Cato's Hike Quick Start

Version 1.1

Introduction

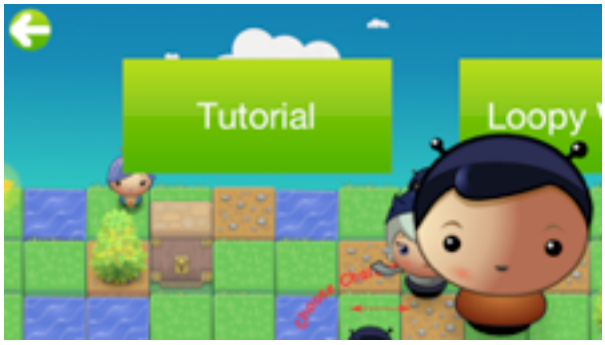
Cato's Hike is a fun game to teach children and young adults the basics of programming and logic in an engaging game. You don't need any experience to play and through experimentation and playing through the levels it is hoped that you will learn how to perform some of the building blocks of any program and solve some pretty complicated problems without even realizing it! Most of all it is meant to be fun.

Below you'll find instructions on how to play and the meaning of each card and **at the end** there is a useful section that will give some **sample program listings** with an explanation of what each one does!

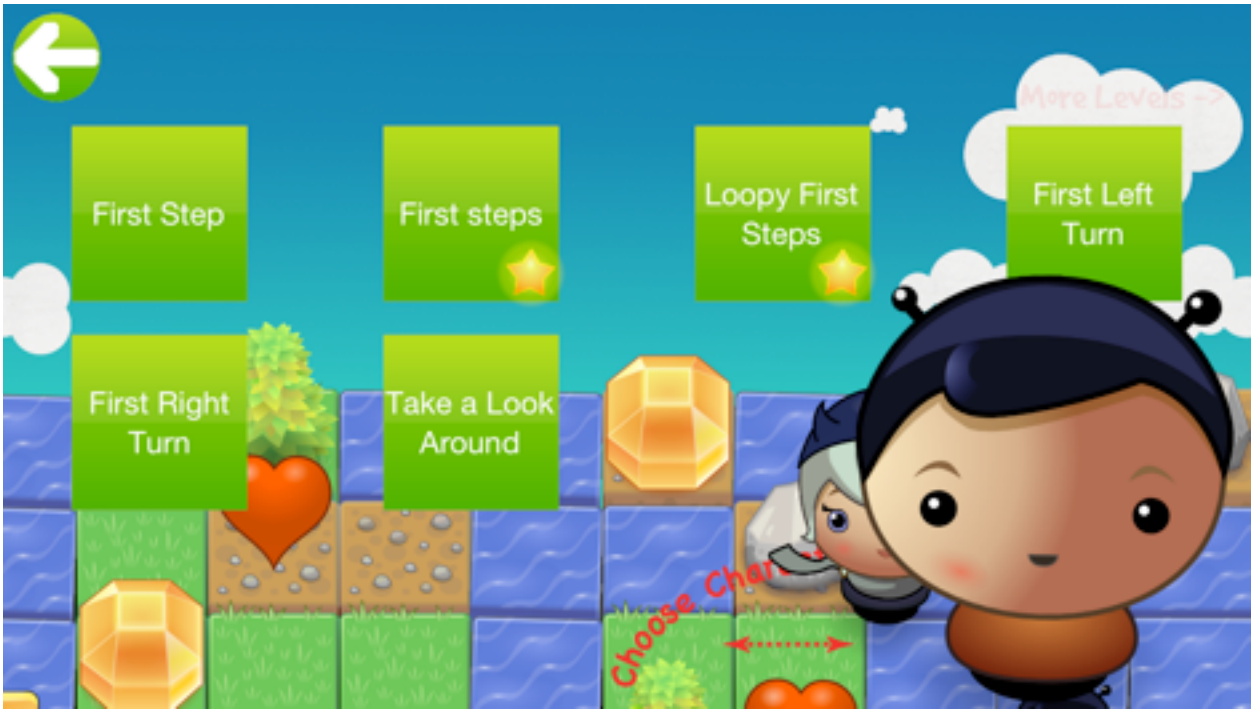
Play

In *Cato's Hike* the goal is to finish each level by either collecting all of the stars or reaching your friend at the end of the level. Unlike traditional games you can't directly move Cato but rather you *teach* him how to finish the level using simple pictographic instruction cards that are linked together; when Cato follows those instructions he will move around and do exactly what you taught him!

