

**ENGLISH VERSION**

**Prevent Ozone Depletion – Part I**

Hi Everyone!

Hope you are enjoying your role as a Climate Action Change Agent and becoming better at developing games. It is now time for us to get started with yet another mission. Let me give you a brief background.

You may have heard about Ozone, a gas that forms a thin protective layer around the Earth. It shields our planet from harmful solar radiation or ultraviolet rays by absorbing almost 98% of them before they hit the earth. But the ozone layer in the earth's atmosphere is getting thinner over time. Chemicals called chlorofluorocarbons or CFCs which are released on earth are leading to the depletion of the ozone layer. Chlorofluorocarbons are made of elements like carbon, fluorine, and chlorine. A single CFC molecule can destroy over 1,00,000 ozone molecules and unfortunately, ozone can be destroyed more quickly than it can be naturally created. The damaged areas in the ozone layer are generally referred to as ozone holes.

Our mission is to create a game in Scratch to save the ozone layer from getting depleted as a result of the harmful gases released by man-made things like vehicles and industries. The goal is to pop the toxic emissions before they hit the ozone layer. Remember, as these gases continue to reach the ozone layer, the colour of the ozone layer changes gradually and eventually when it becomes red, the game gets over.

In order to create the game we will work on the following four tasks –

- a) Designing the screen and coding the car
- b) Creating & popping toxic gas bubbles
- c) Creating & coding the Ozone layer
- d) Creating the Welcome & Game Over Screens

## Task – I: Designing the screen and coding the car

Let us begin with our first task by designing the Screen for our game.

As usual, delete the default cat sprite and any other images on the screen so that it is empty. Add a backdrop of your choice. Go to the extreme right at the bottom and click on the image icon. Choose an image from the available backdrops, create an image in paint or upload an image. We are uploading the backdrop of a city from our own image library. Convert the image into a vector and Stretch it to fill the screen.

Our next step is to add the sprite of a car. We are sure that you are a pro at adding sprites by now. Do not forget to resize the car after you convert it into a vector. While you are you doing that, are you wondering why do we need a car in our game? The car will keep moving from left to right and gas bubbles that will affect our ozone layer will keep following it. The harmful gas bubbles are emissions from the car itself.

We now need to determine the starting position of the car. Since we want it to move from left to right, we feel an initial position of “-200” on the X axis and “-150” on the Y axis will work well. To initialize the Car position such that it always starts from this position that we have decided upon, we will create 2 variables, one each for both the x and y axes. Go to the variables section, click ‘Make a variable’ and select the option ‘For this sprite only’. We are calling the variables “InitialPosX” & “InitialPosY” respectively.

Create a new event to set the default position of the car on both axes. Drag ‘When Flag clicked’ block from the events section. From the Variables section, attach 2 instances of ‘Set Variable to 0’ to this block. Select “InitialPosX” & “InitialPosY” from the drop downs and Set the values of the variables to -200 & -150 respectively. Drag ‘go to X & Y’ block from the motion section and replace the numbers with the variables ‘InitialPosX’ & ‘InitialPosY’. This logic sets the default position of the car for our game.

The next step is to ensure that our car keeps moving continuously from left to right until the game is over. Create a variable called “GameOver” in the Variables section. Make sure that you make it accessible to all sprites in the game. To initialize the variable, drag ‘set GameOver to 0’ and place it between ‘when flag clicked’ and ‘Set InitialPosX to’ block.

Drag the ‘Repeat Until’ block from the Control section and attach it to the ‘go to x = InitialPosX & y = InitialPosY’ block. As you are aware, the repeat until block keeps doing something as long as the condition in the hexagonal slot is true. From the Operators section, drag ‘is equal to’ operator and place it in the <hexagonal slot>. Go to the variables section and drag “Gameover” variable and place it on the left-hand side of the equal to sign. Change the value on the right-hand side of the equal to sign to 1. This will make sure that the loop runs till the value of the “GameOver” variable becomes “1”.

To move the car from left to right on the axis, drag “Change X by 10” block from the motion section and place it inside the repeat until loop. The number 10 is the current speed of the car which is too fast. Replace it by 5 to make the car slower.

Let us test our car now. Great! The car is moving from left to right.

But there is a small problem. Currently, the car stops at the right edge of the screen once it reaches its maximum x-position. We need to ensure that the car goes back to its initial position i.e. the car should automatically go back to the left edge of the screen.

Create another variable “EndPosX” for this sprite only from the Variables section. Drag ‘Set EndPosX to 0’ and place it between ‘Set InitialPosY to -150’ and ‘go to x & y’. Set the value of “EndPosX” to 280. Drag the “If then else” block from the control section and place it inside repeat until block. Move the ‘change x by 5’ block inside the else part of the condition. Drag the ‘Greater than’ block from the operators section and place it inside the <hexagonal slot>.

