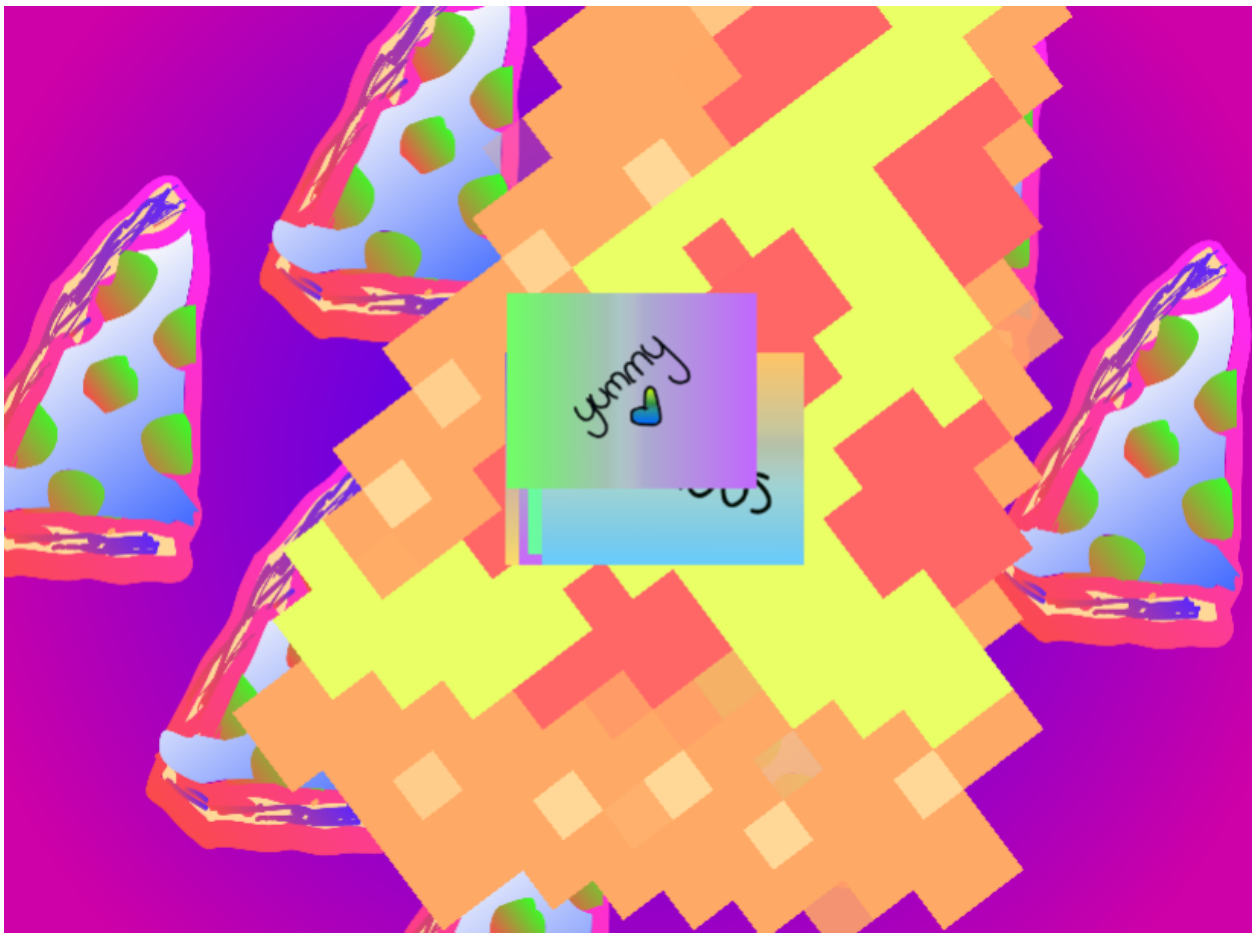
Pizza Bits Expressions



Emily Virginia Ramirez Delgado, Idaly Gonzalez

evr353, idg387

https://scratch.mit.edu/projects/479893441

Table of Contents

Instructions

Welcome to Pizza Bits Expressions! To start the clones you click on the “green flag,” which is located on the top left-hand corner next to the “red octagon.” Once you do that, the sprites will begin to move to different places and to look differently. The backdrops will randomly change to different colors. And a soundtrack will play in the background while the clones are acting. There will be 6 different soundtracks, but each soundtrack will play twice, with a “forever loop.” If you want to stop the clones and the soundtrack, then click on the “red octagon” on the top left-hand corner next to the “green flag.” Also, if you want new information numbers on the list, you may click the “green flag” which is located on the top left-hand corner next to the “red octagon.” Also, if you click on the “green flag,” the soundtrack will start to play again from the beginning. Hope you enjoy it and have a marvelous day!