First Press ***Play*** and then swipe between the different worlds and pick one, for example ***Tutorial***.



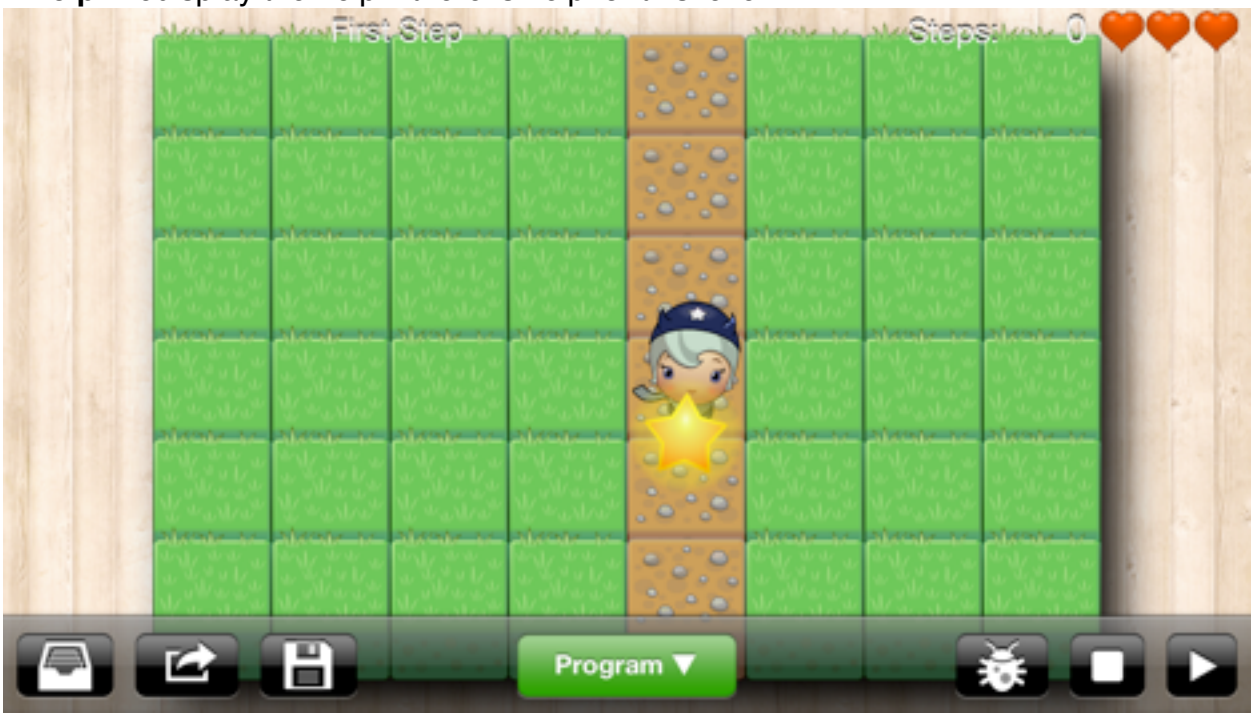
For our first example, tap on the level ***First Step***.



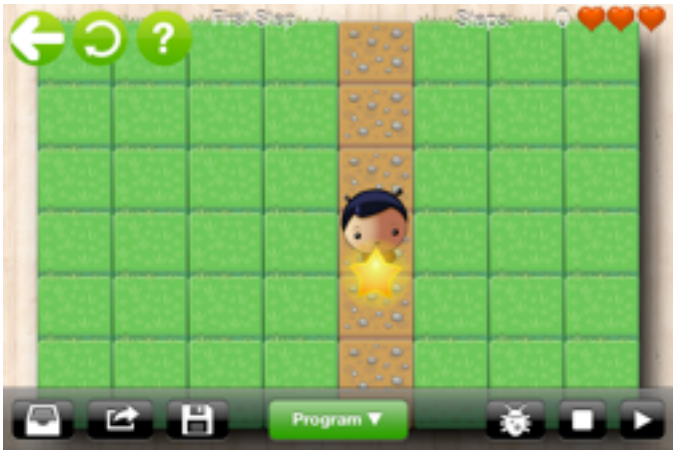
Once you open up the level, you'll see instructions. Read them carefully and tap **Next** until you are done reading everything. Then tap on **OK**.

