Once this has been achieved the virus injects its RNA in to the cell and the replicating process begins. Making this clear in a game was quite challenging. I settled on a simulation where the player has to maneuver the virus so that one of the virus' spike's docks with the cell receptor.

Programming the receptor so that it will only dock in the correct location was difficult but the end result was worth the trouble. I think this level clearly shows the binding process and the RNA being inserted. I also made the cell flash when the RNA was inserted to reinforce to the player that this step had been completed correctly.

Initial draft

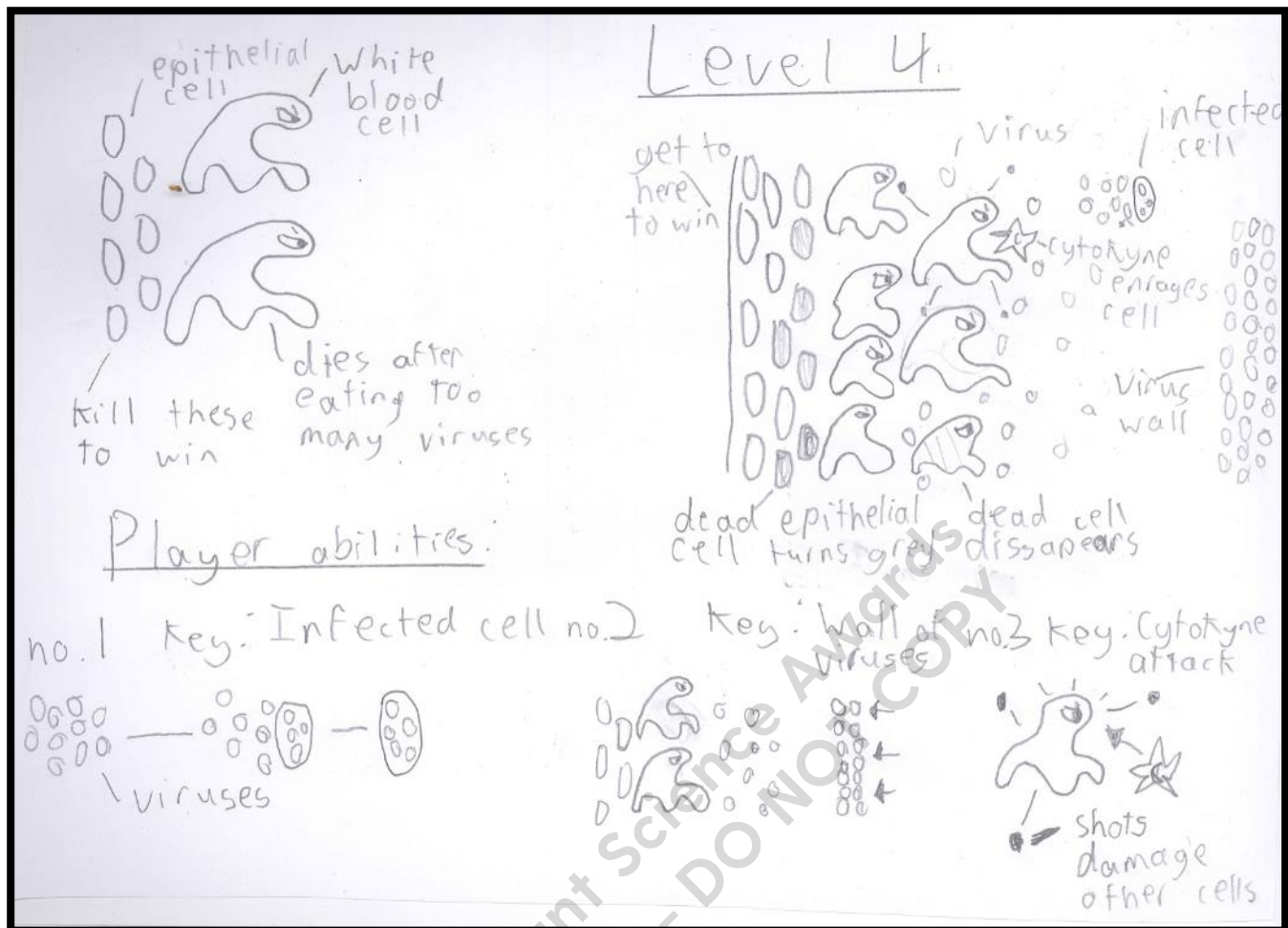




Level 4

In my initial draft outline, I was focused on the overall design of this level more than the mechanics. I knew that I wanted to simulate the virus trying to infect and damage the lung wall cells but I also wanted to make sure that the player understood how this was multifaceted. There are many processes happening at once during this level and I wanted to place the player in charge of as many as I could. I settled on providing them with three processes, placing an infected cell bursting and adding virus numbers to the area, an intensive wave of the virus that happens from time to time and white blood cells interaction with cytokine chemical signals. I also added a cool down timer to these functions so that the player would need to consider their strategy when deploying them.

I gave the white blood cells (that prevent the virus from reaching the lung cells) a terminal limit so that after '#' number of virus touch them (ie they ingested '#' number of virus) they would die. Of course, the human body has millions of white blood cells so they are soon replaced. I thought this added a nice element to this level and I am really happy with how it ended up.



Actual level in my game



Level 5

At the end of the game I wanted a way to wrap up the game and let the player know what happened. I used the virus again to let the player know that the virus had been defeated but as in many serious virus infections the body is left very weak and with a lowered immune system. This is dangerous because the bacteria that would normally be easily controlled by the human's immune system can no longer be kept under control. In this situation bacterial pneumonia often develops which requires a lot of medical intervention and is a very serious problem. However, because there are so many people who have been affected by Covid-19 I decided to make sure that the infected person in the game survived.

2020 Oliphant Science Awards
Student Work - DO NOT COPY

My Scratch programme

Introduction

I am going to write a scratch game about how a virus invades a human body and weakens it so that bacteria have an opportunity to overcome the immune system. A virus is very simple, it's a protein or fatty shell containing some RNA or DNA. Some people argue that a virus isn't alive as it needs the cells of another organism to reproduce. Either way it can cause some serious problems for an infected person. I don't want to make my game too complicated because I want people of all ages to be able to play my game. Hopefully, they will also learn something about the processes involved. I have broken the game down into the following stages:

Level One Finding a new host

Level Two Invading a new host

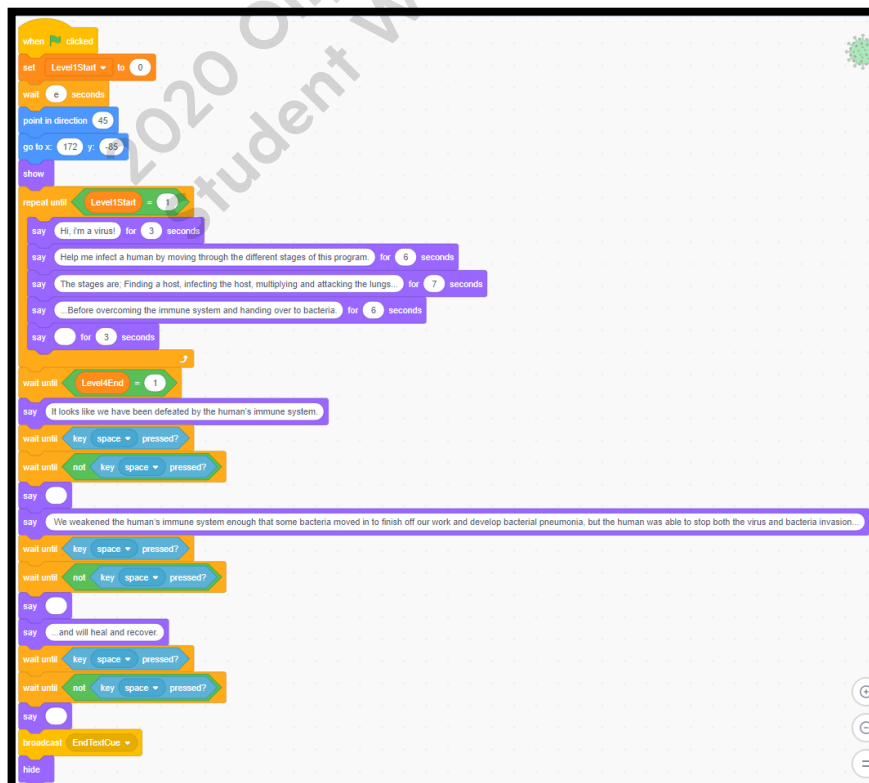
Level Three Multiply numbers

Level Four Lung invasion a cytokine storm!

Level Five Handing over to bacteria and the ending

Splash screen

I added a splash screen to the start of the game so that a player had some information about how to play the game before they started. I added a virus sprite and some text explaining how to play the game. I also added a start button. An example of the code I used is below.



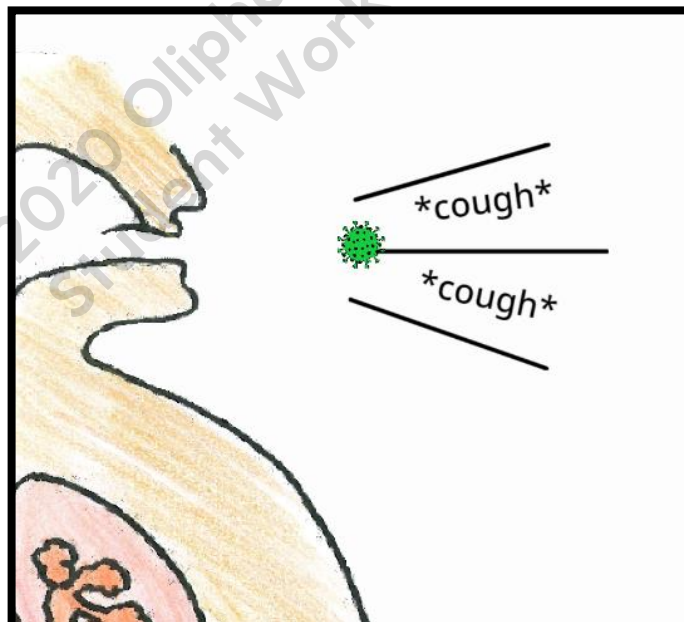
LEVEL ONE – Finding a new host

The object of this level is to move from an infected person on the left of the screen to another healthy uninfected person on the right of the screen. The virus is coughed out of an infected human and then provides the player with some more information about the game. An example of the code I used is here.

Once the virus has finished talking it is launched and begins flying through the air and this level begins.

I used a forest background to simulate movement, however, the player uses the arrow keys to navigate up, down, forward and backward to avoid obstacles. Although a virus can be spread by someone touching an infected surface and then their face, etc it can also be spread through the air as an aerosolised virus. I thought an aerosolised virus would make a more interesting game level so I chose that method of infection. To make it even more interesting I added some dust particles and wind as obstacles. The wind blows the dust at different times to change the dusts trajectory so it makes it more challenging for the player. If the player touches any obstacle the level is reset and they get a chance to try again. This forces the player to play more strategically and understand that the journey of the virus is not straight forward (this is explained more later).

Once the virus is coughed out of the infected human and settles in the air it introduces the game and provides the following information.



As the player plays this game they will learn about the immune system of the human body and how it responds to an invading virus. A virus infection is very complicated and difficult so I have enlisted the help of a virus to help explain the various stages. At

different points throughout the game the virus will appear to explain what is happening or how to play the game.

Although I have tried to keep these words to a minimum (so it doesn't interrupt the flow of the game) longer explanations are required at some key points. The intercellular

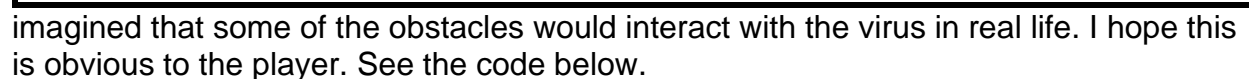


aspects of the immune response are very complicated and difficult to follow so I decided to simplify it in the game.