Sprites

- Pizza 1
- Located in the Pizza Sprite
- Acts to be a Pixelate Pizza with a random pick from the list.
- Will appear on the stage with a random pick from the list or the soundtrack of the beat.

- Pizza 2
- Located in the Pizza Sprite
- Acts to change Pizza's color with a random pick from the list.
- Will appear on the stage with a random pick from the list or the soundtrack of the beat.

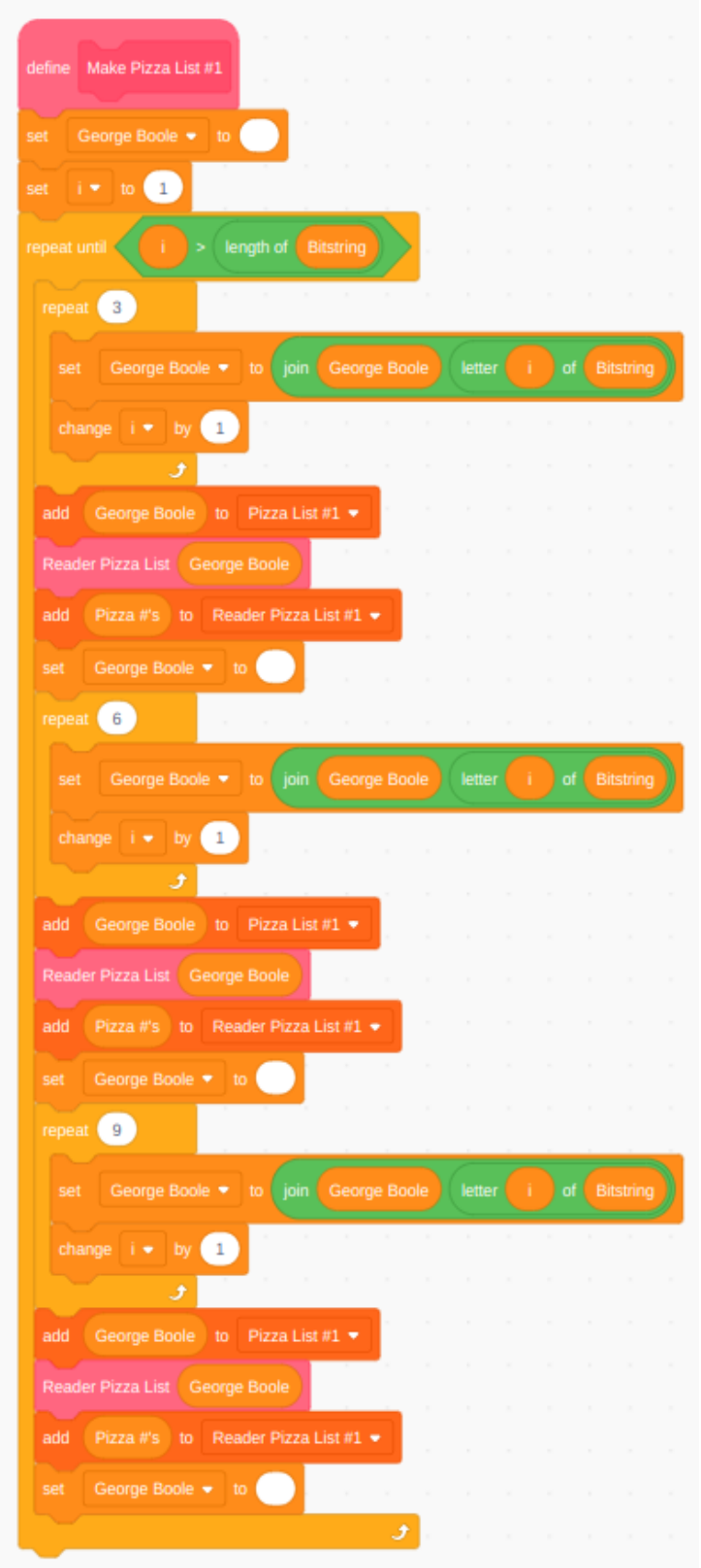
- Pizza 3
- Located in the Pizza Sprite
- Acts to move as a Mosaic Pizza with a random pick from the list.
- Will appear on the stage with a random pick from the list or the soundtrack of the beat.