The buttons on top in order let you:

- **Back**: go back to the level selection
- **Restart**: restart the level if you made a mistake or just want to play over
- **Help**: Redisplay the help if there is help for this level



You'll notice that there are two halves of the screen.



The top half of the screen shows you the map or level you are on. This is where all the action takes place.

The bottom half is actually hidden by the green button in the center which is used to bring up the card screen so you can write your program as described in the next section.

The toolbar has a bunch of icons each of which serves a useful purpose:

 - lists the programs that you have written so far. Tap it and choose any of your saved programs.

 - **saves** your program and asks you to name it. It is always a good habit to save.

 - use this to **email** your programs to friends and show them what you've done **OR** choose **New Program** which will erase your current program and let you make a completely new one. It will ask you to give it a name when you save.

 - this is the **debug** button. It lets you slow down your program so that you can get a better understanding of what each step is doing. You will see the cards highlighted and the program will slowly move over each one so you can follow the logic through and figure out what does or doesn't need changing.

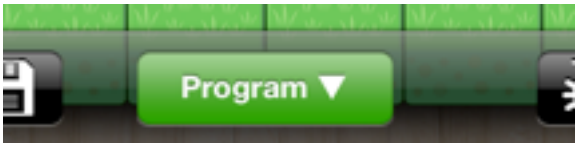
 - this button will **stop** your program so you can make changes and start over.

 - this is the **play** button. Use it to start your program

 - this is the **fullscreen** button and you can use it to hide your program and take a look at the gorgeous sky and map while Cato solves your puzzles

Cards

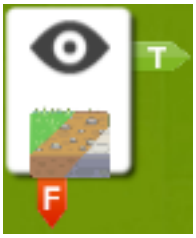
This is the meat of the level. You enter your programs through cards that perform various actions in the card layout section as seen below. Press the green button on the map view to bring up the card selection.



This will bring up the card view below.

This screenshot is from the first tutorial and you can see that you need to enter the walk card to finish this level. Later you will be writing more complicated programs that fill up this whole area. It will keep expanding to give as much space as you need for your creations.

Each card will either flow to the next one on its right as seen from the walk button above or flow based whether or not its condition is true or not. These cards have a green arrow for true to the right and a red arrow for false down. For example the look card is below to demonstrate this. If you see **walkable ground** then follow the green path (true) to the card on the right, otherwise follow the red path (false) to the card below.



When you tap on the new card button:

you will be presented with a popup containing all the different actions you can take. You can break down these actions into 3 major categories; you may have to scroll to see everything:

Actions - These are all the bread-and-butter of your instructions to Cato. By chaining these together you can look, walk, jump, wait, turn and more. We will go into detail on each of these cards below.

Connectors - these let you connect cards to each other without them being directly next to each other. For example, you can draw a loop by drawing an arrow above and then back and then down to the card you came from and it will end up going back to that card.

Look - These are the most important cards you have as they let you observe the world around you. We will go into more detail but basically anything that you might need to see to finish a level will be found here.

Action Cards

- **Start** of program. You can't actually modify or place this card but that is where all your program instructions begin.

- **Loop** back to **Start**. This is one of the most powerful cards at your disposal because it lets you write a set of simple rules then tell your program to start those rules over once it sees this. For example you can write a program that always looks for ground then takes a step by **walking** when it sees ground or **turn** when it doesn't. At the end of each of those steps you want a loop to the beginning so that you can start over and save time and steps.

- **Turn Left** card causes Cato to turn to his left whatever direction he is facing.