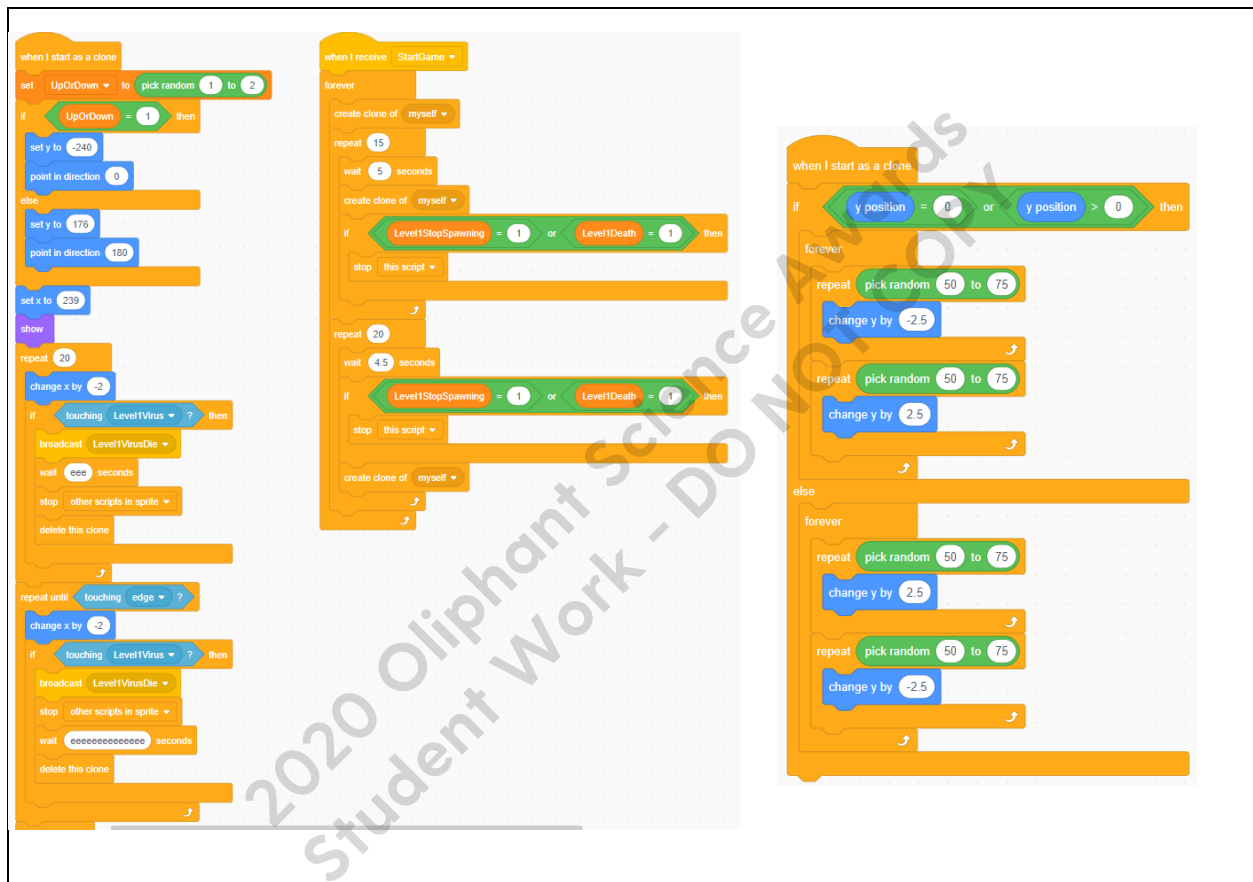
I think I have found a good balance between providing enough information so that you can learn about a virus infection but not so much that players will feel overwhelmed.

I hope you enjoy it and learn something as well. Good luck!

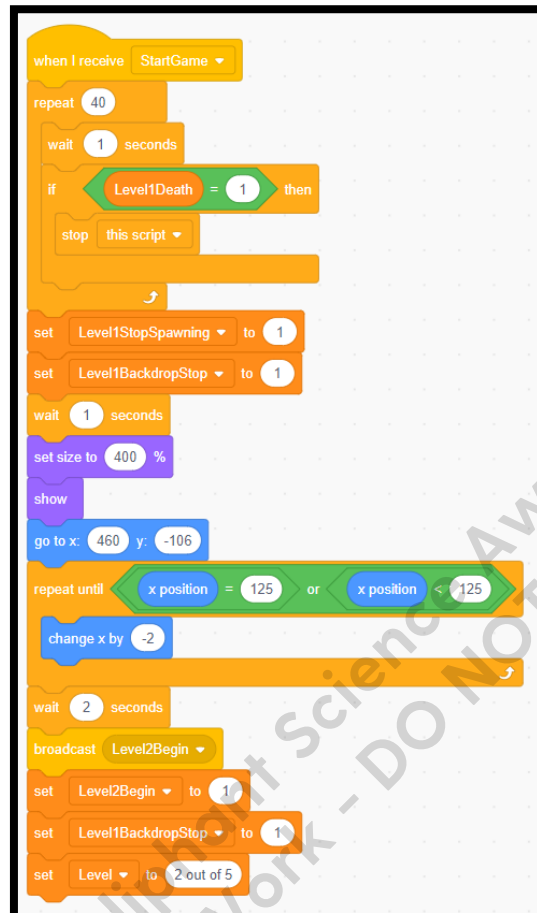
I wanted to make sure that some of the outside elements were included in this level of the game so I added a wind sprite that either blew the dust in a different direction (if the dust touched the wind sprite) or blew and killed the virus if it got too close. This is how I



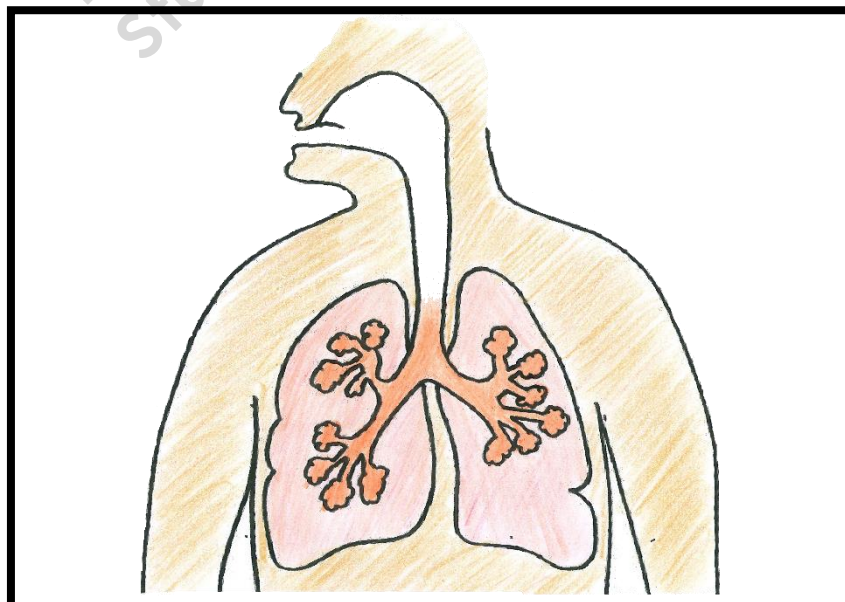
The dust obstacles were created using a sprite that I rotated and moved up and down (using y-axis code) to increase the illusion that it is moving through the air. I also wanted to make sure that the wind had an impact on the dust (just like in real life) so I added a script that pushed the dust sprite if it touches the wind sprite (the dust sprite changes shape slightly if it is affected by the wind). There is also a script for this level that controls the death of the virus if it touches either the dust or wind sprite. See the code below.



I wanted to make sure that a player moved through all of the game levels so that they could understand the entire virus journey not just the initial stages. Once the player navigates the obstacles for 40 seconds they reach the next stage, entering a healthy human. An example of the code I used is here.



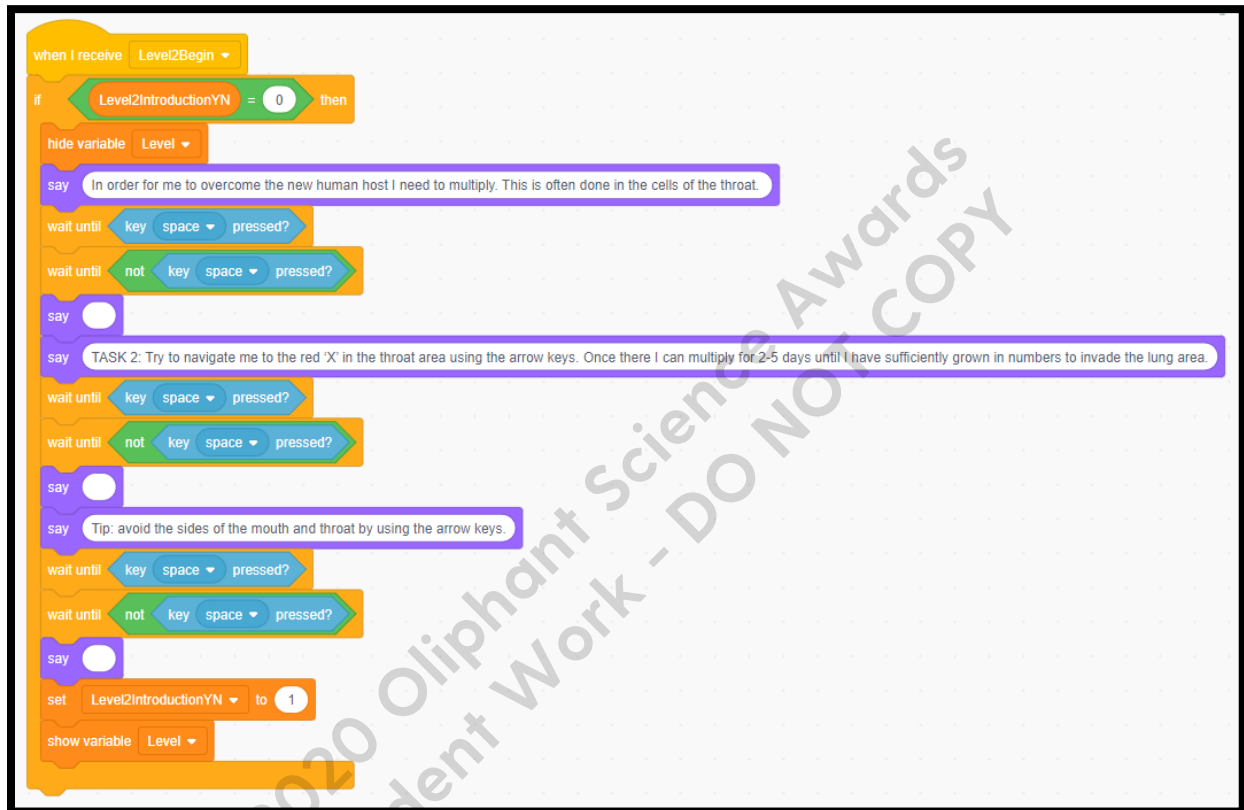
At the end of the journey the human target appears and the virus then tells the player well done they have reached the new host and that they will receive more instructions at the next level.



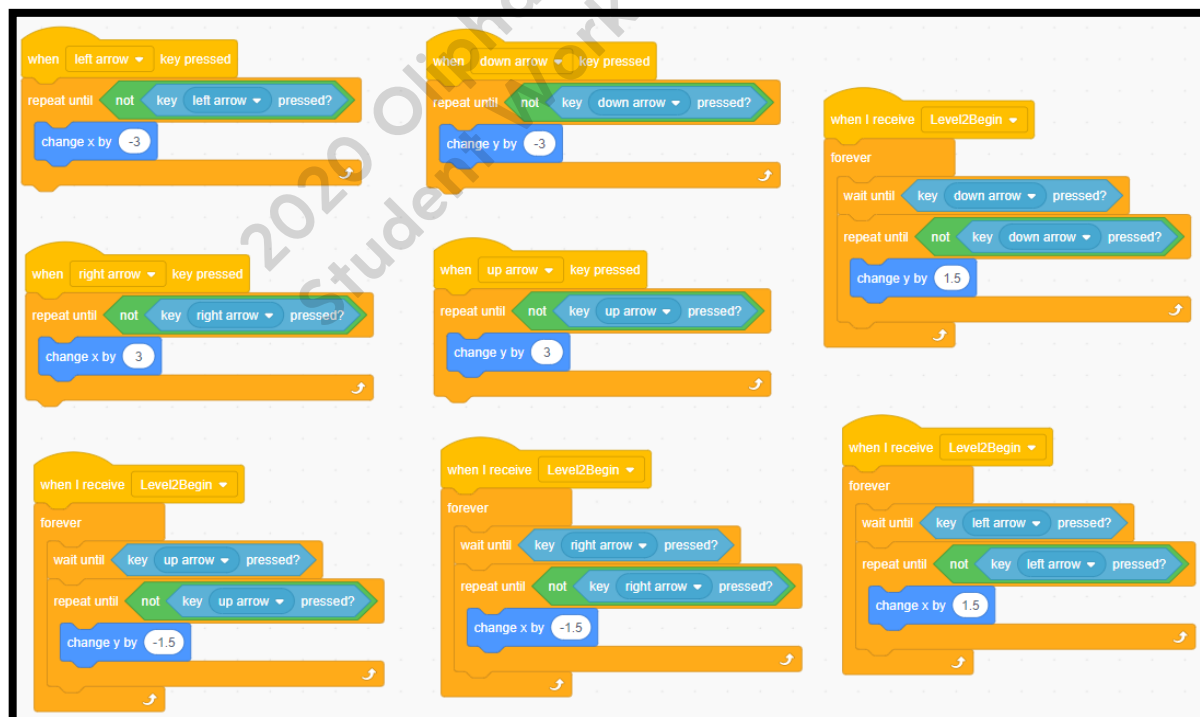
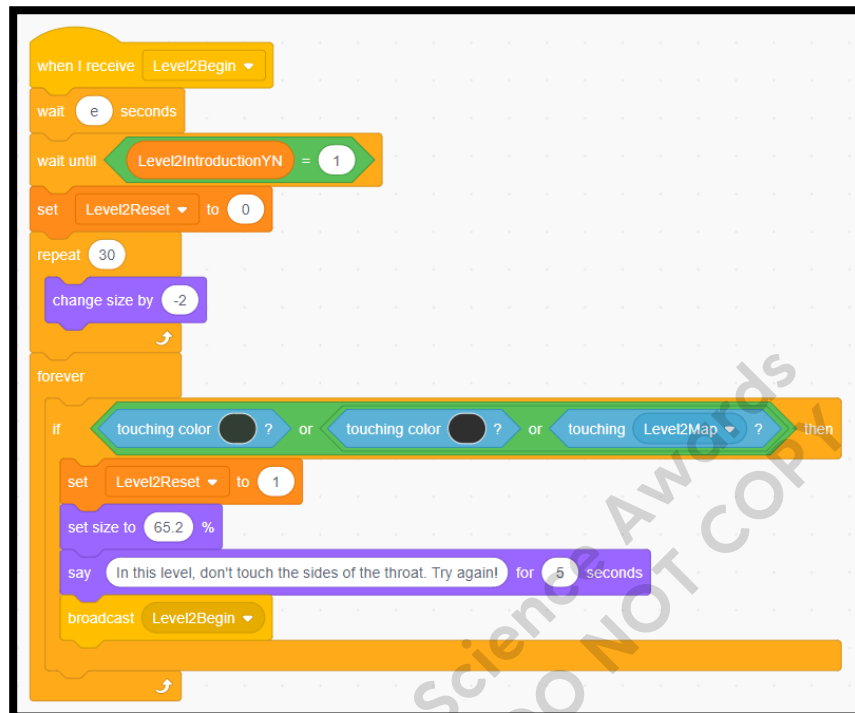
LEVEL TWO – Invading a new host

In this level of the game the virus has reached the new host and needs to make its way inside the body where it can multiply. The player is told by the virus that they need to navigate to the throat area so that it can invade an epithelial cell and start replicating for 3-5 days. Once the virus has multiplied and grown to sufficient numbers it can move on to the next level.

