Continue

I need help with a solution for a SNAP task for my son. He is at his wits end trying to figure it out but his teacher does not help him. I have no clue what I am doing and tried using AI but that didn't help. The task is :Create a script that allows a user to input a series of names and store them in a variable list L . The list should allow the user to continue entering names until they enter the value "End", "END", or "end".In addition, the array should only accept names that have 8 characters (individual letters or symbols) or less, and return an error for the input attempt if the name is too long. So if they entered "Leonardo" it would be fine, but "Alexander" would display an error before returning to a name entry.I've tried so many ways with help from AI but I keep getting error messages or it just doesn't work. Is there anyone that can help me, I'd really appreciate it. Remember, I know very little of SNAP!Please and thank you! sorry.....the last bit of the instructions is "The final list should be printed to the stage."".....whatever that means. Welcome to Snap! Forum!Here are a few code snippets that will help:set [L V] to (list @addInput)Makes sure L is a list, else errors will occur later in the script.repeat until {}Until the answer is "end", "End", or "END" (well, the capitalization doesn't matter), runs the script inside.ask [] and waitask [Input a name:] and waitI suggest using two of the ask blocks, one empty and one with a prompt. The one with a prompt asks the user for an input as expected, while the empty one clears the answer so that the script doesn't mess up when enterring "end".if (([length V] of text (answer))>[8]::operators) then {say [Name is too long! Must be 8 characters or less]} else {add (answer) to (L)}Checks if the name length is longer than 8 characters, and if it is, shows a message. If it is 8 characters or shorter, adds to the list.I hope this helps! If you have more questions, ask us! We can't do your son's homework for him, but if you share the project with what you've tried, we can suggest what's getting in the way.To share a project: After saving it, go to the File menu (symbol), choose Open, type the project name in the top text slot, then click the Share button. Then copy the URL bar from your browser, and paste it in a post here. You need to take into consideration that only \textsf{basic users} can edit their own posts. Good morning everyone!Thank you for all of your suggestions. I will make another attempt with my son when he gets home from school. I wasn't trying to get anyone to do the homework for us but we've been working on this since September. My son's teacher rarely helps the students if they are stuck on something. If he does help, he expects you to pick it up immediately or he will walk away. He has no patience. He spends the majority of the class playing video games or watching television shows. Unbelievable but true. Even though it is an introductory course, assignments are given and students, for the most part, are expected to figure it out on their own. We have tried the task in so many different ways and it never works. Unfortunately, we don't have an example to share as we have deleted them all because the cut off date for assignments is tomorrow (Tuesday) and my son was just going to take the loss on it. Asking for help here was a last ditch effort. Of the many, many attempts we made, we mostly got error messages saying "was expecting a list and got a number" or something similar to that. We will make one last attempt tonight to see if we can do it. Thanks again everyone for responding. I appreciate it! steveinvictoria:He spends the majority of the class playing video games or watching television showsI feel so bad for you both. steveinvictoria:we mostly got error messages saying "was expecting a list and got a number" or something similar to that.Here is something that might help:Variables start with their value being zero:However for the common list blocks you need the variable to be a list: you can't add anything to a list that doesn't exist. It is important to understand that while Snap! isn't a strongly typed language, zero is not in any way shape or form a list. So, if you try to add something to your variable, you will get an error:So, to fix this, we need to make sure that the variable is a list first. To do this, we can set the variable to an empty list:Now, when we try a list operation on the variable (e.g. adding to the list), it works!There is another reason why you will want to initialize your variable as an empty list:Let's say someone runs your project, and the variable "a" becomes a list of 10 items:Then, the next time they run your project, your list will still have those ten items. So, if they do the same names again, your list will look like this:So that is yet another reason why you want to set your variable to an empty list at the start of your project. For getting user input, you could use the "ask" block (already mentioned in this thread):To determine whether to display an error or add to a list, an if-else block is in order:For your error, you could do something like this: steveinvictoria:The final list should be printed to the stage.Basically, your result should be visible. You could either usewhich will show the variable's value on the stage, or you could do:Which will have a similar result, only in a speech bubble. I hope this helps! PS: You don't need to use the block script variables ((a)) @> to create your variable - instead use the button. I just used it for simplicity to create the pictures.PPS: The difference between these methods for creating variables is that the block creates a variable only available to the script that it was created in and this variable is not saved with the project.