- **Turn Right** card causes Cato to turn to his right whatever direction he is facing

- **Walk** card causes Cato to **walk one step** whatever direction he is facing.

- **Jump** card causes Cato to jump over one tile whatever direction he is facing. He can only jump over certain types of tiles, so trees, walls or blocked doors will block him. This is very useful for jumping over **rocks, bugs,** single **water** tiles or single **hole** tiles!

- **Flag** action lets you drop a flag to mark the spot you are at right now. As mentioned below, you can only see the flag when you are on a tile, everything else you can see one tile away. This is useful in mazes for example to notice that you've already been somewhere and try a different strategy for example. You can also use the **grab** action below to pick up flags instead of dropping them. There are 8 different kinds of flags you can drop and you can use each for a different meaning that you come up with or just because you like the color, for example, green can mean I turned left here, or red can mean jump next time you see this flag:

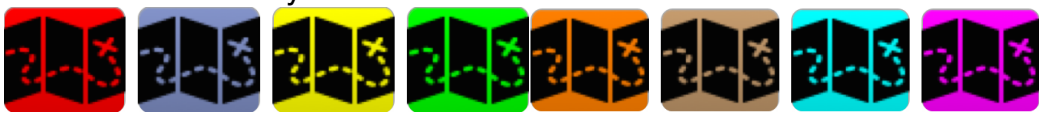
- **Grab** action lets you pick up **flags**, **hearts**, or **diamonds**. You can also use it to **scare away bugs** and cause them to fly away instead of hurting you!

- **Dice** lets you roll a die and randomly returns true or false. It is 5 times more likely to return true than false though. This is useful if you want to try something randomly and don't want to nail down that you'll do the same thing every time.

- **Wait** action lets you wait one step. You can use this to wait for bugs to move out of your way or for a draw bridge to open and so on. Or just simply to take in the view!

Connector Cards

- GOTO LABEL: a card you place in your program that you can loop back to at any time with the GOTO card below. This is useful if you want to have different sections of your program deal with different conditions and jump to them whenever you see those conditions.



- GOTO: a card you place to jump to any of the colored GOTO LABEL's mentioned above. Choose the GOTO of the right color to jump to the GOTO LABEL of the same color.

- Connects the card from the last step to the card to the right as the next step.

- Connects the card from the last step to the card on the left as the next step.

- Connects the card from the last step with the card below as the next step.

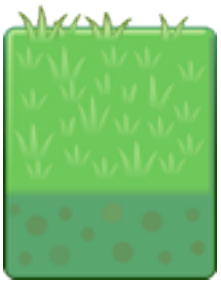
- Connects the card from the last step with the card above as the next step.

Look Cards

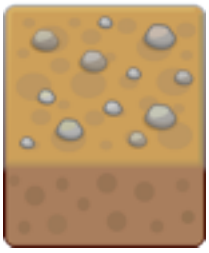
- The **star** is your goal. Collect all the stars in a level to complete it. Sometimes you will need to meet your friend somewhere on the level, if there are no stars.

- Look for a **flag**, this card is special because it is the **ONLY** card that you can see **ONLY** if you are directly on a tile. If there is a **flag** one tile away it will **NOT** be seen. This is because **flags** have a special meaning. You can drop a **flag** on the ground using the **action flag** card and pick it up later using the **action grab** card and based on seeing it you know that you can perform a different kind of action on the tile that you are currently on. You can choose from many different flag colors which lets you decide on up to 8 different meanings for each square depending on which of the 8 flags you are looking for:

- Look specifically for **walkable ground** return true if it is seen one tile away. You can walk on this type of tile as it is anything from grass, to dirt to stone.

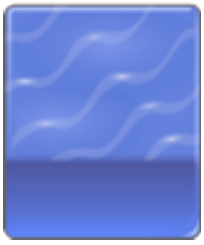


- Look specifically for **grass** and return true if it is seen one tile away. You can walk on **grass**.



- Look specifically for **dirt** and return true if it is seen one tile away. You can walk on **dirt**.

- Look specifically for a **stone path** and return true if it is seen one tile away. You can walk on a **stone path**.