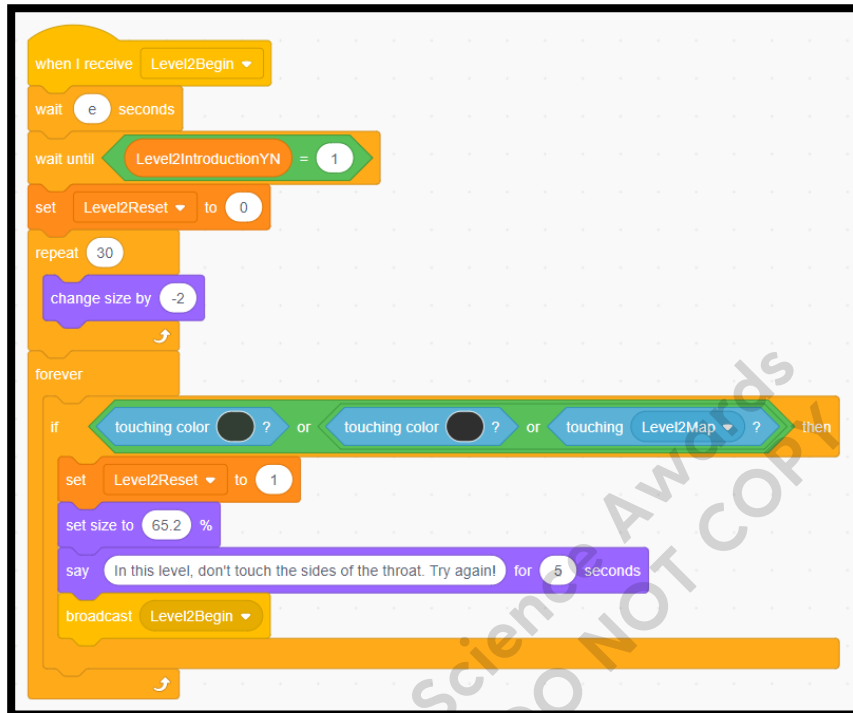
As before, the code for the virus initiates the start of the game once it has finished speaking. See the code below.



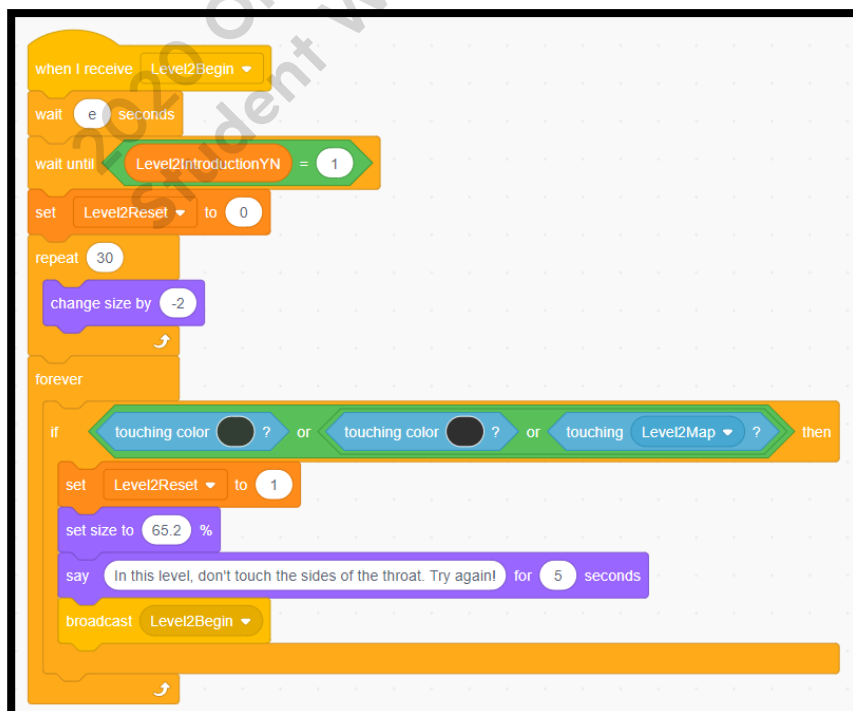
In order for the virus to enter the body I added a piece of code that reduced the virus' size to 5.2. The player uses the arrow keys to navigate in to the mouth and down to an 'X' marked in the throat area. This is the target area and where viruses sometimes start their invasion of the body. I also made the 'X' flash so that the target area was more obvious to the player. Once the player reaches the 'X' a sound is played indicating they have completed the level.



I made the walls of the mouth and throat area sensitive to touch so that if the player touches the virus on the walls of the mouth and throat before reaching the marked 'X' area the level restarts. I did this so that the player understood the area that a virus usually infects first and they realised that if the virus touches another area it would probably not survive.



If the player touches any part of the body before reaching the 'X' the virus gives further instructions on how to complete this level and restarts.



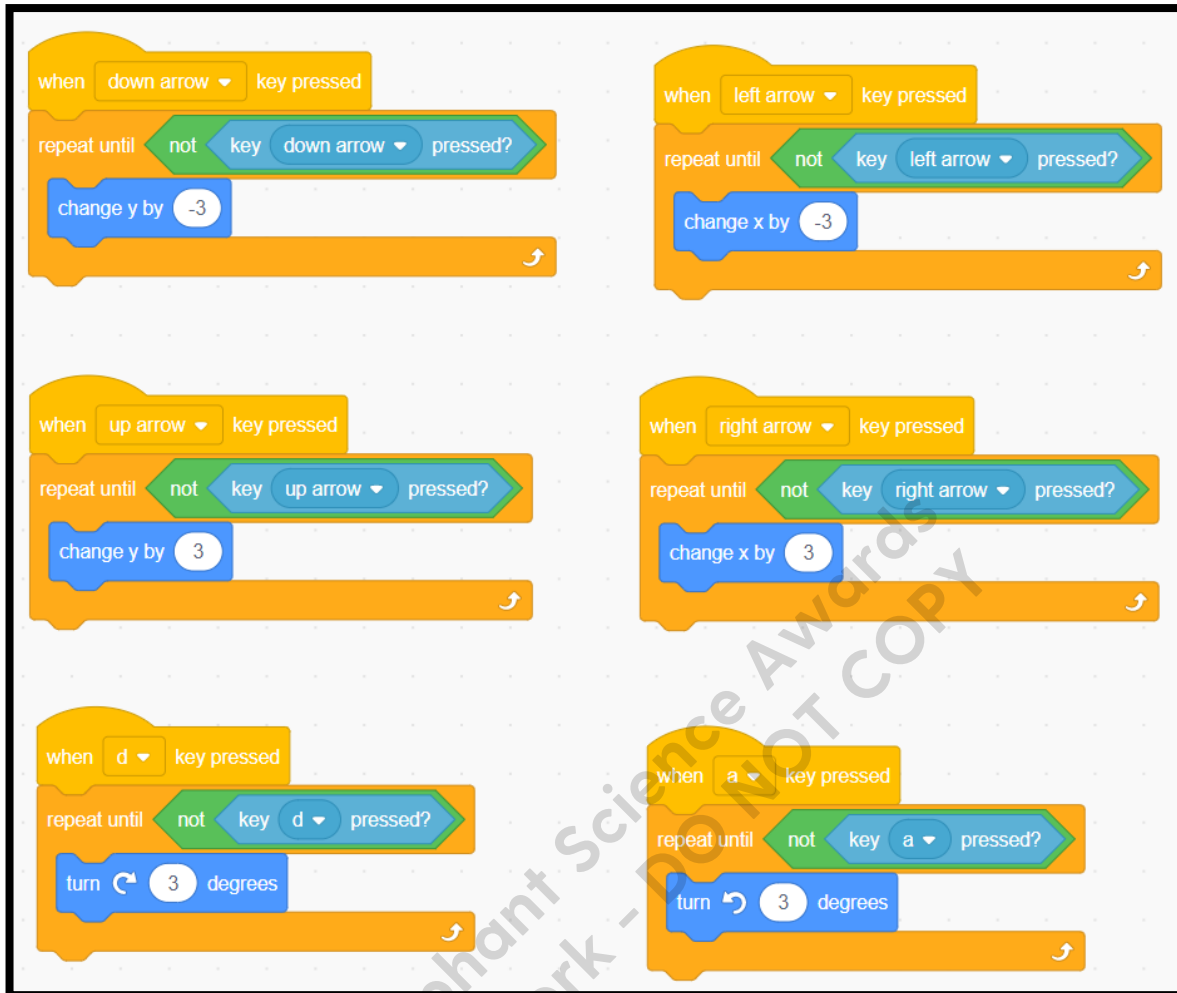
LEVEL THREE – Multiply numbers

Once the player successfully navigates the virus to the destination marked with an 'X' a second screen appears with the virus telling the player that they have done a good job by reaching the throat area but they need to help the virus infect a cell so it can start multiplying.

The code for the virus initiates the start of the game once he has finished speaking. See the code below.



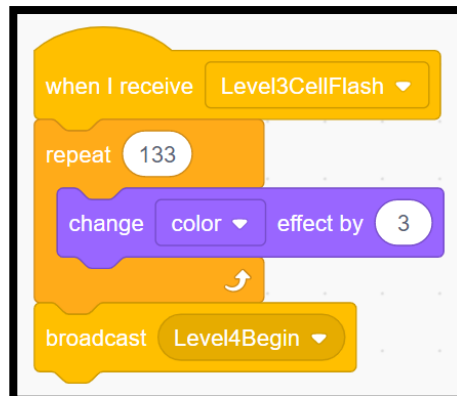
The virus tells the player to use the arrow keys to navigate the virus so that one of the spike proteins on the virus binds with the receptor on the cell.



Once the virus binds successfully to the receptor on the cell the RNA moves from the virus in to the cell ready to start replicating.



I made the cell flash and play a sound once the RNA entered the cell so that it was obvious to the player that it had been infected. This also indicates that they have completed this level of the game.

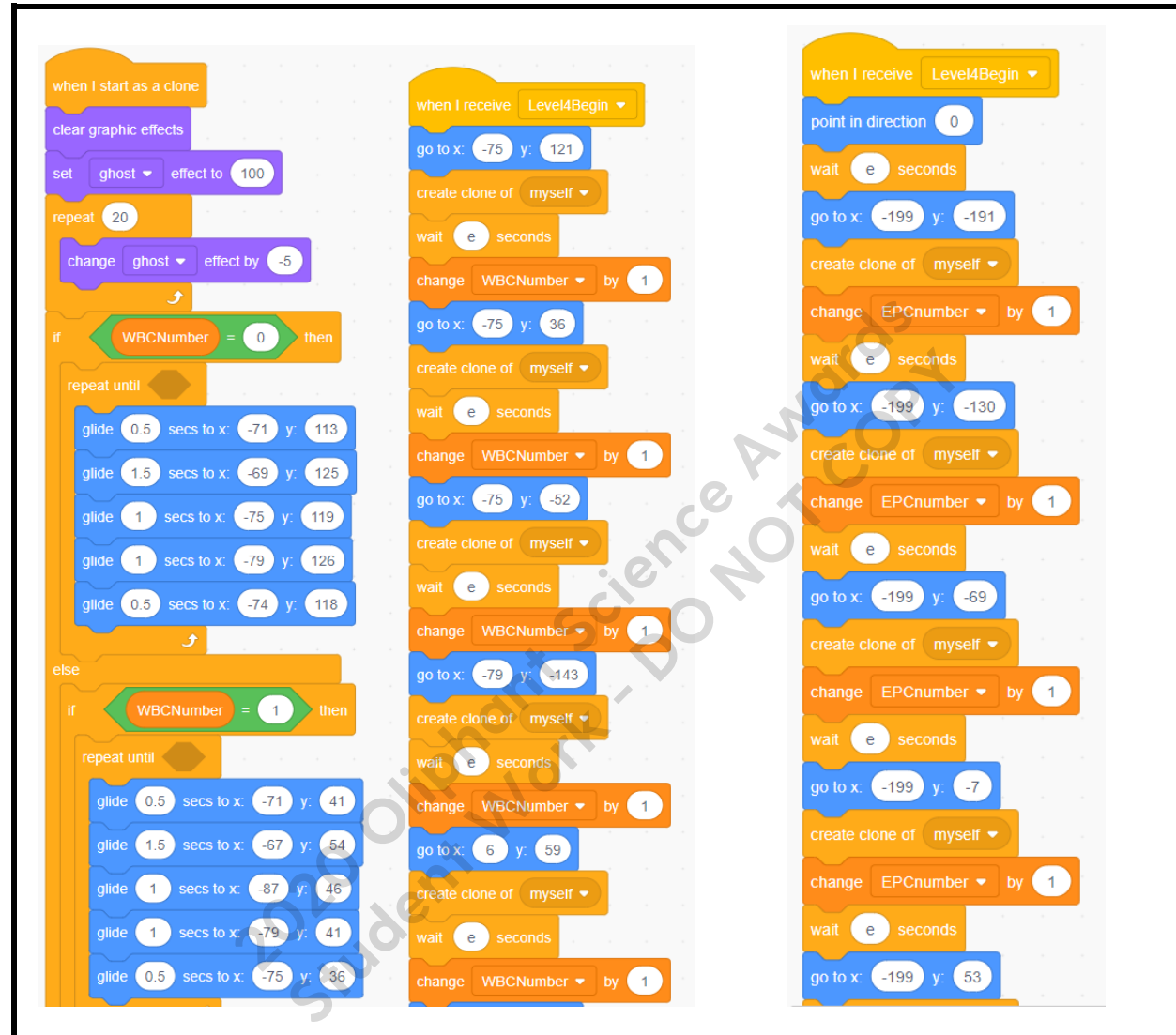


Once completed the virus tells the player that they have successfully injected their RNA in to the cell. The now infected cell has been tricked by the virus and is now reproducing the virus until it is full of newly replicated virus. Once full the cell will burst spreading more virus to nearby cells where the process will be repeated.