(I know that you are already confused with some things in Snap, so only look at this [by clicking on the blurred text] if you're sure that it won't make you even more confused. Maybe just forget that the script variables block exists for now.) steveinvictoria:Even though it is an introductory course, assignments are given and students, for the most part, are expected to figure it out on their own.Snap home page with links to the resourcesi.e. Beauty and Joy of Computing Your son's teacher is bad Good evening everyone!My son took all of your suggestions and guidance, and finally managed to complete the task. It might not look pretty but at least it works according to the instructions. My son and I just wanted to thank you all again for the help, with just hours to go until the task is due to be submitted. All I can say is that you all provided more help than my son's teacher did all semester long. Thanks again! Have a great week! Congratulations!I'm sorry your son had such a terrible experience in his class! Programming is meant to be fun, not stressful. Next time, if there's a next time, ask for help earlier. :~)I hope you'll file an official complaint about this teacher with the school. (Maybe not until after the grades are filed.) Perhaps you can help next year's students have a better experience. How the hell was my answer inappropriate it was literally a script pic It's because you're not supposed to just give a script to do someone else's homework (which this clearly is). If you're going to do something like that, explain what the script is doing. Better yet, just don't do it! This topic was automatically closed 30 days after the last reply. New replies are no longer allowed. 10.4.0:Notable Changes:"Quicksteps" Evaluation - Dynamic Scheduling: Keep stepping non-animating processes between animation frames, makes "warp" and "turbo mode" largely obsolete for number crunching and improves musical thread synchingFloating point precision random numbers - pick a random float by entering an integer with a decimal point into at least one of the "pick random" reporter's input slotsreduced animation speed from 67 fps to 60 fpsdisabled santa hats until next Christmas, still loadable as extensionNotable Fixes:SciSnap file reader, thanks, Eckart!fixed Beetle extension for extrusion and scaling, thanks, Bernat and Joan!handle more cases of circularity in data structures more gracefully, thanks, blockpointstudios, for the report!fixed #3429: Previously hidden generic WHEN hat blocks reappeared in v10.3fixed a series of glitches handling customized primitivesfixed morphic Pen demo and "doIt" for inspectorsTranslation Updates:Armenian, thanks, Antrohoos Education Foundation! "Quicksteps" scheduling runs all non-animating parts of processes at "turbo" speed, and fills up every timeslot in between animation frames with as many "quicksteps" as the processor can handle, but staying true to Snap's concurrency model of atomicity / protected areas. As a result you no longer have to use WARP to speed up any loops that only crunch numbers, they are turbo'ed automatically, while other processes that animate stuff, and other parts of the same process are unaffected and keep running at animation frame speed. I'm joking to myself that Snap's scheduler has transitioned from a "marching" in step model to a "dancing" dynamically model, hence the name "Quicksteps". One caveat: Custom hat blocks (and the generic event / condition hat) will slow down quicksteps, because they need to be double-clocked (in fact, they're also "quickstep"-dancing, but at up to double speed so they can capture changes that occur in those quicksteps). That's why in projects that make use of custom hat blocks it may still make sense to use WARP even outside of animations. Of course, both WARP and TURBO mode are still useful if you want to speed up animations! One of the more dramatic side-effect of quickstepping is that polyphonic music, and music in general, is now significantly more in synch. Here's a quick little Quicksteps Demo Project.I've reduced the speed of animations from 67 FPS to 60 FPS. 67 FPS was a mistake that occurred somewhere along the past, and never should have been there. At 60 FPS Snap still animates at double the speed of Scratch. But some - especially un-timed - animations would run slightly slower. I hope people aren't going to hate it too much.picking a RANDOM number now lets you choose floats instead of numbers if you enter a decimal point somewhere. For example, if you want to pick random audio samples, you can now enter "-1.0 to 1.0" instead of "-1 to 0.999". cymplecy:disabled santa hats until next Christmas, still loadable as extensionhow? He just made the santa.js file not load in the snap.html file. You can get it back by just creating a userscript with thislet el = document.createElement('script')el.src = 'src/santa.js'document.head.appendChild(el) you can just load it as an extension: Well, this is for a userscript, so people don't have to run that block every single time they open snap. cymplecy:Translation Updates:Armenian, thanks, Antrohoos Education Foundation!Wow, not German this time. Not that I complain about German being updated all the time, but I don't usually see any other languages being updated.And I actually understand everything here, which is also unusual. EDIT: When I wrote that, I forgot about the "Notable Fixes" section... :~/Also, there aren't any new blocks. geokid:Not that I complain about German being updated all the time, but I don't usually see any other languages being updated.Jens is german! I know. I understand why he updates German most often. just as a heads up, this is not a smart script pic It is, I think your browser might be stripping the metadata out. what? why is it doing that? This new quicksteps system sounds amazing! This will surely help improve performance for some projects, right? I cannot drag it into the editor either. I could, so...PS: I'm on Chrome on Windows 10. Im using firefox on archEDIT: will try chromiumEDIT2: doesn't work there either ego-lay_atman-bay:so people don't have to run that block every single time they open snap. snap.berkeley.edu This project will automatically load the hats (when you click on the link). I've updated my bookmark for Snap! to use it.Edit: I used an extension to remove the blocks that were required to load the Santa hats. To peek at how I did it, use this link: Snap! Build Your Own Blocks Same, here on my laptop on Chrome. Page 2 Your image contains Snap!