- Yummy <3
- Located in the Expressions Sprite
- Acts to change the background color with a random pick from the list.
- Will appear on the stage with a random pick from the list or the soundtrack of the beat.

- Delicious
- Located in the Expression Sprite
- Acts to change the background color with a random pick from the list.
- Will appear on the stage with a random pick from the list or the soundtrack of the beat.

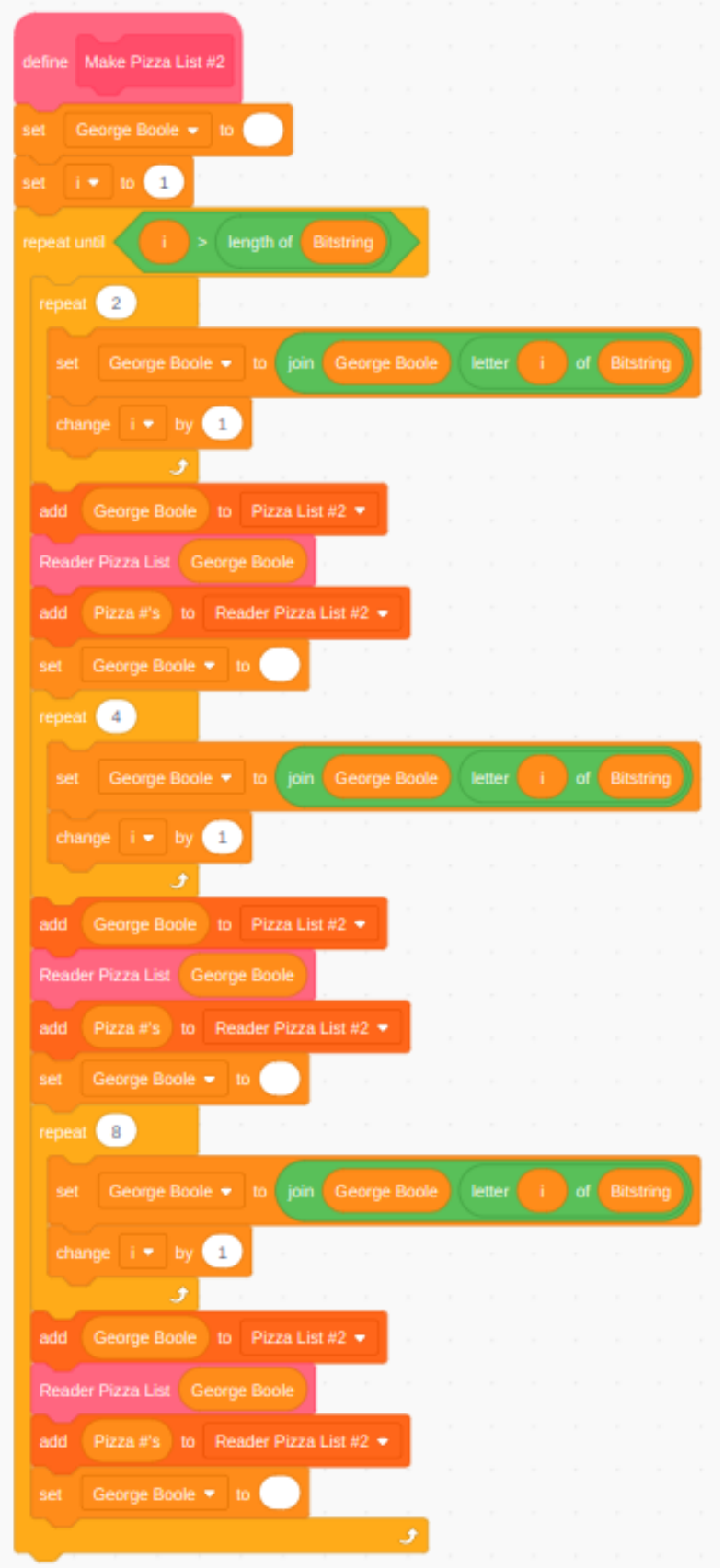
Pizza List #1

Once you click on the "green flag", on the screen "Pizza List #1" will appear. For the starting code for "Pizza List #1," it will begin as "define Make Pizza List #1". Under the "define Make Pizza List #1" it will start the code "set as "George Boole" to 0" this block is a shortcut of 0's and 1's. There will be another code to "set as "i" by 1" meaning the variable shortcut of keeping track of what row you are on in the list. It will have a "repeat until" so it can run all the blocks inside but in a certain order. On the inside side of the "repeat until" it will have a variable "i" "greater than the length of the Bitstring". Which the "Bitstring" is another word to say a list of 0's and 1's but there are only 2,000 of them. First of all, "Pizza List #1" has 336 rows in one list. Which "repeats 3 times" as "George Boole (0's and 1's)." That one row will have a random number of 0's and 1's from the "Bitstring" list. That is how it adds it to 0's and 1's in the "Pizza List #1". Which you can see that I added repeat 3, 6, and 9. Well, it's the same thing, but the difference is that one row will have 3 bits from the "Bitstring". The next row will have 6 bits from the "Bitstring". And the last row will have 9 bits from the "Bitstring". It will continue to repeat as 3 bits, next 6 bits, next 9 bits until the 336 rows are finished and are full of bits. And that is how "Pizza List #1" explains the ways of coding.