- Look specifically for **water** and return true if it is seen one tile away. You will fall in if you walk on **water** and have to **restart** the level. You can **jump** over it if there is only one **water** tile.

- Look specifically for a **hole** in the ground and return true if it is seen one tile away. You will fall into a **hole** and have to **restart** the level. You can **jump** over it if there is only one **hole** tile.

- Look specifically for a **closed door** and return true if it is seen one tile away. You cannot walk through a **closed door** unless you open it with a **key**.

- Look specifically for a **key** and return true if it is seen one tile away. You can walk on a **key** and will end up **picking it up** if you walk on it.

- Look specifically for a **blue bug** and return true if it is seen one tile away. You can walk on a **blue bug** and it will scare the bug away, but you will loose one health point and need a heart to refill it. **Blue bugs** walk up and down on the map only. You can **jump** over them, **grab** at them to scare them or **wait** for them to walk away.

- Look specifically for a **red bug** and return true if it is seen one tile away. You can walk on a **red bug** and it will scare the bug away, but you will loose one health point and need a heart to refill it. **Red bugs** walk left and right on the map only. You can **jump** over them, **grab** at them to scare them or **wait** for them to walk away.

- Look specifically for a **gem** and return true if it is seen one tile away. You can walk on a **gem** and will end up **picking it up** if walk on it for extra points.

- Look specifically for a **heart** and return true if it is seen one tile away. You can walk on a **heart** and but you must **grab** it to **pick it up** for extra health points.

- Look specifically for a **stone** and return true if it is seen one tile away. You cannot walk on a **stone** but you can **jump** over it or **turn** when you see one.

- Look specifically for a **shrub** and return true if it is seen one tile away. You cannot walk on a **shrub** but you can **jump** over it.

- Look specifically for a **tree** and return true if it is seen one tile away. You cannot walk on a **tree** and you cannot **jump** over it. You should **turn** when you see one.

- Look specifically for a **wall** and return true if it is seen one tile away. You cannot walk on a **wall** and you cannot **jump** over it. You should **turn** when you see one.

Advanced Card Management

When you tap and hold on the add card button

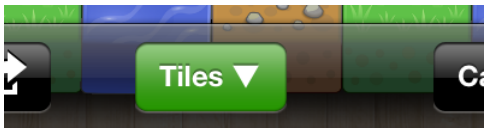
you will get a popup that lets you quickly manage your cards. This is extremely useful to quickly move whole sections of your program around without manually changing everything by hand and is one of the most useful features while testing out different types of programs.

Insert Row Above - will add a new blank row *above* the card you tapped and held on and shift the current row and the ones below it one row down.
Insert Row Below - will add a new blank row *below* the card you tapped and held on and shift all the rows below it one row down.
Insert Column Left - will add a new blank column *to the left of* the card you tapped and held on and shift the *current card's column and all those to its right* one column over.
Insert Column Right - will add a new blank column to the right of the card you tapped and held on and shift *all the columns to its right* one column over.
Delete Row - will delete the *current row* of the card you tapped and held on and shift all the cards one row up. If it is the top row it will not delete the start card.
Delete Column - will delete the *current column* of the card you tapped and held on and shift all the columns left one column. If you are on the first column the start card will not be deleted.

Create Map

When you press **Create Map** you will be presented with a simple map editor. This editor gives you the same tools and abilities that the developers of **Cato's Hike** used to make all the levels in the game. It is very powerful and it should be good fun to experiment with it.

Press the green button in the center to bring up the tile selection dialog:



After tapping on it you should get the tile selection where you can choose different tiles to edit your map:

On the top left there is a **Back** button which takes you back to the home screen. On the top right there is a **Play** button which lets you preview your map and test it. You should

always save your map before hitting play, though if you have already saved and don't hit save again it will save the map for you.

The toolbar has a bunch of icons each of which serves a useful purpose:

 - lists the maps that you have written so far. Tap it and choose any of your saved maps. It will also list below those the built-in maps. You can clone these and edit them or just tap to view them so you can get a better idea of how maps are built.

 - **saves** your map and asks you to name it. It is always a good habit to save.