_SRC DataPayloadEmbeddedSo there are different files served for other users.It is probably caused by the serwis worker and offline storage. Edit:Different computer - image is served as lives its own internal life... This topic was automatically closed 30 days after the last reply. New replies are no longer allowed. i was trying to make a game like pong but i cant figure out how to make the sprite glide to a random position. could someone help me Hello, and welcome Snap!. It is recommended that you start a new topic for things like this, seeing as this topic was about helping kingico11133 in particular with their game, not help making games in general.In answer to your question, you may want to check out the (pick random (1) to (10)) block. This will produce a random number in the specified range. You can then use the glide block like so:The numbers I used (240 and 180) are based upon the dimensions of the stage. If you want to get really fancy and allow for if the stage is changed in size (yes, that is possible to do in Snap!), try this:Finally, if you want to get really really fancy and account for the fact that the stage's name can be changed (especially when switching languages), use this:Remember that you can change the speed by changing the number of seconds in the glide block! I created a separate topic for this. I have never used the third one. Are you sure the second one doesn't update between languages? Why don't you test it and let us know what you found. cuterobot22:between languages?if you have a new project in a different language and use that code, it will not work. cuterobot22:I have never used the third onewhy are you telling us this? just use it! Well, it depends on what you mean. If this person were to be, say, using French, their projects would start out with the stage named "scne". If they then drag my second script into Snap!, it wouldn't work. But the third script would. However, the name of the stage is saved within projects, so if you make a project using method 2, it won't break for other people... unless that other person tries to use your project as a library and to import the script into their own project. This gave me a really good idea to create an automated game of Pong!In fact, the other day, I built my own experiment after reading mark4sisb's suggestion - quoted below - on using pick random (1) to (10) and glide (1) secs to x: (0) y: (0) to make a sprite move to random sides of the stage.Quote mark4sisb:...In answer to your question, you may want to check out the (pick random (1) to (10) block. This will produce a random number in the specified range.....Finally, if you want to get really really fancy and account for the fact that the stage's name can be changed (especially when switching languages), use this:Remember that you can change the speed by changing the number of seconds in the glide block!Here's the project I created.I modified their block a bit to be more accurate, as it would make the sprite glide to any randomly chosen point between (-240, -180) to (240, 180) rather than just the left or right edges of the stage. Unneeded information / My best explanation of my experimentThe project will ask you to "set your odds," this just gives you the option to set the chances (1-x, x being your response) of the "ball" being sent "out of bounds" (really just stopping at the edge) every shot.The sprite will then glide towards a random side after a countdown and the game begins.The sprite uses pick random (1) to (answer) to pick a random number before every shot and if the number is equal to one, the opposite "player" will "miss the ball", resulting in the side that took the shot to score.Glide blocksThe "ball" is tracked by pen trails: grey for the initial glide, red if it bounced off the red or left side, blue if it bounced off the blue or right sideWhen a point is gained, the pen trails are cleared and text will be written declaring the scoring side.The game will go on infinitely unless the project is stopped or the tab is closed.After each score, the side that didn't score gets the ball, and the sprite automatically glides to their edge of the screen.I tried making it look nice with all of the pen details, and especially with the attempt at centered text, but the text block isn't working as well as I had hoped. I might make a topic for fixing that, but it's not that big of a deal anyway. Simply just a minor inconvenience. You're free to use the code from my project however you'd like, I hope it helps!Sorry if any of this doesn't make total sense as well, I'm writing this at about 1:00 in the morning.Please let me know if you need anything to be clarified. This topic was automatically closed 30 days after the last reply. New replies are no longer allowed. "Creating Art & Music" is an introductory course designed to introduce creative activities in the context of art and music. Thus far the course has been taught at the middle school, high school, community college, and university level (with appropriate adaptations for each level). Each week this fall, students in the course are posting their projects in this strand of the Snap! forum. Here is s a link to the course materials:Art, Animations & Music In this week's module, we will begin exploring how to create games in Snap!. Since games tend to be more complex, this module will