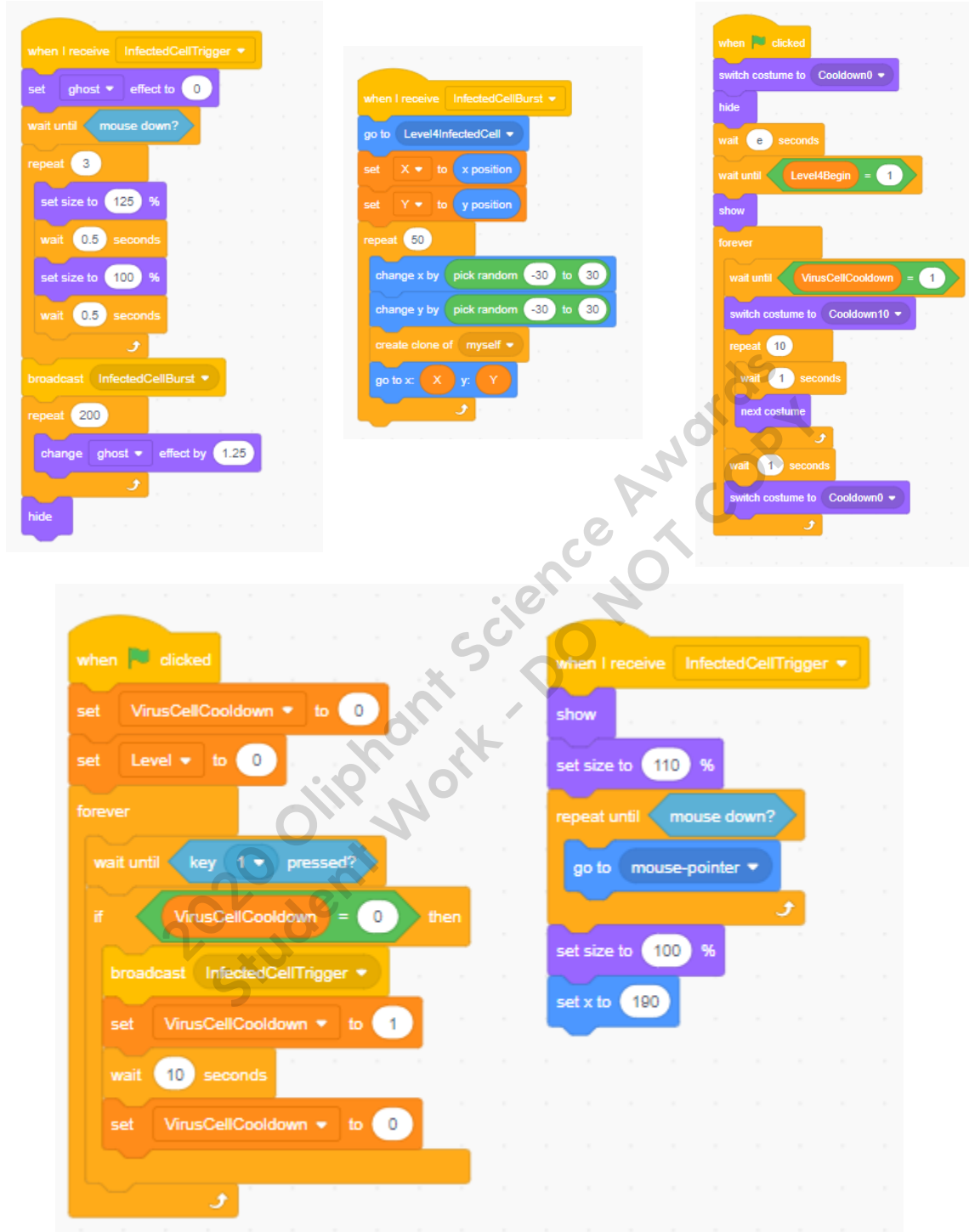


LEVEL FOUR - Lung invasion a cytokine storm!

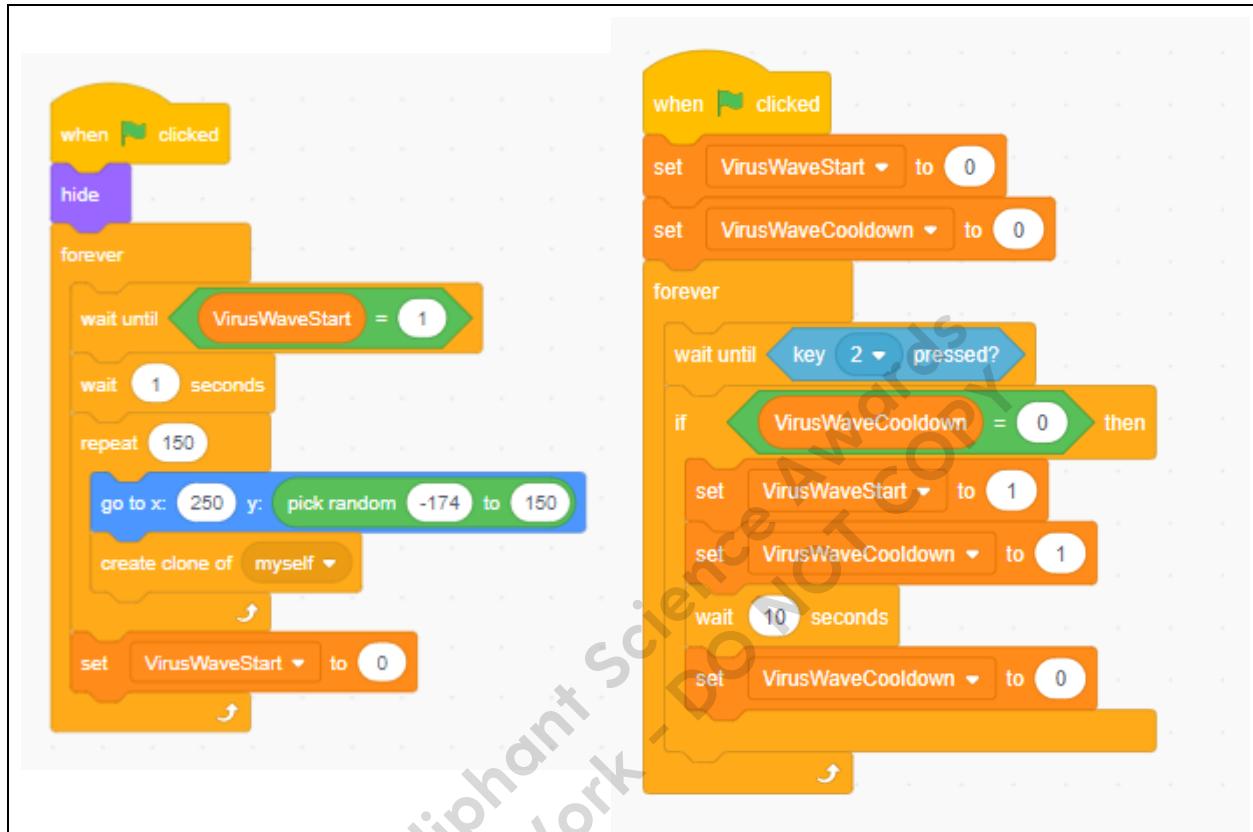
In this level I wanted to simulate the internal immune systems response to an invading virus in the lung. Although the virus attack can occur in many locations within the human body, I chose the lung as the infection point because it is a common place for a viral infection.



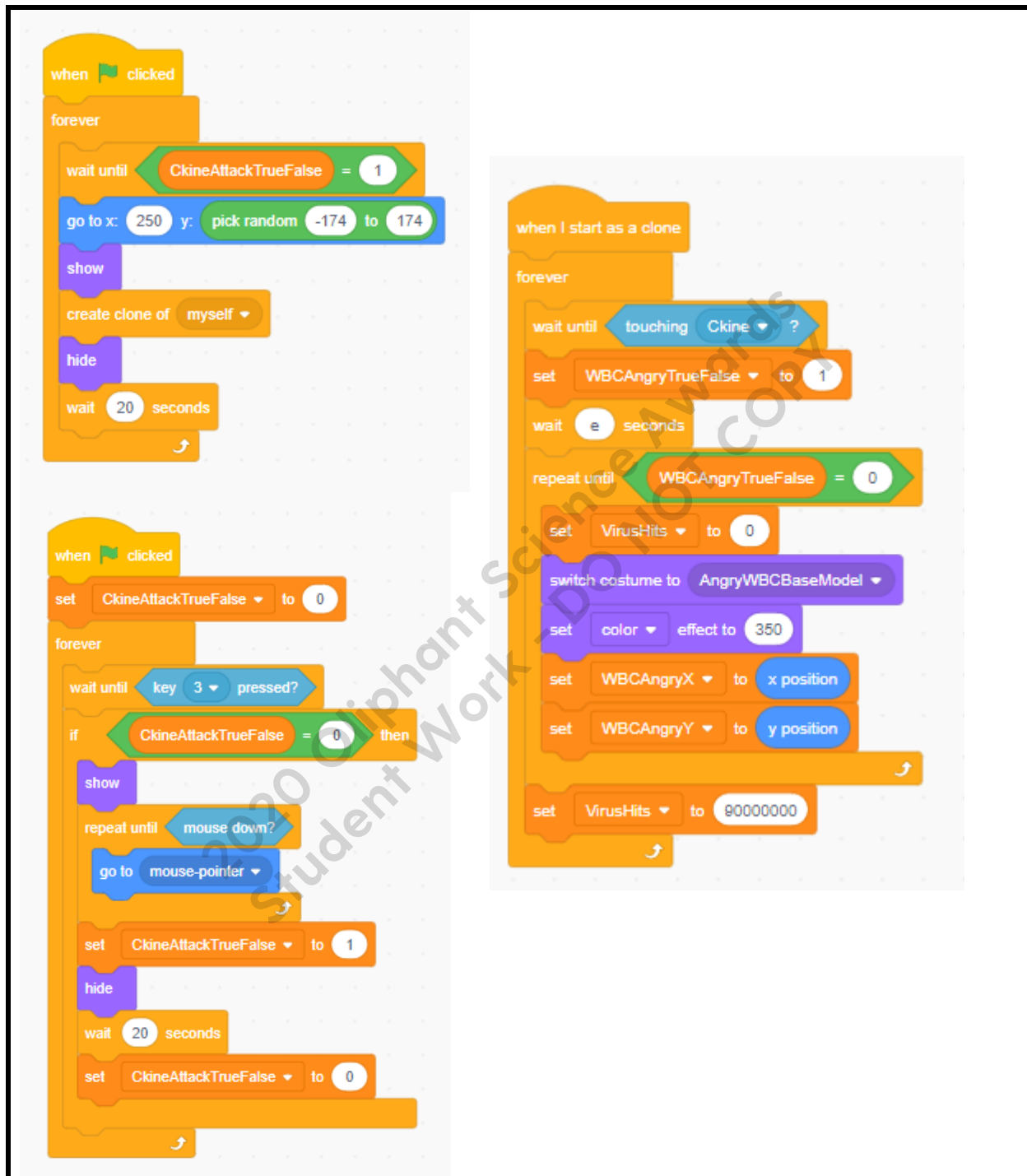
This level begins with three rows of epithelial cells and seven white blood cells ready to protect the epithelial cells. I animated the white blood cells to make it more interesting and to capture invading viruses more effectively.



Often in a location where a virus is attacking infected cells explode (once they have reached their capacity) spreading a lot more virus in the area. I wanted the player to control this element of the game so I added a script that would allow the player to drag and drop an infected cell to add to the virus numbers. I used the following script to control that movement and the number of viruses that are deployed. I wanted the player to consider the cell deployment strategically so I added a cool down element to the sprite so that it could only be deployed every 10 seconds.