Go to the Motion section, take 'x position' block and place it on the left-hand side of the greater than condition. Drag 'EndPosX' from the variables section place it on the right-hand side of the operator.

Drag 'Set x to' block from the motion section and place it inside the 'if' loop. Replace the number with the 'InitialPosX' block from the variables section. Let us recapitulate our logic. We have ensured that if the position of the car on the x-axis becomes greater than the "EndPosX", then the car will go back to the "InitialPosX" else the car will keep on moving on the x axis at a speed of 5.

Time to check our coding and see if the car is working as expected.

Wow! We have completed our first task.

In the next video, we will create and pop toxic gas bubbles that will be emitted from the car.

Bye-bye. See you again!

#### HINDI VERSION

### Prevent Ozone Depletion – Part I

हेलो , everyone

उम्मीद है कि आप एक क्लाइमेट एक्शन चेंज एजेंट के रूप में अपने role को enjoy कर रहे हैं और गेम्स को develop करने में और अच्छे होते जा रहे हैं। अब हमारे लिए एक और मिशन शुरू करने का समय है। मैं आपको इसके बारे में बताती हूँ। आपने ozone के बारे में सुना होगा, एक गैस जो earth के चारों ओर एक पतली protect करने वाली layer बनाती है। यह हमारे planet को sun से आने वाली खतरनाक radiation और ultraviolet rays को earth से टकराने से पहले उनमें से लगभग 98% को absorb करके एक shield की तरह काम करती है। लेकिन Earth के atmosphere में ozone layer टाइम के साथ पतली होती जा रही है, chlorofluorocarbons or CFCs नाम के chemicals, जो Earth पर छोड़े जाते हैं, ओजोन layer को खत्म कर रहे हैं।

Chlorofluorocarbon carbon, fluorine, और chlorine जैसे elements से बना होता है। एक अकेला CFC molecule एक लाख से अधिक ozone molecules को खत्म कर सकता है, और unfortunately , ozone जितनी स्पीड से naturally बनता है उससे ज्यादा स्पीड से इसे खत्म किया

जा सकता है ओजोन layer में damage areas को ozone holes बोला जाता है। हमारा मिशन गाड़ियों और industries जैसी इंसान ने जो चीज़े बनाई, उनसे निकलने वाले harmful गैसों के कारण ozone layer को खत्म होने से बचाने के लिए स्क्रीन में गेम develop करना है, हमारा aim बाहर फ़ैल रहे ज़हरीले gases के bubbles को ozone layer से टकराने से पहले उन्हें फोड़ देना है। याद रखें कि जैसे जैसे ये गैसों ozone layer तक पहुंचती रहती हैं, जिससे ozone layer का colour धीरे-धीरे बदलता है और जब यह red हो जायगा, तो गेम खत्म हो जायगा। गेम बनाने के लिए, हम इन चार tasks पर काम करेंगे,

- स्क्रीन को डिजाइन करना और कार को कोड करना,
- जहरीली गैस के bubbles बनाना और उन्हें फोड़ना,
- ozone layer को बनाना और कोड करना
- welcome और game over screen बनाना

आइए हम अपने गेम के लिए स्क्रीन डिज़ाइन करके अपने पहले टास्क को शुरू करें। हमेशा की तरह, डिफ़ॉल्ट cat स्प्राइट और स्क्रीन पर किसी भी और image को हटा दें ताकि यह खाली हो। अपनी पसंद की backdrop add करें, सबसे नीचे right side की तरफ जाएं और image आइकन पर क्लिक करें, available backdrops में से एक image चुनें, paint में एक image बनाएं या अपलोड करें। हम अपने खुद के image library से एक सिटी का backdrop अपलोड कर रहे हैं, image को एक वेक्टर में change करें और इसको stretch करें ताकि स्क्रीन fill हो जाए।