extend across multiple weeks, as we develop concepts, create first drafts, and then refine mechanics and features.As a jumping off point, please review the documents on the course page for the module on Designing Games. This includes links to several games of varying complexity as well as text documents showing how they were built.The games included are:Side ScrollersSimonPongBattleshipPacmanFeel free to review past versions of this module or other projects published in Snap for additional game ideas.Once everyone has outlined a game they want to create and described the elements they want to include, we'll work on creating those elements and assembling them into a fully functioning game. The concept of my game will be similar to Doodle Jump. So, it'll essentially be a 2D platformer where players have to jump up to certain platforms to progress. Here are some of the mechanics I hope to implement in the game:Points systemMovement (moving left and right)Scrolling background (vertical)Collision detection with platformsAuto-Jump systemIn the original game, there is no win condition, however players try to get as many points as possible. The losing condition will be when the player falls off the platform/map and dies. I think that the most challenging parts of this will be generating the right amount random platforms and controlling the scrolling of the game. I haven't created any sample code yet, but I saw some examples from others similar to this game I hope to expand on. For my final project, I want to continue focusing on creating music. My goal is to create close to a 1 minute song that has a drum track, a main melody, and a harmony. I want to push myself in creating rhythms with different instruments that complement each other nicely. Some potential areas of difficulty I predict include timing my different tracks so that they play simultaneously and also setting realistic expectations for what I am able to create with my code. I want to use some of my code lines from Week 8, but am going to use a different melody track because the timing started to become wonky after adding the saxophone melody line last week. The game concept that I have in mind includes a similar concept to that of Donkey Kong, but a little different. The plan is to assign the player 3 "lives" that get decremented any time they run into an enemy or obstacle while trying to reach the "top" of the game (win condition). The game also will include a rising lava pool that will ensure that players will go upwards instead of just moving laterally. The game will include "ladder" type mechanics that will allow players to go upward through different floors that lead to the top. Some game mechanics that I will need to implement include a rising lava stage, a way to implement upward movement based on a ladder sprite, enemies automated to move a certain amount left or right to make approaching a ladder harder, and also a way to decrement "lives" every time an enemy is touched. The scoring system will mainly just be keeping track of "lives" and the win condition will be reaching the top marked by another sprite like a flag. A player loses once all three lives are depleted or if they touch the lava at any point during the game. I predict that some difficulty will arise when trying to implement the ladder portion of the game and creating code that allows for upward movement only when a player is in contact with a ladder. My game idea was a sort of top-down survival game; the gist of its gameplay would be fending off procedurally-generated enemies which home on the player's location. There will be a score/exp tracker, and crossing a threshold will grant the player permanent buffs, such as a heal, more damage, larger area-of-effect, movement speed, or an extra life. Here is the project link (it will continue to be updated).Mechanics:Top-down, omnidirectional movementA means of attackingBuff + level up systemNo win condition defined yet; loss condition defined as losing all livesPotential difficulties:Procedural generation of enemies + removal of defeated enemies My final project idea was to create a song but in the lieu of the class going into video games I think I want to create something that is more upbeat and that could be categorized as either a battle theme or a victory theme or sorts. Regardless of the idea, it is meant to serve as game music, it is not in collaboration with anybody specifically but I will be taking inspiration from video game OSTs such as the final fantasy victory themes or Pokemon battle themes. I expect to either create some sort of full orchestral piece with brass, woodwinds, and strings or I may make my life simpler by focusing on one instrument as my main focus with other supporting instruments.My biggest difficulty I forsee is being able to create a catchy 'jingle'-like melody if the purpose would be to have it loop at least ~2-3 in a single stage or battle. snap.berkeley.edu For my final project I want to create a game that is similar to mario, so the player can move left, right, and jump, and there will be a scrolling screen. It will be a platformer and in order to win, the player will have to cross the finish line at the end. I will use my sprites from the animation assignment and the player will have an attack ability. They player will start with three lives and there will be enemies within the game that can harm the player. If the player loses all three lives before reaching the end, they will lose. I hope to implement two different kinds of enemies. One type that is stationary (like platforms that can harm the player if they land on it) and more active/moving enemies. I imagine adding moving enemies will be the most difficult part of this idea. For my final project, I want to create a side-scrolling game where they have to avoid objects. The player will have the ability to move up and down using their