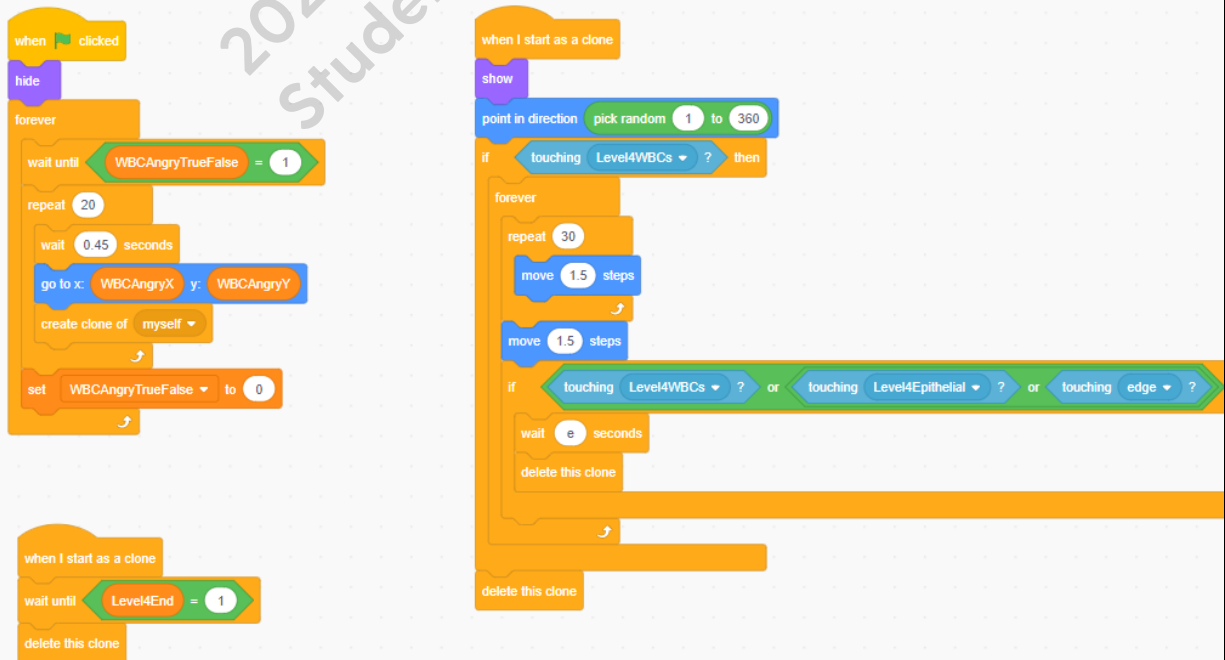
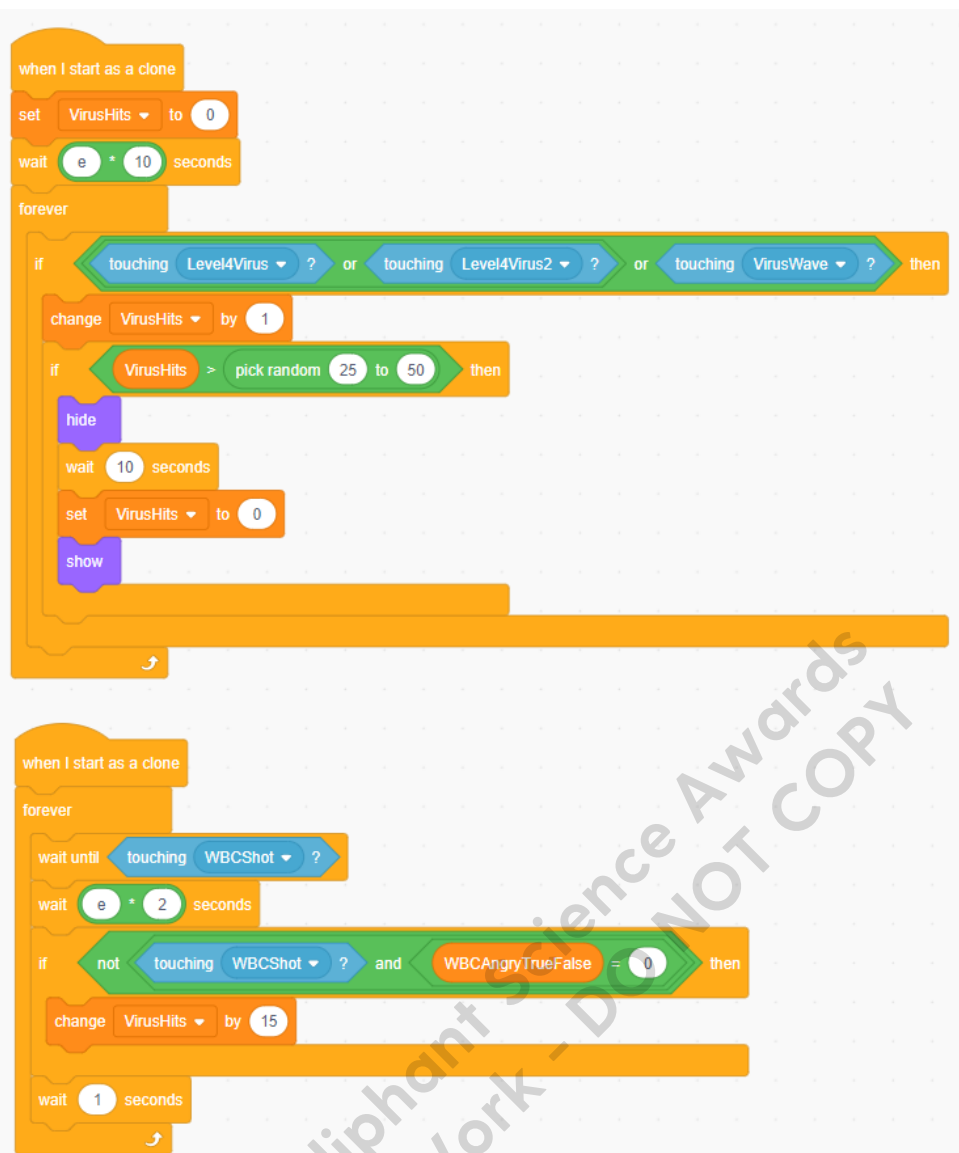


There are times when a virus attacks an area of the human body the virus increases its numbers periodically (in a wave). This is a natural part of the fight with the human immune system. To simulate this, I added a sprite that the player could control that creates a lull followed by an intense wave of viruses. This script also has a cooldown of 10 seconds so that the player has to consider this action strategically. An example of the script I used for this part of the game is below.

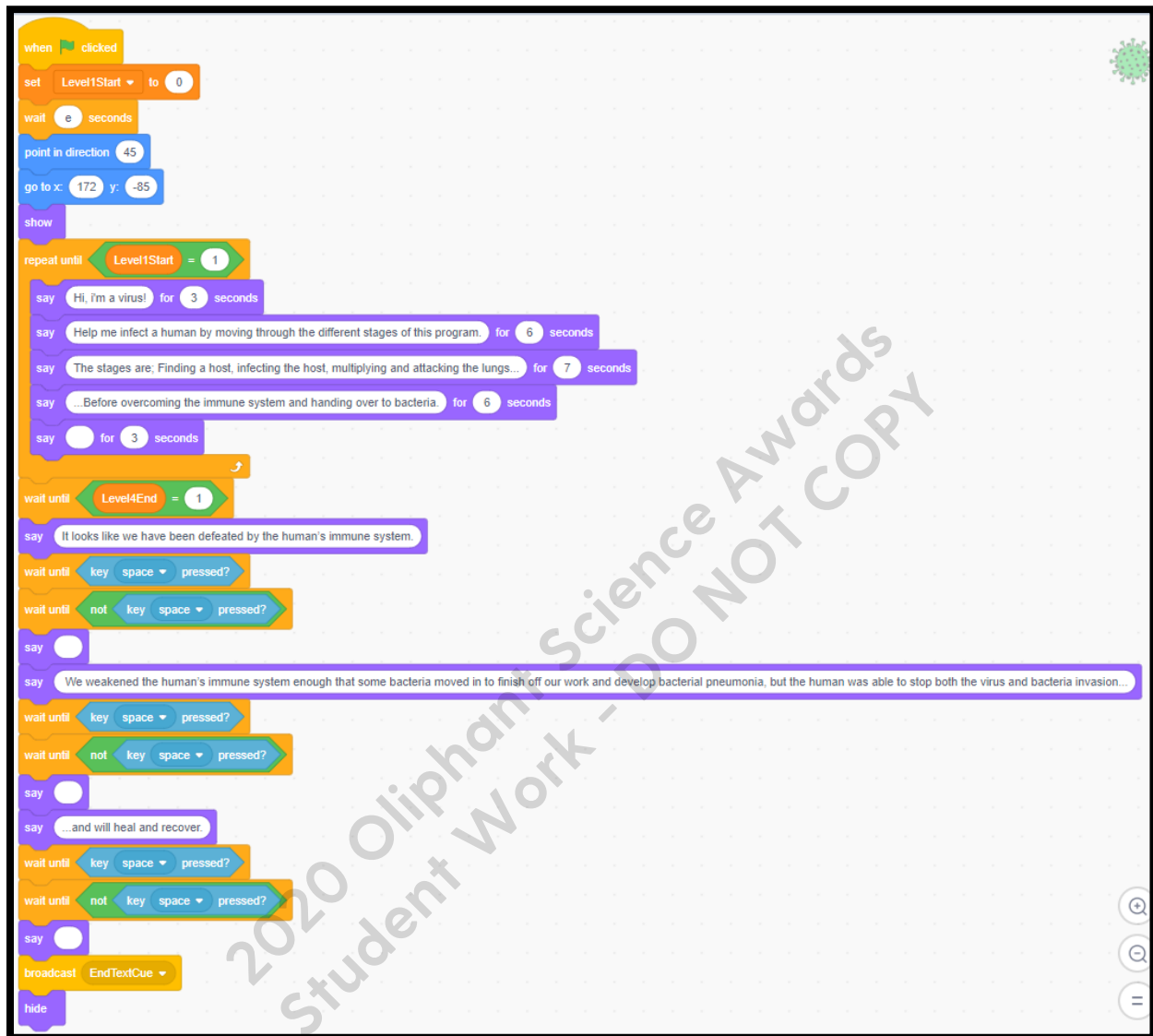


Cytokines chemicals are released by white blood cells when skin is damaged or invading pathogens are identified. This tells other cells in the area that there is trouble and that they should come and help. Sometimes the Killer T and Neutrophil cells send signals to infected or damaged cells that they should die. Viruses are very clever in that they trick these cells to sometimes tell kill healthy cells. This exacerbates their impact on an area they are infecting significantly weakening the body's response. Originally, I split these chemicals up into their sub-categories but this made the game too complicated so I left them as a general classification and created a way for the player to use cytokine signals in the game. I added a script for controlling the cooldown of the cytokine, another for spawning the cytokine itself and a third script to change the white blood cell's colour and behaviour. See the code below.





The scripts for the death and damage of epithelial and white blood cells in this level were quite difficult to balance. I wanted to make sure that the cells had enough health to progress through the level quickly but did not make it too easy to complete. The script I used for this part of the game is below.



LEVEL FIVE - Handing over to bacteria and the ending

I added this level to wrap up the game and provide a few last comments. I wanted to make sure that the player knew that the virus had weakened the body enough that a very serious bacterial infection developed in the lungs after the virus had been defeated but that the human recovered. There are so many people who have been affected by Covid-19 I wanted to make sure that the player knew that the infected person in the game survived.

THE END.....well almost....

Some acknowledgements.....

Thanks to Mum and Dad.

Dad helped me to research the human body's immune system and what books or internet pages I might like to look at or read to learn more.

Dad helped me carry out the experiments but I did all the measuring, observations and recording of data.

Sometimes Dad sat with me when I did the programming and asked me questions but I put the code together myself.

I would like to thank ME because I wrote in my journal when I worked on my entry, I continued to work on this project even when things got tough, I learnt a lot about this topic and I finished it on time! I had a goal and I accomplished it. I am really proud of that.

THE END

Project Reflection

I really enjoyed this project although when I first decided to make a scratch programme about a virus infecting a human body, I didn't realise how big the project was going to be. I didn't understand how many aspects there were to the immune system and it wasn't until I started researching it that I began to consider how many sprites, costumes and lines of code would be needed.

The scope of the game also increased the more I learned and understood about viruses so it was easy to get sidetracked. This was very obvious when I was considering how a virus tricks the immune response in to hurting itself. This part of the game became very technical and clunky to play so I ended up redesigning those levels. By deleting a lot of that code, I was able to make sure that the game is still fun to play but also gives clear information on how a virus achieves this. This also really tested my ability to make sure that any code was very efficient otherwise the game would have lag problems.

If I was going to make a project like this again, I would try to plan the scope of the project more at the start and consider other coding options before beginning. This is a really important step and although I did a lot of planning, I think I could have given it more attention to at the start of this project. Luckily it turned out ok and I was able to finish the project. I am very proud of this game and I hope everyone who plays it learns something and has fun!