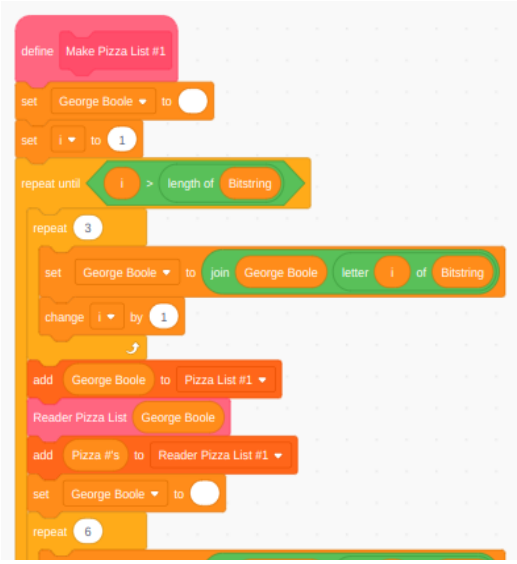


Pizza List #2

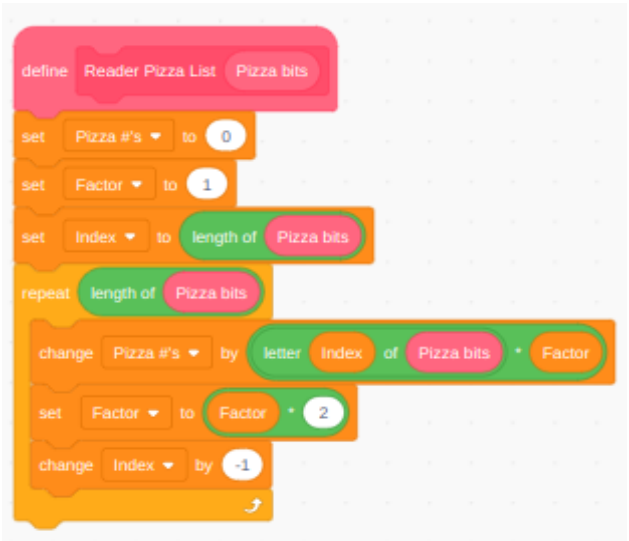
Once you click on the "green flag," on the screen "Pizza List #2" will appear. For the starting code for "Pizza List #2," it will begin as "define Make Pizza List #2". Under the "define Make Pizza List #2" it will start the code "set as "George Boole" to 0" this block is a shortcut of 0's and 1's. There will be another code to "set as "i" by 1" meaning the variable shortcut of keeping track of what row you are on in the list. It will have a "repeat until" so it can run all the blocks inside but in a certain order. On the inside side of the "repeat until" it will have a variable "i" "greater than the length of the Bitstring". Which the "Bitstring" is another word to say a list of 0's and 1's but there are only 2,000 of them. First of all, "Pizza List #2" has 429 rows in one list. Which "repeats 3 times" as "George Boole (0's and 1's)." That one row will have a random number of 0's and 1's from the "Bitstring" list. That is how it adds it to 0's and 1's in the "Pizza List #2". Which you can see that I added repeat 2, 4, and 8. Well, it's the same thing, but the difference is that one row will have 2 bits from the "Bitstring". The next row will have 6 bits from the "Bitstring". And the last row will have 8 bits from the "Bitstring". It will continue to repeat as 2 bits, next 3 bits, next 8 bits until the 429 rows are finished and are full of bits. And that is how "Pizza List #2" explains the ways of coding.