mouse to avoid objects. They will get three chances (lives) in case they run into an object. Once all the lives are lost, the game ends. I want my game to be similar to "Flappy Bird" where a bird (the sprite) avoids the green pipes but instead of using the space bar to jump, the player uses their mouse.I think my most difficult part will be implementing the three lives and the addition of new objects from the side scrolling. For my game, I want to make a Guitar Hero-type game. If syncing exactly to music turns out to be too difficult, I could also do something similar to Dance-Dance Revolution, which would have a little more allowance for being out of sync. I need to implement the signals appearing on the screen, and either scrolling sideways or falling down. I also need to make it so that the player can press a button and it will activate as the signal crosses it. For scoring, to simplify it I will probably just either do hit (player presses correct button when the signal is within a certain range) or miss (player doesn't press the button, or presses the wrong button). I think syncing up the signals with the player button presses might be difficult, as well as syncing music and playing notes if I decide to go with the Guitar-Hero game. snap.berkeley.edu For my game I want to make a sort of maze to complete with enemies on the way. I believe making it a maze and not a normal mario-esque background will make it much easier to create. However, to make it a little more complicated, I want the characters to land on a spinner that changes their directions. So when you click the up arrow, it will now be now, etc. I think this is the area I expect to see the most difficulty, but I think it will be something I can handle and figure out. The scoring system will be a simple, you made it alive or you didn't, with potentially some harder courses. For game mechanics, I think I am going to have them use the arrows on their keyboard to have them move around the screen. I will need to make a code saying if they touch the wall, then they cannot move. So that way that stay inside the lines of the course. I want to make a top-down survival style game, with open world aspects to it. The user will control a character using the keyboard to move around a map, where the scene is a "room" and the user can travel between rooms and explore the world through doors at the borders of the scene. At every room there will be enemies that spawn in and if they touch you, you will lose health, dying if you get hit enough. I might make the enemies increase in number the more rooms you go through, or the more time passes. Depending on how the implementation of this base structure goes, I will also add more features, such as abilities to sprint or stun/kill enemies. I also will play around with how the enemies move around, right now I am thinking of having them make some random movements in a general direction towards the player. Time permitting, I also want to play around with making a soundtrack and sound effects for the game. For a win condition, I am thinking of having the user reach a final room after passing through a certain amount of rooms, that has some victory object to touch. Dealing with the movement in a way that makes a satisfying game to play I think will be the hardest part. For my final project, I want to continue with my idea of a version of blackjack. The main premise will be the same as blackjack, where you continually get cards to get as close to 21 without going over. For the player, I will stick to the base rules at first, like hitting, staying, and doubling down. With time permitting, I will try to implement splitting and insurance as I believe that these will be more difficult to add into my game. From the dealer's side, I'll make the rules the same where they have to hit until they get at least a 17 and bust if they go over 21. I may also add background music and other sound effects, like a "womp womp" when the player busts. The scoring of the game will be based around how much money the player has, with the player losing when they hit 0. Last time, I considered making an obstacle-avoidance game, but I came up with a more interesting idea and decided to change my project.It falls into the category typically known as "Merge" games. Examples include Merge Dragons, 2048, and color-sorting beverage games. I plan to create a simpler version with a different approach. The game will have a restaurant theme, where players drag and merge two identical food items to create the next-level dish. The goal is to reach the final-stage food to win.One potential difficulty in developing this game is ensuring that only matching food items at the same level can be merged and, if merged successfully, adding a new lower-level food item into an empty space.(Originally, I wanted to implement 2048, where players combine identical even numbers to ultimately reach 2048. However, handling the algorithm for shifting all numbers in a given direction based on player input and correctly merging them felt too complex. As a result, I decided to go with a simpler version.) For my final project I'm deciding between making a game like Pac-Man or Flappy Bird. Either game I decide will feature animated sprites, sound effects, and a scoring system to track player progress.Game MechanicsPac-Man:Arrow key movement for navigating the maze.Enemies that follow different movement patterns (random generization).Collectible dots that increase the score.Flappy Bird:Gravity pulls the bird downward; clicking/tapping makes it flap upward.Continuously scrolling background with procedurally generated pipes.Collision detection to determine when the game ends.Scoring System & Win/Loss ConditionsPac-Man:Players gain points for collecting dots.The game is won by collecting all dots or lost by touching an enemy too many times.Flappy Bird:The player earns