 - use this to **email** your maps to friends and show them what you've done!

Cards - use this to **disable certain cards** from being allowed on the map. This is very powerful because you can make the level more difficult or force players to solve it a certain way to learn new techniques

Details - lets you name your maps and add a description. You can put anything you want in the description or leave it blank. If you write something you can use 3 **dashes "---**" to add a new page to your help or startup description. You can also set a **maximum number of cards** after which the player will be prevented from adding more cards to add further challenge to the level.

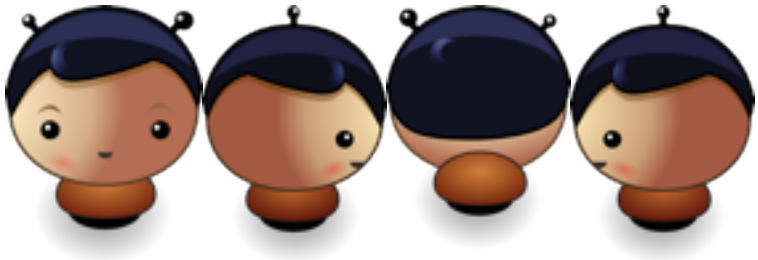
New - erases your current map and lets you make a completely new one. It will ask you to give it a name when you save.

Clear - clears the current map and sets everything to grass tiles.

Advanced Techniques when Creating a Map

Multiple Cards for Help - Use "---" to break up your help or description to more than one card.

Change Character Start Direction - Tap on the same square that you placed Cato and he will turn. Use this to change which way he is facing on start!



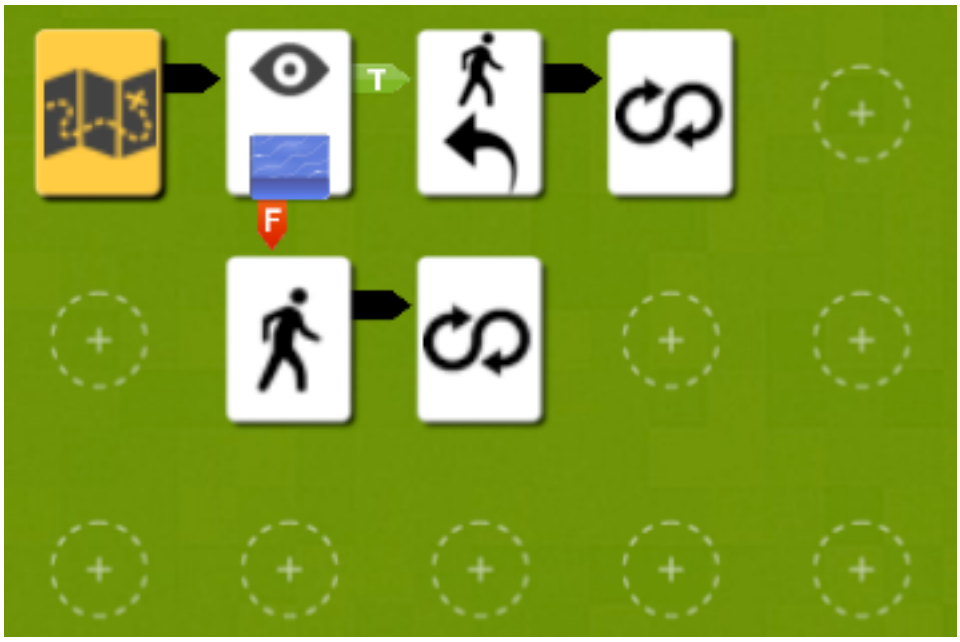
Settings

Use this screen to disable the music or sound effects or change their volume.