Raeder Pizza List #1

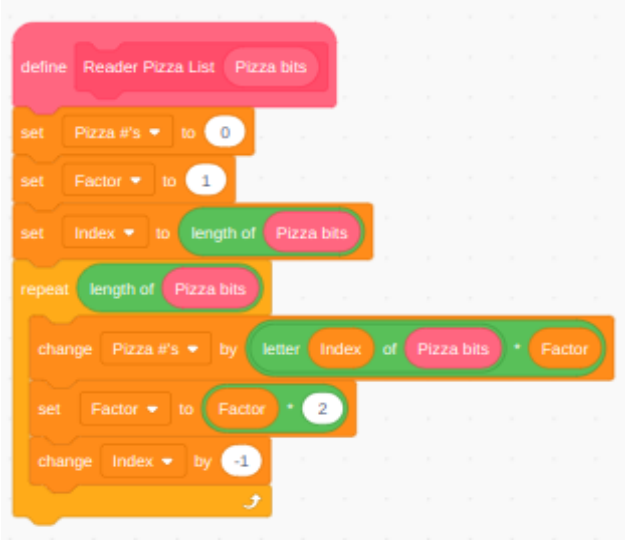


Once you click on the "green flag," on the screen "Reader Pizza List #1" will appear. For the starting code for "Reader Pizza List #1", it will begin as "define Make Pizza List #1". It will then start the code like as we explain on "Pizza List #1". Continuing explaining, like we said that "Pizza List #1" has 336 rows in one list. Well, also the "Reader Pizza List #1" has 336 rows. It will continue to repeat as 3 bits, next 6 bits, next 9 bits until the 336 rows are finished and are full of bits. But we can see that we added a block called "Reader Pizza List." Let's explain why the "Reader Pizza List" helps us convert bits by adding the 1's and converting them as whole numbers. Well, the "Reader Pizza List" of "Make a block" is "defined Reader Pizza List." On the side of it has a variable called, "Pizza Bits." The first block of information in the "defined Reader Pizza List" would be "set Pizza #'s to 0" meaning it will help the computer read, so it can convert and add the bits into whole numbers. The next one is the "set Factor to 1" which divides the whole number to 1 so it can double-check if the computer read the bits correctly. Then there is a "set Index" which another word for its variable "i". Next is the, "set Index to the length of Pizza bits" another word for "Pizza bits" is "Reader Pizza List." This helps the computer read the "Index" to the "Pizza List #1" of converting the bits to numbers, and placing those numbers to "Reader Pizza List #1." The bits there has a "repeat to the length of Pizza bits" because this understands how many times it should be repeated. Because it depends on how many bits, and rows you have in the list. It can be counted many times you want without you creating multiple lists. In the "repeat length of Pizza Bits", in it, it has a block called, "change Pizza #'s by letter Index of Pizza bits times Factor." Meaning it will change numbers by just grabbing the bits from row to row in the "Pizza List #1" to the "defined Reader Pizza List." And converting bits to numbers. The next block has the "set Factor to factor times 2." The "factor" is 1 but it will times it by 2. And now the "factor" is a division of 2. Once, that is done, now it converted the bits to numbers, then it will be divided by 2. And the last block in the "define Reader Pizza List" is called, "change Index by -1." That helps to covert the bits of the negative number to convert it to a positive whole number to, "Reader Pizza List #1." Now I look back to the "define Make Pizza List #1," I can see that we added, "add Pizza #'s to Reader Pizza List." It means that it sends it to the new List of "Reader Pizza List #1." And that's how "Reader Pizza List #1" is done.