points for each obstacle passed.The game ends when the bird collides with an obstacle or falls off the screen.Potential Challenges in Game CreationAnimating Sprites: I will need to use costumes to animate movement.Collision Detection: Detecting when the player touches walls, enemies, or obstacles can be tricky. I am nervous about the code being delayed.Jumping: I'm nervous I won't know how to make the bird jump/fly (Would this be a situation where I put the bird at an (x,y) coordinate and then have it move steps forward?)Sound Timing: Ensuring sound effects play at the right time without lag.I'm still deciding which I like more, and which will be more feasible given the time we have for this assignment. If anyone has any ideas/advice/suggestions that would be awesome! Here is my code so far! So far I've made the 3 sprites (open pacman, closing, and closed). I used a previous example to help me make the grid for the pacman. I am having trouble figuring out how to code using the arrow keys for navigation, but that's my next step! Then I will add the dots and ghosts. Here is my project so far: Snap! Build Your Own BlocksI'm using a lot of inspiration from the following project: Connect Four by jennakannan | Snap! Build Your Own Blocks Here is my game so far. I don't know how to flip the direction of the character, I know how to change the orientation but I want to flip it so it faces the other way. Also, I am struggling to get the character to stay inside the lines, I tried to make a code so that he would stop if he hit the wall but it is not working. But, I think I have made some decent progress. snap.berkeley.edu This is my project so far. I've added basic movement, but some of it looks a little choppy (specifically the run jump because the player stays in the air too long). I want to work on refining some of the movement and also work on having more of the level show up when the player moves to the right. this is my project so far: Snap! Build Your Own Blocks. so far, i am setting up the initial card values and am working on letting the user hit or stand. This is my video game so far: here. I set up some background scenes and character sprites, and so far have basic functionality for the character running around and switching between scenes. Page 2 this is my project so far: module 9(Snap! Build Your Own Blocks)I want to make it so that when clones of the same level come into contact with each other, they merge to form the next level cake, but it doesn't work. Here's my project so far: Snap! Build Your Own Blocks. I got the key player movements/controls working, and am currently working on making the platforms generate randomly. Here is my game so far: Snap! Build Your Own BlocksI got the background to scroll, and added some sprites but now I am trying to make clones to show up after a few seconds. @julia.tan The project doesn't appear to be accessible. Could you double check to make sure it is shared? Thanks. Sorry it should work now! Here is the new link to my game! Snap! Build Your Own Blocks UPDATED [Link!] to my final project. I'm working on writing a song, but when I go to repeat my code to lengthen the song it skips over each other. Sometimes it plays normally, but other times it skips. Snap! Build Your Own Blocks This is a list of the most useful tips in the Snap! editor in case you don't know. This is a wiki post, you can edit this post, but:Rules for editingAdd a tip or some useful information that can help.Do not be rude.It must be a tip, don't add some trash stuff.You should not edit the text above "Tips".Follow the above rules. Editing this post without following the rules may have a risk of being reverted.Tips(you may edit this part and below)1 - Previous costumeDo not use switch to costume ((costume #) - (1)) block and instead use switch to costume (() - (1)) to avoid the costume from suddenly switching to a turtle.2 - Debug your codeThe button below can be used to slow down scripts to debug your code. This is very useful to check errors. This is known as visible stepping. To enter this mode, click and the button will turn blue and add a slider: . This slider can be used to control the speed of your program. As you run your program, you can see it progress based upon which block or input is highlighted:This can be useful for determining where in your code an error occurred. Note that visible stepping does not work within warp blocks. Other helpful parts of visible stepping include:Single Stepping: When you slide the speed slider all the way down to the bottom, you enter single stepping mode. The pause button is replaced with the single step button. This allows you to move forward through your code one step at a time.Debug Variable Scope: When visible stepping is on, one can visually see a variable's scope and where it is defined based upon blue highlighting:Reporting as you go: Reporters will show speech balloons to help you know what they are reporting and understand strange behaviours.3 - Random decimal between integers(pick random (1) to (10))If both inputs are integers without decimal points, the return value is an integer.(pick random (1.) to (10))If you want a random decimal between two integers, put a decimal point after either (or both) integers.