2020 Oliphant Science Projects
Student Work - DO NOT REMOVE

Bibliography

BOOKS

The Human Body, London, Quercus. Brewer, S (2011)

Life of Poo, London, Kyle Books. Hart, P (2015)

Human Body from A to Z, New York, Marshall Cavendish Reference, Smith, V, Temple, N and Clifton, M (2012)

How the Body Works: the facts visually explained, London, Dorling Kindersley Limited (2016)

Tell me About? Answers to Hundreds of Questions! London, Chancellor Press (2010)

The Body: a complete user's guide, New York, National Geographic Society (2014)

Molecular Cell Biology, 4th Edition, New York, W. H. Freeman (2000)

A kid's Guide to Viruses and Bacteria, New York, Rae Simons (2016)

WEBSITES

<https://au.pinterest.com/pin/227220743675314751>

www.education.com/worksheets

<https://www.sprisers-resource.com>

www.spritedatabase.net

<http://ourmusicbox.com>

<https://www.soundsnap.com>

https://www.sprisers-resource.com/playstation_2/phantombrave/sheet/4988/

<https://www.sounds-resource.com>

<https://scratch.mit.edu/discuss/>

<http://www.orange freesounds.com/>

All of the code for my game

StartBackground

when @greenFlag clicked
hide

when I receive [SpawnVirus v]
glide (11) secs to x: (-700) y: (0)
hide

when I receive [BeginGame v]
show
set [ghost v] effect to (0)::looks
go to [back v] layer
go [forward v] (2) layers
wait (e) seconds
go to x: (0) y: (0)
set size to (110) %

Button3

when @greenFlag clicked
set size to (150) %
go to x: (-135) y: (0)
show
wait until <<touching [mouse-pointer v]?> and
<mouse down?>>
set size to (120) %
wait until <not <mouse down?>>
wait (0.25) seconds
set size to (150) %
wait (0.2) seconds
broadcast [BeginGame v]

when I receive [BeginGame v]
hide

Level1Cougher

when @greenFlag clicked
go to [front v] layer
go to x: (-194) y: (-118)
hide

when I receive [BeginGame v]
set [End v] to [0]
show
wait (1) seconds
broadcast [Cough v]
glide (3.2) secs to x: (-400) y: (-118)
hide

when I receive [BeginGame v]
set [End v] to [0]
show
wait (1) seconds
play sound [Cough2 v] until done
wait (2.5) seconds

Level1Cougher2

when I receive [Cough v]
show
wait (0.5) seconds
hide
wait (0.5) seconds
show
wait (0.5) seconds
hide

```
wait (0.5) seconds
show
wait (0.5) seconds
hide
```

```
when @greenFlag clicked
hide
glide (1) secs to x: (18) y: (9)
```

CoronaStartandEnd

```
when @greenFlag clicked
set [Level1Start v] to [0]
wait (e) seconds
point in direction (45)
go to x: (172) y: (-85)
show
repeat until <(Level1Start) = [1]>
  say [Hi, i'm a virus!] for (3) seconds
  say [Help me infect a human by moving
through the different stages of this program.] for
(6) seconds
  say [The stages are; Finding a host, infecting
the host, multiplying and attacking the lungs...]
for (7) seconds
  say [...Before overcoming the immune
system and handing over to bacteria.] for (6)
seconds
  say [] for (3) seconds
end
wait until <(Level4End) = [1]>
say [It looks like we have been defeated by the
human's immune system.]
wait until <key [space v] pressed?>
```

```
wait until <not <key [space v] pressed?>>
say []
say [ We weakened the human's immune
system enough that some bacteria moved in to
finish off our work and develop bacterial
pneumonia, but the human was able to stop
both the virus and bacteria invasion...]
wait until <key [space v] pressed?>
wait until <not <key [space v] pressed?>>
say []
say [...and will heal and recover.]
wait until <key [space v] pressed?>
wait until <not <key [space v] pressed?>>
say []
broadcast [EndTextCue v]
hide
```

```
when I receive [BeginGame v]
hide
wait until <(Level4End) = [1]>
show
forever
  set [Level v] to [5 out of 5]
end
```

Level1Virus

```
when @greenFlag clicked
hide variable [Level v]
set [Level v] to [0]
go [forward v] (4) layers
clear graphic effects
hide
point in direction (90)
```



```

set size to (15) %
go to x: (-107) y: (44)

when [up arrow v] key pressed
repeat until <not <key [up arrow v] pressed?>>
  change y by (3)
end

when [down arrow v] key pressed
repeat until <not <key [down arrow v]
pressed?>>
  change y by (-3)
end

when [right arrow v] key pressed
repeat until <not <key [right arrow v]
pressed?>>
  change x by (3)
end

when [left arrow v] key pressed
repeat until <not <key [left arrow v] pressed?>>
  change x by (-3)
end

when I receive [StartGame v]
set [Level1Start v] to [1]
set [Level1Reset v] to [0]
set [Level1VirusDie v] to [0]
set [Virus1PreDeath v] to [0]
set [Level1Death v] to [0]
set [ghost v] effect to (0)::looks
repeat until <(Virus1PreDeath) = [1]>

```

```

wait (e) seconds
if <<touching [Wind v]?> or <touching [Dust
v]?>> then
  set [Virus1PreDeath v] to [1]
  set [Level1Death v] to [1]
  repeat (2)
    set [ghost v] effect to (75)::looks
    wait (0.25) seconds
    set [ghost v] effect to (0)::looks
    wait (0.25) seconds
  end
  repeat (60)
    turn @turnRight (4) degrees::motion
    change size by (-1)
    change [ghost v] effect by (1)::looks
  end
  stop [other scripts in sprite v]
  set [Level1Reset v] to [1]
  broadcast [Level1Reset v]
end
end

when I receive [Level1Reset v]
point in direction (90)
clear graphic effects
set size to (65.2) %
go to x: (-172) y: (14)
wait (1) seconds
say [Oops! I can't survive if I touch the dust or
get blown away by the wind. Let's try again!] for
(5) seconds
broadcast [StartGame v]

```

```

when @greenFlag clicked
forever
  if on edge, bounce
end

when I receive [Level2Begin v]
wait (e) seconds
wait until <(Level2IntroductionYN) = [1]>
set [Level2Reset v] to [0]
repeat (30)
  change size by (-2)
end
forever
  if <<touching color [#323e34]?> or
  <<touching color [#2f2f2f]?> or <touching
  [Level2Map v]?>>> then
    set [Level2Reset v] to [1]
    set size to (65.2) %
    say [In this level, don't touch the sides of
    the throat. Try again!] for (5) seconds
    broadcast [Level2Begin v]
  end
end

when @greenFlag clicked
forever
  go to [front v] layer
end

when I receive [Level2End v]
hide

when I receive [StartGame v]

```

```

forever
  repeat (pick random (100) to (350))
    turn @turnRight (1.5) degrees::motion
  end
  repeat (pick random (100) to (350))
    turn @turnLeft (1.5) degrees::motion
  end
end

when @greenFlag clicked
wait (e) seconds
wait until <(Level1StopSpawning) = [1]>
repeat (60)
  go to x: (-140) y: (0)
  wait (0.1) seconds
end
forever
  wait (1) seconds
end

when I receive [BeginGame v]
wait (1) seconds
show
broadcast [SpawnVirus v]
repeat (100)
  change size by (0.5)
  turn @turnRight (3) degrees::motion
  change x by (2)
end
wait (4) seconds
glide (2) secs to x: (-165) y: (0)
say [IMPORTANT - Make sure you press the
SPACE BAR after I say anything. Press the

```

```

SPACE BAR now. ]
wait until <key [space v] pressed?>
wait until <not <key [space v] pressed?>>
say [As you play this game you will learn about
the immune system of the human body and
how it responds to an invading virus. A virus
infection is very complicated and difficult so I
will help explain the various stages.]
wait until <key [space v] pressed?>
wait until <not <key [space v] pressed?>>
say [ At different points throughout the game I
will appear to explain what is happening or how
to play the game. Although I have tried to keep
these words to a minimum (so it doesn't
interrupt the flow of the game) longer
explanations are required at some key points. ]
wait until <key [space v] pressed?>
wait until <not <key [space v] pressed?>>
say [ The intercellular aspects of the immune
response are very complicated and difficult to
follow so I decided to simplify it in the game. I
think I have found a good balance between
providing enough information so that you can
learn about a virus infection but not so much
that players will feel overwhelmed.]
wait until <key [space v] pressed?>
wait until <not <key [space v] pressed?>>
say [TASK 1: Manoeuvre the virus to the
human on the right hand side of the screen.
Make sure to avoid the dust and wind along the
way. TIP: use the arrow keys to navigate]
wait until <key [space v] pressed?>
wait until <not <key [space v] pressed?>>

```

```

say []
say [ I hope you enjoy the game and learn
something as well. Good luck!]
wait until <key [space v] pressed?>
wait until <not <key [space v] pressed?>>
say []
say [Credit for websites where I used
sprites/sound effects in this game have been
listed in the project bibliography page. ]
wait until <key [space v] pressed?>
wait until <not <key [space v] pressed?>>
say []
broadcast [StartGame v]
set [Level v] to [1 out of 5]
set [Level2IntroductionYN v] to [0]
show variable [Level v]

when I receive [Level2Begin v]
if <([Level2IntroductionYN] = [0])> then
    hide variable [Level v]
    say [In order for me to overcome the new
human host I need to multiply. This is often
done in the cells of the throat. ]
    wait until <key [space v] pressed?>
    wait until <not <key [space v] pressed?>>
    say []
    say [TASK 2: Try to navigate me to the red
'X' in the throat area using the arrow keys.
Once there I can multiply for 2-5 days until I
have sufficiently grown in numbers to invade
the lung area.]
    wait until <key [space v] pressed?>
    wait until <not <key [space v] pressed?>>

```

```

say []
say [Tip: avoid the sides of the mouth and
throat by using the arrow keys.]
wait until <key [space v] pressed?>
wait until <not <key [space v] pressed?>>
say []
set [Level2IntroductionYN v] to [1]
show variable [Level v]
end

when I receive [Level2Begin v]
forever
  if <<(Level2IntroductionYN) = [0]> or
<(Level2Reset) = [1]>> then
    repeat until <not <<(Level2IntroductionYN)
= [0]> or <(Level2Reset) = [1]>>>
      point in direction (90)
    end
  end
end

when I receive [Level2Begin v]
forever
  if <<touching color [#323e34]?> or
<<touching color [#2f2f2f]?> or <touching
[Level2Map v]?>>> then
    repeat (15)
      go to x: (-140) y: (0)
      wait (0.1) seconds
    end
  end
end
end

```

```

when I receive [StartGame v]
forever
  wait until <<touching [Wind v]?> or <touching
[Dust v]?>>
  play sound [pacman death v] until done
end

when I receive [Level2Begin v]
forever
  wait until <key [up arrow v] pressed?>
  repeat until <not <key [up arrow v]
pressed?>>
    change y by (-1.5)
  end
end

when I receive [Level2Begin v]
forever
  wait until <key [right arrow v] pressed?>
  repeat until <not <key [right arrow v]
pressed?>>
    change x by (-1.5)
  end
end

when I receive [Level2Begin v]
forever
  wait until <key [left arrow v] pressed?>
  repeat until <not <key [left arrow v]
pressed?>>
    change x by (1.5)
  end
end
end

```

```

when I receive [Level2Begin v]
  forever
    wait until <key [down arrow v] pressed?>
    repeat until <not <key [down arrow v]
    pressed?>>
      change y by (1.5)
    end
  end
end

```

Wind

```

when I receive [StartGame v]
  forever
    create clone of [myself v]
    repeat (15)
      wait (5) seconds
      create clone of [myself v]
      if <<(Level1StopSpawning) = [1]> or
      <(Level1Death) = [1]>> then
        stop [this script v]
      end
    end
    repeat (20)
      wait (4.5) seconds
      if <<(Level1StopSpawning) = [1]> or
      <(Level1Death) = [1]>> then
        stop [this script v]
      end
      create clone of [myself v]
    end
  end
end

```

when I start as a clone

```

set [UpOrDown v] to (pick random (1) to (2))
if <(UpOrDown) = [1]> then
  set y to (-240)
  point in direction (0)
else
  set y to (176)
  point in direction (180)
end
set x to (239)
show
repeat (20)
  change x by (-2)
  if <touching [Level1Virus v]?> then
    broadcast [Level1VirusDie v]
    wait (eee) seconds
    stop [other scripts in sprite v]
    delete this clone
  end
end
repeat until <touching [edge v]?>
  change x by (-2)
  if <touching [Level1Virus v]?> then
    broadcast [Level1VirusDie v]
    stop [other scripts in sprite v]
    wait (eeeeeeeeeeeeeeee) seconds
    delete this clone
  end
end
delete this clone

when @greenFlag clicked
set [Level1StopSpawning v] to [0]
hide

```

```

when I start as a clone
forever
  switch costume to [SmallWind v]
  wait (1) seconds
  switch costume to [MedWind v]
  wait (1) seconds
  switch costume to [LargeWind v]
  wait (1) seconds
  switch costume to [MedWind v]
  wait (1) seconds
end

```

```

when I start as a clone
set size to (50) %
wait until <(Level1Reset) = [1]>
delete this clone

```

```

when I receive [Level1Reset v]
forever
  if <(Level1Reset) = [1]> then
    delete this clone
  end
end

```

```

when I receive [Level1Reset v]
set [Level1StopSpawning v] to [0]

```

Dust

```

when I start as a clone
wait until <<(Level1Reset) = [1]> or
<(Level1StopSpawning) = [1]>>
delete this clone

```

```

when I receive [StartGame v]
wait (10) seconds
create clone of [myself v]
repeat (10)
  wait (6) seconds
  create clone of [myself v]
  if <<(Level1StopSpawning) = [1]> or
<(Level1Death) = [1]>> then
    stop [this script v]
  end
end
repeat (10)
  wait (5.5) seconds
  create clone of [myself v]
  if <<(Level1StopSpawning) = [1]> or
<(Level1Death) = [1]>> then
    stop [this script v]
  end
end
end

```

```

when I start as a clone
forever
  turn @turnRight (3) degrees::motion
end

```

```

when @greenFlag clicked
hide

```

```

when I start as a clone
set x to (239)
set y to (pick random (-127) to (127))
show

```



```

repeat (20)
  change x by (-3)
  if <touching [Level1Virus v]?> then
    broadcast [Level1VirusDie v]
    wait (eee) seconds
    delete this clone
  end
end
repeat until <touching [edge v]?>
  change x by (-3)
  if <touching [Level1Virus v]?> then
    broadcast [Level1VirusDie v]
    wait (0.2) seconds
    delete this clone
  end
end
delete this clone

when I receive [Level1Reset v]
stop [other scripts in sprite v]

when I start as a clone
forever
  if <(Level1Reset) = [1]> then
    delete this clone
  end
end

when I start as a clone
set [pixelate v] effect to (0)::looks
forever
  if <touching [Wind v]?> then
    set [pixelate v] effect to (30)::looks
  end
end