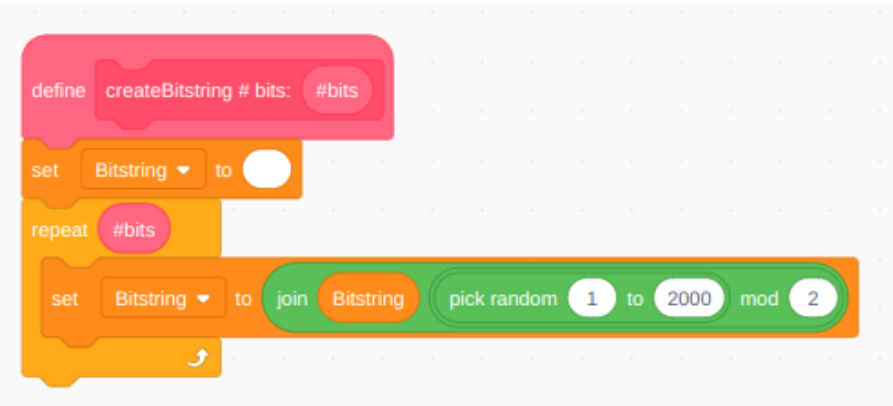
Reader Pizza List 2

Once you click on the "green flag," on the screen "Reader Pizza List #2" will appear. For the starting code for "Reader Pizza List #2", it will begin as "define Make Pizza List #2". It will then start the code like as we explain on "Pizza List #2". Continuing explaining, like we said that "Pizza List #2" has 429 rows in one list. Well, also the "Reader Pizza List #2" has 429 rows. It will continue to repeat as 2 bits, next 4 bits, next 8 bits until the 429 rows are finished and are full of bits. But we can see that we added a block called "Reader Pizza List." Let's explain why the "Reader Pizza List" helps us convert bits by adding the 1's and converting them as whole numbers. Well, the "Reader Pizza List" of "Make a block" is "defined Reader Pizza List." On the side of it has a variable called, "Pizza Bits." The first block of information in the "defined Reader Pizza List" would be "set Pizza #'s to 0" meaning it will help the computer read, so it can convert and add the bits into whole numbers. The next one is the "set Factor to 1" which divides the whole number to 1 so it can double-check if the computer read the bits correctly. Then there is a "set Index" which another word for its variable "i". Next is the, "set Index to the length of Pizza bits" another word for "Pizza bits" is "Reader Pizza List." This helps the computer read the "Index" to the "Pizza List #2" of converting the bits to numbers, and placing those numbers to "Reader Pizza List #2." The bits there has a "repeat to the length of Pizza bits" because this understands how many times it should be repeated. Because it depends on how many bits, and rows you have in the list. It can be counted many times you want without you creating multiple lists. In the "repeat length of Pizza Bits", in it, it has a block called, "change Pizza #'s by letter Index of Pizza bits times Factor." Meaning it will change numbers by just grabbing the bits from row to row in the "Pizza List #2" to the "defined Reader Pizza List." And converting bits to numbers. The next block has the "set Factor to factor times 2." The "factor" is 1 but it will times it by 2. And now the "factor" is a division of 2. Once, that is done, now it converted the bits to numbers, then it will be divided by 2. And the last block in the "define Reader Pizza List" is called, "change Index by -1." That helps to covert the bits of the negative number to convert it to a positive whole number to, "Reader Pizza List #2." Now I look back to the "define Make Pizza List #2," I can see that we added, "add Pizza #'s to Reader Pizza List." It means that it sends it to the new List of "Reader Pizza List #2." And that's how "Reader Pizza List #2" is done.



Bitstring

"Bitstring" is a list of bits, bits of 0's and 1's. The number of bits that can have "2,000 bits" in the "Bitstring." But how does the "Bitstring" work to get that certain amount of bits it has? First, when you click the "green flag" you can see the list on your screen. But the code under the when "green flag clicked" you can see a block under called, "createBitstring # bits: 2000." Well, the "2,000" was our obligation to put the numbers of bits we wanted, so we picked "2,000." Are block "createBitstring #bits: 2000" is taking us to the code "define createBitstring # bits: #bits." That code is first "set Bitstring to blank," but that blank is the meaning of 0's and 1's of its own for that specific code. Now it says, "repeat #bits" meaning "2000 bits" for only, it cannot go beyond "2,000 bits." It repeats so that every time that you click on the "green flag" it will change the bits, but it won't go beyond "2,000." In the "repeat" block, there is a "set Bitstring to join Bitstring pick random 1 to 2000 mod 2." That means it's set to the "Bitstring" list, and it will join in as the number of 0's and 1's. It has a "random pick" when starting this program all the time when you click on the "green flag." And "mod" is when it adds up the 1's from the bits of the "List Pizza #1 and List Pizza #2," and get that total number in the "binary" form numbers 0's and 1's. Once the computer has the total numbers it will divide by 2 and it adds it up in the "Reader Pizza List #1 and Reader Pizza List #2." And that is the coding of a "Bitstring."



Soundtrack

_____The code of the “green flag” in the last block is “broadcast showmeyourstuff”. Which activates to jump to the other codes called, “when it receive showmeyourstuff,” such as the soundtrack. Once it jumps to there, it first “hides” the soundtrack but you can still hear the soundtrack playing. In this code, it has “6 soundtracks” in total. Each of them “repeats twice but forever.” For example, the first soundtrack is called, “Dance Around.” Which “repeats 2 times” the soundtrack will “play twice” but then play again “forever.” It will wait for 1 second for the soundtrack to start, and to have the “clear graphic effects” to clear the stage. Then it comes in the “create a clone of myself” which when it adds the Pizza’s and expressions to show up on your screen. Now the soundtrack comes in its way to give the audience good vibes, and cheer just by looking at the background and hearing the soundtrack. Also, the soundtrack of the beat can make the clones move or appear in different directions. Once the soundtrack is done, it will “clear graphic effects” so it won’t freeze and it will not be too much for the stage to handle. So it repeats again but gives it 1 more second. So it will give the “clear graphic effects” time to handle the cleaning of the stage. That’s soundtrack reader code.