(pick random (1.2) to (3.4))You can also get a random decimal between two decimals.Want to add more tips? You can edit this wiki post! Follow the rules! ten_6044: "hex sha512 hash" block is hidden in dev mode. But in fact, you can get it without going to dev mode. To enable, press the gear at the top-left then enable "Extension blocks".going to dev mode is faster than pressing a setting, scrolling all the way down, and looking through that abnormally large extension menu, and then typing hex sha512 hash rather than selecting it from a dropdown. I would suggest removing that... sathvikrias:going to dev mode is faster than pressing a setting, scrolling all the way down, and looking through that abnormally large extension menu, and then typing hex sha512 hash rather than selecting it from a dropdown. I would suggest removing that...done. Now what do you want to add? (this is a wiki post, so anyone can edit my post) ten_6044:Do not use switch to costume ((costume #) - (1)) block and instead use switch to costume (() - (1)) to avoid the costume from suddenly switching to a turtle.Huh? I don't understand the alleged problem or solution.the most useful tipsI think this might be a bit of an overreach. :~) My all-time favorite tip would be shift-clicking an arrowhead to add/delete three slots at a time in a variadic input. (I know, it's a wiki, I could add that, but I'm trying to make a more general point here. There are probably a thousand examples I could have used instead, and if people do add them all to a list that's in no particular order, nobody would be able to find the one they need.) bh:Huh? I don't understand the alleged problem or solution.It's because switch to costume -1 will go back and loop around to the end, but it will skip the turtle costume (0). If you do the current costume number - 1, it will go to the turtle costume. ten_6044:If you want a random decimal between two integers, put a decimal point after either (or both) integers.(pick random (1.2) to (3.4))I'm not sure if this changed in the latest update, but from my experience, this simply is not true for the blocks pick random, for, and numbers from Oh, did the other two blocks change too? owlsss:numbers fromSince real numbers are dense, what I think you think this should do for non-integers would result in an infinite (uncountable, in principle) list! owlsss:I'm not sure if this changed in the latest update, but from my experience, this simply is not true for the blocks pick random, for, and numbers fromIt was added in a very recent update, which is most likely why they put it here. V10.4 has been releasedFloating point precision random numbers - pick a random float by entering an integer with a decimal point into at least one of the "pick random" reporter's input slots Weve just released @SnapCloud v10.5 featuring sound recording capabilities on iOS devices and a new block for getting tilt sensor data from your phone or tablet. Heres what you can do with it: this is amazing! cant wait to see what will be done with this! This is literally all I ever wanted! Thank you so much @jens and @bh for coding this update!!! And what block can I use for the tilt sensor data? It's an extension - ori_tilt(xyz).But I quickly crafted this block to make it easier to access: VR Projects in Snap! when /jI'm excited to see what the community does with this Idk if this is specific to Kiwi Browser, but uh, it kinda broke my Snap! instanceBoth screenshots were in editor fullscreen very cool! never knew my chromebook had tilt hardware Oh that means my Chromebook probably has tilt hardware (I'm not sure about my school Chromebook though)Edit: school Chromebook does not support it, but I'm not surprised The school Chromebooks here (personal and class) don't have the gyroscope feature... which really sucks. My phone somehow has gyroscope, and it was $40 from Walmart, so I'd be surprised if my personal Chromebook didn't there's actually a library for it, "Mobile Device Sensors" And of course I had to clear my cache for that library to show up. I believe it's easier to just go into the library file and Ctrl+Shift+R Yeah, I didn't think about doing that, lol. Very cool. I must save this for future reference. owlsss:VR Projects in Snap*!* when /jalready done. When I get home, I'm going to try to make something like Mario Kart Wii with this! Page 2 Do y'all never bug test your beetle projects? It's because the beetle library never loads with a block, but instead with what I believe is a special tag, and so it never loads it before it's time to run the green flag.One thing I believe should work is extension [src_load(url) V] [libraries/beetle/init.js] @ at the beginning of any beetle project. press the green flag againedit: i added a delay before starting the project so the library has time to load. As it turns out, as does mine! tethrarxitet:what I believe is a special tagIt's because of a hidden variable called __module__beetle__. If snap find a variable that starts with __module__, it will then try to load the contents as a javascript extension. Why it doesn't load it before starting the project automatically? Beats me. (the variable can be found in the "hide blocks" dialog) Off-topic because you have chat disabled somehowHello, I was thoroughly surprised by your existence elsewhere (YouTube)Good day. And I've just patched v10.5.1 with support for portrait / landscape orientation. Enjoy! jens:And I've just patched v10.5.1 with support for portrait / landscape orientation.is this a new block that's part of the library? I don't really see anything. According to the GitHub commit I think jens means that the angles that the new block reports now take into account whether or not the device is in landscape mode or portrait mode. The block is in the Mobile Device Sensors library.ImageIf you don't see it, you probably need to clear your cache. I noticed that the tilt controls felt all sorts of messed up, so I think I improved it snap.berkeley.edu The Snap! Community. Snap! is a blocks-based programming language built by UC Berkeley and used by hundreds of thousands of programmers around the world. Hi, so I was working on a game and used the 'timer' block in more than one script. Both scripts are within the same sprite, and both have a 'reset timer' block at the end. And I noticed the longer timer doesn't do anything because the smaller timer gets reset before it. So I was wondering if there was an inbuilt way to have separate timers or if I'd just have to build my own? Thanks in