```

```

if <(y position) > [0]> then
  forever
    change y by (-5)
  end
else
  forever
    change y by (5)
  end
end
end
end
end

when I start as a clone
if <<(y position) = [0]> or <(y position) > [0]>>
then
  forever
    repeat (pick random (50) to (75))
      change y by (-2.5)
    end
    repeat (pick random (50) to (75))
      change y by (2.5)
    end
  end
end
else
  forever
    repeat (pick random (50) to (75))
      change y by (2.5)
    end
    repeat (pick random (50) to (75))
      change y by (-2.5)
    end
  end
end
end
end

```

```

when I receive [Level1Reset v]
set [Level1StopSpawning v] to [0]

```

Level1Background3

```

when @greenFlag clicked
set [ghost v] effect to (0)::looks
go to [front v] layer
wait (e) seconds
go [backward v] (1) layers
go to x: (475) y: (0)
hide
set size to (110) %
wait until <(Level1BackdropStop) = [1]>
create clone of [myself v]
wait (e) seconds
stop [this script v]

```

```

when I start as a clone
show
forever
  move (-2) steps
  if <<(x position) = [0]> or <[0] > (x position)>>
then
  stop [this script v]
end
end

```

```

when I receive [Level2End v]
repeat (100)
  change [ghost v] effect by (1)::looks

```

```

end
hide

```

Level2Map

```

when I receive [StartGame v]
repeat (40)
  wait (1) seconds
  if <(Level1Death) = [1]> then
    stop [this script v]
  end
end
set [Level1StopSpawning v] to [1]
set [Level1BackdropStop v] to [1]
wait (1) seconds
set size to (400) %
show
go to x: (460) y: (-106)
repeat until <<(x position) = [125]> or <(x position) < [125]>>
  change x by (-2)
end
wait (2) seconds
broadcast [Level2Begin v]
set [Level2Begin v] to [1]
set [Level1BackdropStop v] to [1]
set [Level v] to [2 out of 5]

```

```

when @greenFlag clicked
go to [front v] layer
hide
set [Level2Begin v] to [0]
set [Level1BackdropStop v] to [0]

```

```
when I receive [Level2End v]
hide
```

Level2goal

```
when @greenFlag clicked
hide
go to x: (123) y: (-30)
```

```
when I receive [Level2Begin v]
set [Level2Begin v] to [1]
repeat (2)
  go to [front v] layer
  show
  wait (0.5) seconds
  hide
  wait (0.5) seconds
end
show
```

```
when I receive [Level2Begin v]
wait until <touching [Level1Virus v]?>
stop all sounds
wait (2.07) seconds
set [Level3Begin v] to [1]
broadcast [Level2End v]
set [Level v] to [3 out of 5]
hide
```

```
when I receive [Level2Begin v]
wait until <touching [Level1Virus v]?>
play sound [Win v] until done
```

Level3Cell

```
when @greenFlag clicked
hide
set size to (300) %
set [color v] effect to (0)::looks
```

```
when @greenFlag clicked
wait (e) seconds
wait until <(Level3Begin) = [1]>
show
```

```
when I receive [Level3CellFlash v]
repeat (133)
  change [color v] effect by (3)::looks
end
broadcast [Level4Begin v]
```

```
when I receive [Level4Begin v]
hide
```

Level3Virus

```
when [up arrow v] key pressed
repeat until <not <key [up arrow v] pressed?>>
  change y by (3)
end
```

```
when [left arrow v] key pressed
repeat until <not <key [left arrow v] pressed?>>
  change x by (-3)
end
```

```
when [right arrow v] key pressed
repeat until <not <key [right arrow v]
```

```

pressed?>>
  change x by (3)
end

when [down arrow v] key pressed
repeat until <not <key [down arrow v]
pressed?>>
  change y by (-3)
end

when @greenFlag clicked
set [Level3Begin v] to [0]
set size to (100) %
hide
wait until <(Level3Begin) = [1]>
show

when @greenFlag clicked
repeat until <(Level3Begin) = [1]>
  go to x: (-175) y: (0)
end

when I receive [Level2End v]
set [Level3Begin v] to [1]

when [d v] key pressed
repeat until <not <key [d v] pressed?>>
  turn @turnRight (3) degrees::motion
end

when [a v] key pressed
repeat until <not <key [a v] pressed?>>
  turn @turnLeft (3) degrees::motion

```

```

end

when I receive [Level2End v]
hide variable [Level v]
set [Level3StoppedTalking v] to [0]
go to x: (-332) y: (321)
wait (1) seconds
glide (3) secs to x: (-194) y: (147)
point in direction (-117)
say [Now I have made it to the throat I need to
infect a healthy epithelial cell with my RNA so
that the cell can start replicating me.]
wait until <key [space v] pressed?>
wait until <not <key [space v] pressed?>>
say []
say [TASK 3: Navigate me to the red 'X' using
the arrow keys so that my protein spike binds
with the healthy cell's receptor.]
wait until <key [space v] pressed?>
wait until <not <key [space v] pressed?>>
say []
set [Level3StoppedTalking v] to [1]
show variable [Level v]

when I receive [Level2End v]
forever
  if <<touching [Level3Receptor v]?> or
<touching [Level3Cell v]?>> then
    glide (1) secs to x: (-193) y: (14)
  end
end

when @greenFlag clicked

```

```

set [Level3StoppedTalking v] to [0]
wait until <(Level3StoppedTalking) = [1]>
forever
  go to [front v] layer
end

```

```

when I receive [Level3Bind v]
glide (1) secs to x: (-159) y: (17)
point in direction (-111)
set [X v] to (x position)
set [Y v] to (y position)
forever
  go to x: (X) y: (Y)
end

```

```

when I receive [Level4Begin v]
hide

```

```

when I receive [Level2End v]
wait (4) seconds
repeat until <(Level3StoppedTalking) = [1]>
  go to x: (-194) y: (147)
end

```

Level3Receptor

```

when I receive [Level2End v]
show

```

```

when @greenFlag clicked
hide

```

```

when I receive [Level4Begin v]
hide

```

Level3ReceptorBackground

```

when I receive [Level2End v]
show

```

```

when @greenFlag clicked
hide

```

```

when I receive [Level4Begin v]
hide

```

Level3ReceptorBackground2

```

when I receive [Level2End v]
show

```

```

when @greenFlag clicked
hide

```

```

when I receive [Level4Begin v]
hide

```

RNA Segment

```

when I receive [Level3Bind v]
wait (2) seconds
show
go to x: (-24) y: (-6)
point in direction (10)
glide (2) secs to x: (85) y: (-16)
repeat (10)
  turn @turnRight (3) degrees::motion
end
glide (2) secs to x: (156) y: (-112)
repeat (10)

```

```
    turn @turnLeft (4) degrees::motion
end
glide (0.5) secs to x: (150) y: (-110)
broadcast [Level3CellFlash v]
```

```
when @greenFlag clicked
hide
```

```
when I receive [Level4Begin v]
hide
```

ReceiverTriggerLevel3

```
when @greenFlag clicked
hide
```

```
when I receive [Level2End v]
show
```

```
when @greenFlag clicked
wait until <touching [Level3Virus v]?>
broadcast [Level3Bind v]
hide
```

Level3Goal

```
when @greenFlag clicked
clear graphic effects
hide
go to x: (-14) y: (-8)
```

```
when I receive [Level2End v]
go to x: (-14) y: (-8)
show
wait (3) seconds
```

```
repeat (3)
  show
  wait (0.5) seconds
  hide
  wait (0.5) seconds
  show
end
```

```
when I receive [Level4Begin v]
hide
```

```
when I receive [Level3Bind v]
wait (2) seconds
stop all sounds
play sound [Fairy Dust Magic Sound
Effect_cut.mp3 v] until done
```

Level4WBCs

```
when @greenFlag clicked
set [CkineAttackTrueFalse v] to [0]
set [WBCNumber v] to [0]
set [Level4Begin v] to [0]
set size to (75) %
hide
point in direction (90)
```

```
when I receive [Level4Begin v]
go to x: (-75) y: (121)
create clone of [myself v]
wait (e) seconds
change [WBCNumber v] by (1)
go to x: (-75) y: (36)
create clone of [myself v]
```



```

wait (e) seconds
change [WBCNumber v] by (1)
go to x: (-75) y: (-52)
create clone of [myself v]
wait (e) seconds
change [WBCNumber v] by (1)
go to x: (-79) y: (-143)
create clone of [myself v]
wait (e) seconds
change [WBCNumber v] by (1)
go to x: (6) y: (59)
create clone of [myself v]
wait (e) seconds
change [WBCNumber v] by (1)
go to x: (5) y: (-20)
create clone of [myself v]
wait (e) seconds
change [WBCNumber v] by (1)
go to x: (3) y: (-124)
create clone of [myself v]

when I start as a clone
wait (e) seconds
go to x: (-36) y: (72)
set [color v] effect to (0)::looks
set size to (75) %
show

when I start as a clone
clear graphic effects
set [ghost v] effect to (100)::looks
repeat (20)
    change [ghost v] effect by (-5)::looks

```

```

end
if <(WBCNumber) = [0]> then
    repeat until <>
        glide (0.5) secs to x: (-71) y: (113)
        glide (1.5) secs to x: (-69) y: (125)
        glide (1) secs to x: (-75) y: (119)
        glide (1) secs to x: (-79) y: (126)
        glide (0.5) secs to x: (-74) y: (118)
    end
else
    if <(WBCNumber) = [1]> then
        repeat until <>
            glide (0.5) secs to x: (-71) y: (41)
            glide (1.5) secs to x: (-67) y: (54)
            glide (1) secs to x: (-87) y: (46)
            glide (1) secs to x: (-79) y: (41)
            glide (0.5) secs to x: (-75) y: (36)
        end
    else
        if <(WBCNumber) = [2]> then
            repeat until <>
                glide (1) secs to x: (-78) y: (-45)
                glide (0.5) secs to x: (-69) y: (-64)
                glide (1) secs to x: (-67) y: (-56)
                glide (.5) secs to x: (-79) y: (-49)
                glide (.5) secs to x: (-81) y: (-54)
            end
        else
            if <(WBCNumber) = [3]> then
                repeat until <>
                    glide (0.5) secs to x: (-75) y: (-135)
                    glide (0.5) secs to x: (-64) y: (-146)
                    glide (1.5) secs to x: (-79) y: (-154)

```

```

        glide (.5) secs to x: (-83) y: (-145)
        glide (1) secs to x: (-78) y: (-144)
    end
else
    if <(WBCNumber) = [4]> then
        repeat until <>
            glide (1.5) secs to x: (3) y: (45)
            glide (0.5) secs to x: (-5) y: (38)
            glide (1) secs to x: (4) y: (63)
            glide (1) secs to x: (6) y: (48)
            glide (0.5) secs to x: (8) y: (57)
        end
    else
        if <(WBCNumber) = [5]> then
            repeat until <>
                glide (1) secs to x: (4) y: (-25)
                glide (0.5) secs to x: (5) y: (-21)
                glide (0.5) secs to x: (8) y: (-35)
                glide (1) secs to x: (-5) y: (-28)
                glide (1.5) secs to x: (7) y: (-27)
            end
        else
            repeat until <>
                glide (0.5) secs to x: (-6) y: (-134)
                glide (1) secs to x: (5) y: (-139)
                glide (0.5) secs to x: (0) y: (-126)
                glide (1.5) secs to x: (-2) y: (-115)
                glide (0.5) secs to x: (3) y: (-124)
            end
        end
    end
end
end
end
end

```

```

    end
end

when I start as a clone
switch costume to (pick random (1) to (13))
forever
    repeat until <(costume [number v]) = [13]>
        next costume
        wait (0.1) seconds
        set [color v] effect to (0)::looks
    end
    switch costume to [macrophage1_1 v]
    wait (0.1) seconds
end

when I start as a clone
set [VirusHits v] to [0]
wait ((e) * (10)) seconds
forever
    if <<<touching [Level4Virus v]?> or <touching
[Level4Virus2 v]?>> or <touching [VirusWave v]?>>
    then
        change [VirusHits v] by (1)
        if <(VirusHits) > (pick random (25) to (50))>
    then
        hide
        wait (10) seconds
        set [VirusHits v] to [0]
        show
    end
end
end
end

```

```

when I receive [Level4Begin v]
repeat (10)
  set [Level v] to [4 out of 5]
  wait (e) seconds
end
set [Level4Begin v] to [1]

when I start as a clone
forever
  wait until <touching [Ckine v]?>
  set [WBCAngryTrueFalse v] to [1]
  wait (e) seconds
  repeat until <(WBCAngryTrueFalse) = [0]>
    set [VirusHits v] to [0]
    switch costume to [AngryWBCBaseModel v]
    set [color v] effect to (350)::looks
    set [WBCAngryX v] to (x position)
    set [WBCAngryY v] to (y position)
  end
  set [VirusHits v] to [90000000]
end

when I start as a clone
forever
  wait until <touching [WBCShot v]?>
  wait ((e) * (2)) seconds
  if <not <<touching [WBCShot v]?> and
  <(WBCAngryTrueFalse) = [0]>>> then
    change [VirusHits v] by (15)
  end
  wait (1) seconds
end

```

```

when I start as a clone
wait until <(Level4End) = [1]>
delete this clone