Movement

You may be wondering, how the sprites cause it to move, I just click on the “green flag.” Well, every time you click that “green flag,” it will activate a signal to the other codes. As you can see the block on the left-hand corner, you can see a block called, “when green flag clicked” under there, there are a bunch of blocks of codes. Every time you click on the “green flag” it will reactivate the codes but have a certain specific way. You can’t see a block that says, “broadcast clone or any name” for that clone. But you can see on the next screenshot, where the block says, “when I start as a clone”. The clones are the sprites, and the sprites are the clones. So it says it will show on the stage. Therefore, it has a “forever,” which indicates the clone to move while the sound is playing and what the list has chosen to where to move. Now there is an “If then else,” indicates and the movements of the clones. Between the “If then,” it has a “pick random 1 to 2 equals to 1.” That means it's between the selections of “else.” I know there are 3 of the same “If then pick random 1 to 2 equals 1.” But it fits perfectly to the question to answer the four answers in the “else statement.” In the first statement of “else” is “set Pizza Walk to item pick random 1 to the length of Reader Pizza List #2 of Reader Pizza List #2 plus negative 2.” That means it is set as for the x-axis. It will have a “random pick to 1 to from the list Reader Pizza List #2, and it adds up to negative 2,” but that means it can go to the negative side of the x-axis. In the second statement of “else” is “set Pizza Walk to item pick random 1 to the length of Reader Pizza List #2 of Reader Pizza List #2 plus positive 2.” That means it is set as for the x-axis. It will have a “random pick to 1 from the list Reader Pizza List #2, and it adds up to a positive 2,” but that means it can go to the positive side of the x-axis. In the third statement of “else” is “set Pizza Wish” to item “pick random 1 to the length of Reader Pizza List #2 of Reader Pizza List #2 plus negative 2.” That means it is set as for the y- axis. It will have a “random number to pick 1 to from the list Reader Pizza List #2, and it adds up to negative 2,” but means that it can go to the negative side of the y- axis. In the fourth statement of “else” is “set Pizza Wish” to “item pick random 1 to the length of Reader Pizza List #2 of Reader Pizza List #2 plus positive 2.” That means it is set as for the y- axis. It will have a “random pick to 1 from the list Reader Pizza List #2, and it adds up to positive 2,” but means that it can go to the positive side of the y- axis. And so that gives the sprites to get its movement on the stage. Also, you have to know that your stage contains X and Y- axis, the X and Y- axis won’t show but you have to add in the code. So the sprites or the clone can appear and make their move. That’s the explaining code for movement.