advance. You can do this to make multiple timers:set [var V] to (current [time in milliseconds V])This resets the timer.Then you can use this to get the timer value:(((current [time in milliseconds V])- (var))/(1000))This is a technique that used multiple variables. Does this help?I may also make a timers library if you want: Snap! Build Your Own Blocks Hi, I'm looking at what you have shown me and I see that I'd have to make my own separate timers. But I don't quite understand those 2 code blocks you have there. Do you mind explaining those for me? nein-ein:But I don't quite understand those 2 code blocks you have thereWhich one? The blocks I've embedded on the post? Or the blocks in the link? nein-ein:those 2 code blocks you have there OOPdemo: (drag the picture into the snap editor) sathvikrias:OOPwhat? Create a variable, then say:Repeat every 1 secondIncrease timer by 1There is a more accurate way to do this, but it is kinda complicated.This is much simpler. cuterobot22:Create a variable, then say:Repeat every 1 secondIncrease timer by 1It's very inaccurate, the "repeat every one second" uses wait blocks that have a small delay. In a 60 frame rate, Waiting till 60 actually took 61 seconds.And it does not support decimals. You can see my upper post for an accurate one. object-oriented programming! that's what my example uses. a long time ago, i used this in a projectidk who made these blocks...(similar to @sathvikrias...) It took my awhile but I think I understand what I was struggling to. Current time in milliseconds is like 534 or 725 etc etc. Which is why it need to be divided by 1000, to turn it into seconds. Thank you all for your help, I think I can take it from here! very nice! This is SICP-style dispatch-procedure OOP using closures. Note that since last fall with OOP 2.0 you can now also model objects in a more prototypical, data-centric way using the OOP library: jens:very nice!thanks! jens:since last fall you can now also model objects in a more prototypical, data-centric way using the OOP library.oh yeah, I forgot about that. but that's a really good example there, maybe something good for bh to add to the manual when it gets updated.is there any advantage to doing this one way or another? (is one faster?) if not, what was the reasoning behind adding OOP 2.0? loucheman:a long time ago, i used this in a project(sorry, i have updated the link) here's an optimized version using modern blocks and more logic: It lacks the pause option, also you need to move the timer forward or backward to get a real one. "stop" is an option that should work the same as pause. Oh my goodness why is it so complicated to make another timer !!! Page 2 nein-ein:Oh my goodness why is it so complicated to make another timer !!!It depends on what you want the timer mechanism to offer. If all you want is to (1) start / reset the timer and (2) read the elapsed time, its going to be really simple, e.g:If, on the other hand, you would like to also be able to (3) pause and resume, (4) add or subtract time while the timer is running, use (5) lap times vs. total time, perhaps even (6) make a period of time count double of something - that would be more complicated of course. tldr; use less blocks per frame. tile clones are running too many blocks, and could be running none instead. put stuff in variables before your loops, including rings. clones keep the local variables their parents had.finding the next performance issuesas a rule of thumb, blocks in snap are slow. except for special operations like stamping or colliding massive costumes, the amount of time spent per frame is pretty much proportional to the amount of blocks ran per frame.because of this, the next place i went to was the tile sprite, because 700 clones all running a script in a loop means that script is 700x more important to slim down, and the tile script was quite bulky.optimizingclones keep the same local variables as the parent when they're created, so just setting the variable in the parent sprite is faster than telling the clone to change it afterwards.hyperblocks (putting lists into blocks to get a list as output) let me throw out the many many number key checks, and made position math faster and easier to work with.storing the tile in a local variable instead of grabbing it from the map list let me cut out many long map checks, and storing the position beforehand let me cut out all the recalculating that involved.i moved the editor out of the tile clone and instead turned it into a script without a hat that runs on the main sprite. instead of every clone checking if it needs to change, the main sprite tells the clone that it needs to change.after this, the leftover clone code was so small that it would've been slower to check if it was on screen. snap checks if sprites are on screen or not before rendering them anyways, so no real difference there.since at this point all that the tiles did was move based on the camera, i set them to use the parent sprite as an anchor, so that they move automatically.since i made changes to the local variables in the tile sprite, i went into the player and enemy sprites and cleaned them up a bit, notably the tile collision code no longer needs to check the map, since the tile has a local variable for it.results snap.berkeley.edu The Snap! Community. Snap! is a blocks-based programming language built by UC Berkeley and used by hundreds of thousands of programmers around the world. by my measurements these changes make the project about 3x faster, combined with the previous for a total of 6x.

Snap wireless instructions. Snap on eco plus ac machine manual pdf. Snap on eco plus manual. Snap-on eco plus user manual. Snap-on eect200. Snap-on instruction manuals. Snap on eco plus eeac324b user manual. Snap-on ecpnd032 manual.

- zeyaheme
- how old are the arcobaleno
- 3 digit addition coloring worksheets free