हमारा अगला next step car के स्प्राइट को add करना है, हमें भरोसा है कि आप अब तक स्प्राइट को जोड़ने के काम में expert हो गए होंगे, आप इसे वेक्टर में change के बाद car का साइज़ change करना न भूलें। जब आप ऐसा कर रहे हैं, तो क्या आप सोच रहे होंगे कि हमें अपने गेम में car की जरूरत क्यों है? car left से right चलती रहेगी और gas के bubbles जो हमारे ozone layer को affect करेंगे और car को follow करते रहेंगे। harmful गैस bubbles car से ही बाहर निकल रहे हैं। अब हमें car के starting point को decide करना होगा, क्योंकि हम इसे left से right चलाना चाहते हैं, हमें लगता है कि x axis पर -200 और y axis पर -50 की starting position अच्छी तरह से काम करेगा।

car की position को ऐसे initialize करना है कि यह हमेशा इस position से शुरू हो जो कि हमने decide किया है इसके लिए हम दो variable बनाएंगे, एक x- axis के लिए और एक y- axis लिए। variable सेक्शन पर जाएं, क्लिक करें, एक variable बनाएं और for this Sprite only आप्शन चुनें। वैरिएबल को initial pos X और initial pos Y का नाम दें। दोनों axis पर car की डिफ़ॉल्ट position सेट करने के लिए एक नया ईवेंट बनाएं। ईवेंट सेक्शन से when flag clicked block को खींचें, वैरिएबल सेक्शन से, दो 'set variable to zero' ब्लॉक को इस block से attach करें, ड्रॉप डाउन से initial pos X और initial pos Y चुनें और variables के value को -200 और -150 सेट करें। motion section से go to X and Y block को खींचें और numbers को Initial pos X और initial pos Y से बदलें। यह logic हमारे गेम के लिए car की डिफ़ॉल्ट position

decide करेगा। हमारा next step यह पक्का करना है कि गेम finish होने तक हमारी car left से right लगातार चलती रहे। variable सेक्शन में 'game over' नाम एक variable बनाएं। और ध्यान रखें कि यह गेम के सभी स्प्राइट्स इसे access कर सकें। variable को initialize करने के लिय set game over to zero को खींचें और when flat clicked और set initial pos X to block बीच रखें। कण्ट्रोल सेक्शन से repeat until the block को खींचें और go to X equals initial pos X and Y equals initial pos Y block से इसे attach करें। repeat until block का मकसद उसके अंदर कोड तब तक चलता रहेगा जब तक हेक्सागोनल स्लॉट कि value true है। ऑपरेटर सेक्शन 'is equal to' operator को खींचें और इसे हेक्सागोनल स्लॉट में रखें, variable सेक्शन में जाएं और 'game over' variable को खींचें करें और इसे equal to sign के left साइड रखें, 'equal to' sign के राइट हैंड साइड की value change करके 1 कर दें। यह पक्का करेगा कि जब तक loop चलेगा तब तक game over variable की value 1 न हो जाए। axis पर कार को left से right की ओर ले जाने के लिए, motion section से change X by 10 blocks को खींचें और इसे repeat until loop के अंदर रखें।

number 10 car की अभी की स्पीड है, जो बहुत तेज है। कार को slow करने के लिए इसे 5 कर दें। आइए अब हम अपनी कार को test करें। great, car left से right रही है, लेकिन एक छोटी सी problem है। अभी जब कार अपने maximum x position पर पहुँचती है तो स्क्रीन के right किनारे पर आकर रुक जाती है, हमें यह पक्का करने की जरूरत है कि कार अपनी starting position में वापस आ जाए, यानी कार को खुद ही स्क्रीन के left किनारे पर वापस जाना चाहिए। variable सेक्शन for this sprite only के लिय एक और variable 'End pos X' बनाएँ। set End pos X to zero को खींचें और इसे set initial pos Y to minus 150 और go to X and Y के बीच रखें और End pos X की value 280 सेट करें। control सेक्शन if then else block को खींचें और इसे repeat until block के अंदर रखें। change X by 5 block को condition के else पार्ट के अंदर ले कर आयें। ऑपरेटर सेक्शन greater than block खींचें और इसे हेक्सागोनल स्लॉट के अंदर रखें। मोशन सेक्शन में जायें X position block को खींचें और इसे greater than condition के लेफ्ट हैंड साइड पर रखें और variable सेक्शन से End pos X को खींचें और इसे ऑपरेटर के राइट हैंड साइड पर रखें। motion सेक्शन से set X to block को खींचें और if loop के अंदर रखें, variable सेक्शन से initial pos X block को set X to block के number की जगह पर रखें। आइए हम logic को एक बार याद कर लें। हमने यह पक्का किया है कि अगर x axis पर car की position End pos X से ज्यादा हो जाती है, तो car initial pos X पर वापस जाएगी नहीं तो कार x axis पर 5 की स्पीड से चलती रहेगी। अब कोडिंग को टेस्ट करते हैं और यह देखते हैं कि क्या car जैसी उम्मीद थी वैसा ही काम कर रही है? wow, हमने अपना पहला टास्क पूरा कर लिया है। अगले विडियो में हम ज़हरीले gas bubbles बनायेंगे और उन्हें फोड़ेंगे।

bye bye। जल्द ही फिर मिलेंगे।