Pizza's 1, 2, and 3

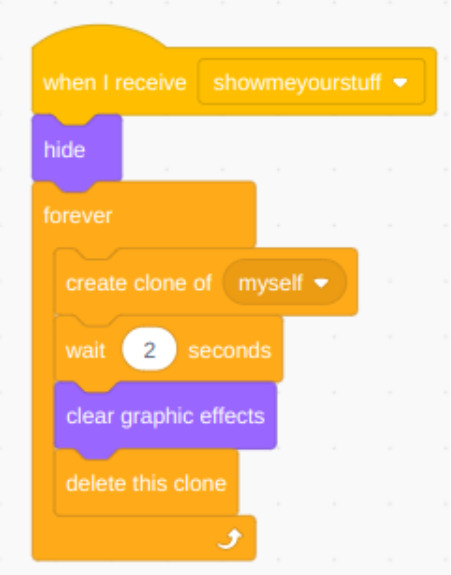
For this, we all know that it first starts as a, "when I start as a clone". Then the next block is "show", which shows the clones or the sprites. It has a "forever loop" and also has the steps of the movements as we explained previously. Alright, there are 3 different types of pizza costumes. We are going to explain "Pizza 1, then Pizza 2, and then Pizza 3." The first block says, "switch costume to Pizzzaaaa1". That is in the first block there because it has the plan information for only that costume specifically, that is why the "switch costume to Pizzzaaaa1" is the first one. Then the next block is "change pixelate effect by pick random item 1 of Reader Pizza List #1 to the length of Reader Pizza List #2." Meaning instead of the normal costumes, it will also become "pixels" like 0's and 1's but in square form. "Pixels" is like how it makes humans eyes see it as a square form, but how it is showing us in its form of computer style. "Pixels" are like zeros and ones but places as squares in images. So, "costume Pizza 1" will pop out as "pixels" meaning squares. Then it has a "pick random item 1 of Reader Pizza List #1 to a length of Reader Pizza List #2," meaning it allows to grab any numbers from "Reader Pizza List #1" to "change the pixels" to bigger or fewer "pixels" on the "costume." Now it has a "switch backdrop to random backdrops." Which leads the "backdrop" to change to any different "backgrounds" that we created in the "backdrop." The "backdrop" can change with the beat of the soundtrack. There is another block called, "glide 3 secs to x: Pizza Walk y: Pizza Wish," which allows the "sprite, costume, or the clone" to move to any position that you allow it to move on that stage. The "glide 3 secs" means how fast or slow I want my costume to move? We pick 3 for the best move. Underneath the "glide 3 secs to x: Pizza Walk y: Pizza Wish," is another block called," "clear graphic effects." "Clear graphic effects" means that it "clears the costumes off the stage" so it won't go crazy with just the same "costume." Also, this helps to give a chance to the other costumes to appear on the stage. And that is for a "costume one", all of the costumes are the same thing but the difference is only on the, "change effect by, glide # secs, and costumes" are different, but it has the same code. "Pizzzaaaa1" is the costume, then the "glide is 3 secs," and the "change effect is pixelate." "Pizzzaaaa2" is the costume, then the "glide is 2 secs," and the "change effect color." "Pizzzaaaa3" is the costume, then "glide is 1 secs," and the "change effect is mosaic." Once the "Pizzzaaaa3" is done, it will "wait 5 seconds" to start again. So it can have time to "delete these clones." Meaning other words "clones are costumes," that will clear out the stage so it could start all over again with the "Forever Loop," but they just keep going like a cycle. Giving each costume a chance to shine on the stage. And that is how you do the "Pizzzaaaa1, Pizzzaaaa2, and Pizzzaaaa3" code for this program.

"Pizzzaaaa1"

"Pizzzaaaa2"

"Pizzzaaaa3"

Expressions



"Expression" is another "Sprite" but it is like the "pizza sprite," it has the same acting, but it kinda has a different coding. First of all, I'm going to explain the top left-hand corner to the screenshot, then I will talk about the bottom screenshot total expression coding. We all know when you click on the "green flag" it will directly "broadcast showmeyourstuff" once that is broadcast it will jump to "when I receive showmeyourstuff." Then it would jump to "hide," so that "hides," it, so it won't show the actual leader from the Sprite to the stage. Then it would jump in the "forever Loop," then in the "forever loop," it would have a "create from myself" which they're going to use the costumes in that specific sprite. Then it "waits 2 seconds" so it can let each costume go on and shine itself on to the stage. It then will "clear graphics effects" on its own while the two seconds are done then. "Clear graphic effects" helps to delete the costume so there won't be too many costumes on stage and helps by preventing it from being caused to freeze. That's why I added this code to help make the program look better and to act better. Now for the coding of "expression," we all know it will start as a, "when I start as a clone." And the next block will be "show", then the next block will be the "movement" and then "forever loop" which we covered previously. Like we already know these steps, then we are going to move on to the block called, "switch costume to yummy<3," which is another costume that will appear on the stage. Then the next block called, "go forward 100 layers." We put that because this sprite is just words, and if you put words in the sprite it's hard to see it during the stage. We place the words 100 layers forward, so we can be able to see the words clearly without you forcing your eyesight. Next block we added, "change color effect by pick random item 1 of Reader Pizza List #2 to the length of Reader pizza List #2." This block makes a change in the costume background, which is really nice and very cool. In what ways does the background color change? It changes just by grabbing the "Read Pizza List #2" and make them into colors. The next block has a "glide 2 secs to x: Pizza Walk y: Pizza Wish." Which is how fast and slow you want the costume to move. X and Y it's what allows your costume to move on the stage. Lastly, it will take 2 seconds, so it can give it time to act on the stage. Costume, "yummy<3 and Delicious" have the same code but have different on the "glide # secs, go forward # layer, and costume." And that last block, "delete this clone" is the reason so the stage will not freeze and so there will not be too many clones appearing on the stage, which causes it to freeze. This will repeat this process "forever" but with a specific order and time.

I hope you enjoy **"Pizza Bits Expressions!"**