```

Level4Epithelial

```

when I start as a clone
wait until <(Level4End) = [1]>
delete this clone

when I start as a clone
forever
  wait until <touching [WBCShot v]?>
  if <(WBCAngryTrueFalse) = [1]> then
    change [VirusHits v] by (5)
    change [ghost v] effect by (20)::looks
  end
end

when I receive [Level4Begin v]
wait (e) seconds
go to x: (-199) y: (-191)
create clone of [myself v]
change [EPCnumber v] by (1)
wait (e) seconds
go to x: (-199) y: (-130)
create clone of [myself v]
change [EPCnumber v] by (1)
wait (e) seconds
go to x: (-199) y: (-69)
create clone of [myself v]
change [EPCnumber v] by (1)
wait (e) seconds
go to x: (-199) y: (-7)

```

```

create clone of [myself v]
change [EPCnumber v] by (1)
wait (e) seconds
go to x: (-199) y: (53)
create clone of [myself v]
change [EPCnumber v] by (1)
wait (e) seconds
go to x: (-199) y: (113)
create clone of [myself v]
change [EPCnumber v] by (1)
wait (e) seconds
go to x: (-199) y: (173)
create clone of [myself v]
change [EPCnumber v] by (1)
wait (e) seconds
go to x: (-226) y: (-149)
create clone of [myself v]
change [EPCnumber v] by (1)
wait (e) seconds
go to x: (-226) y: (-90)
create clone of [myself v]
change [EPCnumber v] by (1)
wait (e) seconds
go to x: (-226) y: (-30)
create clone of [myself v]
change [EPCnumber v] by (1)
wait (e) seconds
go to x: (-226) y: (30)
create clone of [myself v]
change [EPCnumber v] by (1)
wait (e) seconds
go to x: (-226) y: (91)
create clone of [myself v]

```

```

change [EPCnumber v] by (1)
wait (e) seconds
go to x: (-226) y: (151)
create clone of [myself v]
change [EPCnumber v] by (1)
wait (e) seconds
go to x: (-169) y: (149)
create clone of [myself v]
change [EPCnumber v] by (1)
wait (e) seconds
go to x: (-169) y: (85)
create clone of [myself v]
change [EPCnumber v] by (1)
wait (e) seconds
go to x: (-169) y: (21)
create clone of [myself v]
change [EPCnumber v] by (1)
wait (e) seconds
go to x: (-169) y: (-45)
create clone of [myself v]
change [EPCnumber v] by (1)
wait (e) seconds
go to x: (-169) y: (-111)
create clone of [myself v]
change [EPCnumber v] by (1)
wait (e) seconds
go to x: (-169) y: (-176)
create clone of [myself v]
change [EPCnumber v] by (1)

when @greenFlag clicked
set [EPCnumber v] to [0]
hide

```

```
when I start as a clone  
show
```

```
when I start as a clone  
set [ghost v] effect to (0)::looks  
set [brightness v] effect to (0)::looks  
set [Brightness v] to [0]  
set [VirusHits v] to [0]  
forever  
  wait until <<touching [Level4Virus v]?> or  
<touching [Level4Virus2 v]?>>  
  change [VirusHits v] by (1)  
  change [ghost v] effect by (4)::looks  
  if <(VirusHits) > [15]> then  
    delete this clone  
  end  
end
```

```
when I receive [ConvertBlocks v]  
set [Blocks v] to [100000]  
show  
glide (5) secs to x: (-10) y: (-90)  
say [Convert all of my blocks to English...]  
wait until <key [space v] pressed?>  
wait until <not <key [space v] pressed?>>  
say [So that I can show everyone how much  
coding I did to create my game.]  
wait until <key [space v] pressed?>  
wait until <not <key [space v] pressed?>>  
switch costume to [HappyandProudFace v]  
glide (5) secs to x: (270) y: (-90)  
broadcast [BeginGame v]
```

Level4InfectedCell

```
when @greenFlag clicked  
set size to (100) %  
hide  
point in direction (0)
```

```
when I receive [InfectedCellTrigger v]  
show  
set size to (110) %  
repeat until <mouse down?>  
  go to [mouse-pointer v]  
end  
set size to (100) %  
set x to (190)
```

```
when I receive [InfectedCellTrigger v]  
set [ghost v] effect to (0)::looks  
wait until <mouse down?>  
repeat (3)  
  set size to (125) %  
  wait (0.5) seconds  
  set size to (100) %  
  wait (0.5) seconds  
end  
broadcast [InfectedCellBurst v]  
repeat (200)  
  change [ghost v] effect by (1.25)::looks  
end  
hide
```

```
when @greenFlag clicked  
set [VirusCellCooldown v] to [0]
```

```

set [Level v] to [0]
forever
  wait until <key [1 v] pressed?>
  if <(VirusCellCooldown) = [0]> then
    broadcast [InfectedCellTrigger v]
    set [VirusCellCooldown v] to [1]
    wait (10) seconds
    set [VirusCellCooldown v] to [0]
  end
end

```

Level4Virus2

```

when I start as a clone
wait until <(Level4End) = [1]>
delete this clone

```

```

when @greenFlag clicked
hide

```

```

when I receive [InfectedCellBurst v]
go to [Level4InfectedCell v]
set [X v] to (x position)
set [Y v] to (y position)
repeat (50)
  change x by (pick random (-30) to (30))
  change y by (pick random (-30) to (30))
  create clone of [myself v]
  go to x: (X) y: (Y)
end

```

```

when I start as a clone
show
forever

```

```

change x by (-1.5)
if <touching [Level4WBCs v]?> then
  wait (e) seconds
  delete this clone
end
end

```

```

when I start as a clone
show
wait (2) seconds
wait until <touching [edge v]?>
delete this clone

```

```

when I start as a clone
wait until <touching [Level4Epithelial v]?>
wait ((e) * (15)) seconds
delete this clone

```

VirusWave

```

when I start as a clone
wait until <(Level4End) = [1]>
delete this clone

```

```

when I start as a clone
wait until <touching [Level4Epithelial v]?>
wait ((e) * (15)) seconds
delete this clone

```

```

when I start as a clone
wait (e) seconds
wait until <(VirusWaveStart) = [0]>
show
forever

```



```

change x by (-1.5)
if <touching [Level4WBCs v]?> then
  wait (e) seconds
  delete this clone
end
end

when @greenFlag clicked
hide
forever
  wait until <(VirusWaveStart) = [1]>
  wait (1) seconds
  repeat (150)
    go to x: (250) y: (pick random (-174) to (150))
    create clone of [myself v]
  end
  set [VirusWaveStart v] to [0]
end

```

```

when I start as a clone
wait (e) seconds
wait until <(VirusWaveStart) = [0]>
wait (2) seconds
wait until <touching [edge v]?>
delete this clone

```

Ckine

```

when @greenFlag clicked
wait until <(Level4End) = [1]>
hide
stop [other scripts in sprite v]

```

```

when @greenFlag clicked
set [CkineAttackTrueFalse v] to [0]
hide

```

```

when @greenFlag clicked
forever
  wait until <(CkineAttackTrueFalse) = [1]>
  go to x: (250) y: (pick random (-174) to (174))
  show
  create clone of [myself v]
  hide
  wait (20) seconds
end

```

```

when I start as a clone
glide (3) secs to [CkineAttackTarget v]
wait ((e) * (100)) seconds
delete this clone

```

```

when I start as a clone
wait until <touching [Level4WBCs v]?>
wait ((e) * (100)) seconds
delete this clone

```

CkineAttackTarget

```

when @greenFlag clicked
set [CkineAttackTrueFalse v] to [0]
forever
  wait until <key [3 v] pressed?>
  if <(CkineAttackTrueFalse) = [0]> then
    show
    repeat until <mouse down?>
      go to [mouse-pointer v]

```

```

    end
    set [CkineAttackTrueFalse v] to [1]
    hide
    wait (20) seconds
    set [CkineAttackTrueFalse v] to [0]
  end
end

```

```

when @greenFlag clicked
hide
forever
  go to [front v] layer
end

```

```

when @greenFlag clicked
wait until <(Level4End) = [1]>
hide
stop [other scripts in sprite v]

```

WBCShot

```

when I start as a clone
wait until <(Level4End) = [1]>
delete this clone

```

```

when @greenFlag clicked
hide
forever
  wait until <(WBCAngryTrueFalse) = [1]>
  repeat (20)
    wait (0.45) seconds
    go to x: (WBCAngryX) y: (WBCAngryY)
    create clone of [myself v]
  end
end

```

```

  set [WBCAngryTrueFalse v] to [0]
end

```

```

when I start as a clone
show
point in direction (pick random (1) to (360))
if <touching [Level4WBCs v]?> then
  forever
    repeat (30)
      move (1.5) steps
    end
    move (1.5) steps
    if <<touching [Level4WBCs v]?> or
    <<touching [Level4Epithelial v]?> or <touching
    [edge v]?>>> then
      wait (e) seconds
      delete this clone
    end
  end
end
delete this clone

```

Level4End

```

when @greenFlag clicked
set [Level4End v] to [0]
hide

```

```

when I receive [Level4StartSpawning v]
go to x: (-240) y: (0)
point in direction (89)
show
repeat (10)
  wait until <touching [Level4Virus v]?>

```

```
end
broadcast [Level4End v]
set [Level4End v] to [1]
hide
```

CooldownMenu

```
when I receive [Level4End v]
hide

when @greenFlag clicked
hide
wait (e) seconds
wait until <(Level4Begin) = [1]>
show
```

InfectedCooldown

```
when @greenFlag clicked
hide
wait (e) seconds
wait until <(Level4Begin) = [1]>
show

when I receive [Level4End v]
hide
```

WaveCooldown

```
when I receive [Level4End v]
hide

when @greenFlag clicked
hide
wait (e) seconds
```

```
wait until <(Level4Begin) = [1]>
show
```

CellCooldownBox

```
when @greenFlag clicked
switch costume to [Cooldown0 v]
hide
wait (e) seconds
wait until <(Level4Begin) = [1]>
show
forever
  wait until <(VirusCellCooldown) = [1]>
  switch costume to [Cooldown10 v]
  repeat (10)
    wait (1) seconds
    next costume
  end
  wait (1) seconds
  switch costume to [Cooldown0 v]
end

when I receive [Level4End v]
hide
```

VirusCooldown

```
when I receive [Level4End v]
hide

when @greenFlag clicked
switch costume to [Cooldown0 v]
hide
wait (e) seconds
```

```

wait until <(Level4Begin) = [1]>
show
forever
  wait until <(VirusWaveCooldown) = [1]>
  switch costume to [Cooldown10 v]
  repeat (10)
    wait (1) seconds
    next costume
  end
  wait (1) seconds
  switch costume to [Cooldown0 v]
end

```

CytokineCooldown

```

when I receive [Level4End v]
hide

when @greenFlag clicked
hide
wait (e) seconds
wait until <(Level4Begin) = [1]>
show

```

Level4Intro

```

when @greenFlag clicked
hide

when I receive [Level4Begin v]
set [Level4Begin v] to [1]
show
say [Congratulations, we made it to the lungs.
Now there are thousands of viruses ready to do
some real damage. ]

```

```

wait until <key [space v] pressed?>
wait until <not <key [space v] pressed?>>
say []
say [Spawn an infected cell to create lots more
viruses in that area by pressing 1.]
wait until <key [space v] pressed?>
wait until <not <key [space v] pressed?>>
say []
say [Make the viruses move forward in a
coherent wave by pressing 2.]
wait until <key [space v] pressed?>
wait until <not <key [space v] pressed?>>
say []
say [Turn the human's immune system against
itself by sending out confusing cytokine signals
by pressing 3. Use the target scope to select a
place for the cytokine to attack.]
wait until <key [space v] pressed?>
wait until <not <key [space v] pressed?>>
say []
say [Each action has a cooldown that is shown
in the bottom right. TASK 4: Cause as much
damage as possible to finish the level! ]
wait until <key [space v] pressed?>
wait until <not <key [space v] pressed?>>
say []
hide
broadcast [Level4StartSpawning v]

```

EndText

```

when @greenFlag clicked
set [ScrollTime v] to [0]
go to x: (0) y: (-200)

```

```

hide

when I receive [EndTextCue v]
show
repeat until <(ScrollTime) = [41.5]>
  change y by (0.75)
end
hide
wait (3) seconds
stop [all v]

when I receive [EndTextCue v]
forever
  wait (0.5) seconds
  change [ScrollTime v] by (0.5)
end

```

EndTextBackground

```

when @greenFlag clicked
hide

when I receive [EndTextCue v]
show

```

Stage

```

when @greenFlag clicked
switch backdrop to [blood v]
start sound [Menu music background v]

when I receive [Level1VirusDie v]
stop all sounds

when I receive [StartGame v]

```

```

stop all sounds
start sound [Level one v]
wait until <(Level2Begin) = [1]>
switch backdrop to [Level3 v]
stop all sounds
start sound [Level two v]
wait until <(Level3Begin) = [1]>
stop all sounds
start sound [Level one v]
wait until <(Level4Begin) = [1]>
stop all sounds
start sound [Level two v]
wait until <(Level4End) = [1]>
stop all sounds
start sound [Closing credits v]
switch backdrop to [blood v]

when I receive [BeginGame v]
stop all sounds
wait (10.5) seconds
play sound [DD background v] until done

```

Forest

```

when I receive [StartGame v]
forever
  go to x: (-50) y: (0)
  go to [back v] layer
  next costume
  wait (e) seconds
end

when I receive [Level2End v]
hide

```

```
when @greenFlag clicked  
hide
```

```
when I receive [BeginGame v]  
show
```

2020 Oliphant Science Awards
Student Work - DO NOT COPY