Sample Programs

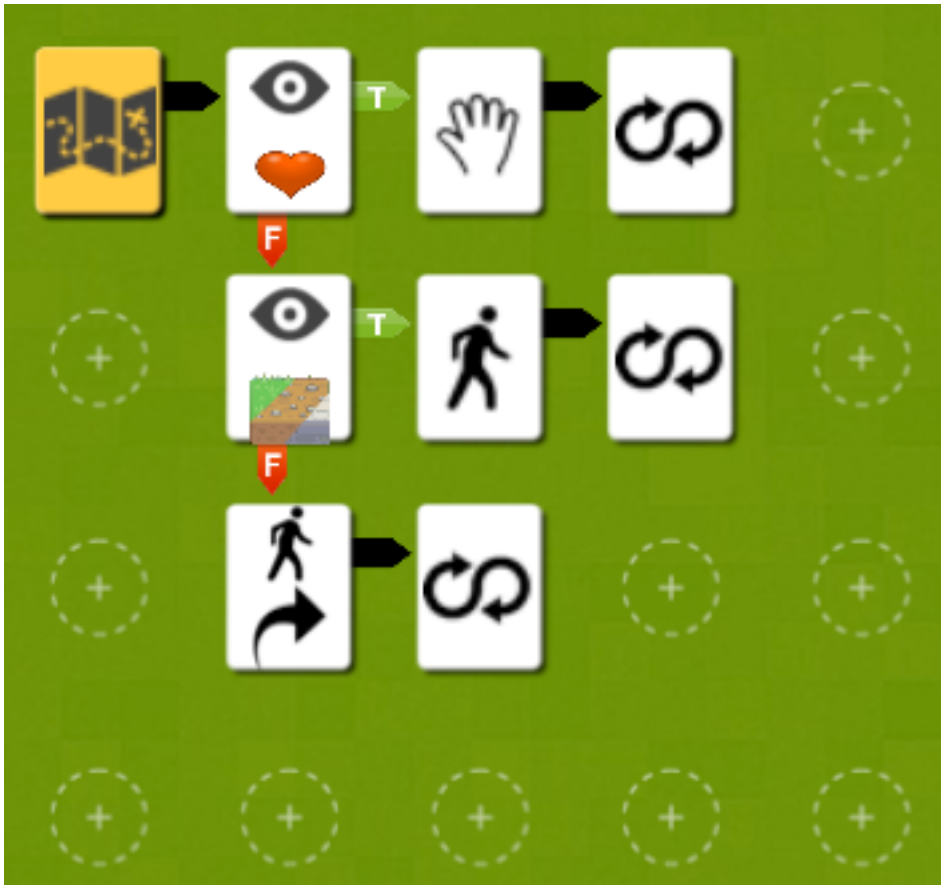
Below are some sample programs to give you an idea of how some of the more advanced programs work and to inspire you for your own future programs!

Basic - Look for Water



This is a very simple program you may recognize from the tutorial. What it tries to teach you is how to loop and how to look for things. In this case the program starts off simply by **looking to see if there is water ahead**. Obviously falling in water is bad and will end our level, so we decide that if we see water, we **turn to our left** the **loop** back to the **start**. If there isn't any water, then we **walk forward one step** because it is safe to do so and then start the program over, looking for water, turning if we see or walking if we don't. It is simple rules like this that can make the difference between a long and painful program of directly writing walk, turn, etc... straight to the goal and an elegant and simple program that teaches Cato how to get to the goal for that level. This is programming.

Basic - Look for Ground, Grab Hearts



This is another simple program that modifies the **look for water** above to show you how you can grab hearts off the ground for more health and otherwise just continue walking where it is safe to do so.

Intermediate - Looking Ahead, Jumping Forest

This is a relatively complicated program that solves the “Jumping Forest” level. This is interesting because it shows how one can look ahead. By chaining your looks you can look as many steps ahead as you want.

The program reads as follows:

- look one step ahead for ground, if:
  - ground, look one step ahead again, if:
    - ground, then it is safe to jump, even if it is jumping over land take the jump then start over
    - no ground, then it is safe to walk forward one step because there is ground one tile away, but we can’t jump because there isn’t ground two tiles away. Loop over
  - no ground, then look for ground, if:
    - ground, then it is safe to jump because we are jumping over water or a hole, then start over
    - no ground, then turn to the right and start over hoping that we can find ground or a place to jump

This relatively complicated program solves the level quickly and elegantly, can you solve it better?

Intermediate - GOTO's and GOTO LABEL's, a simple introduction

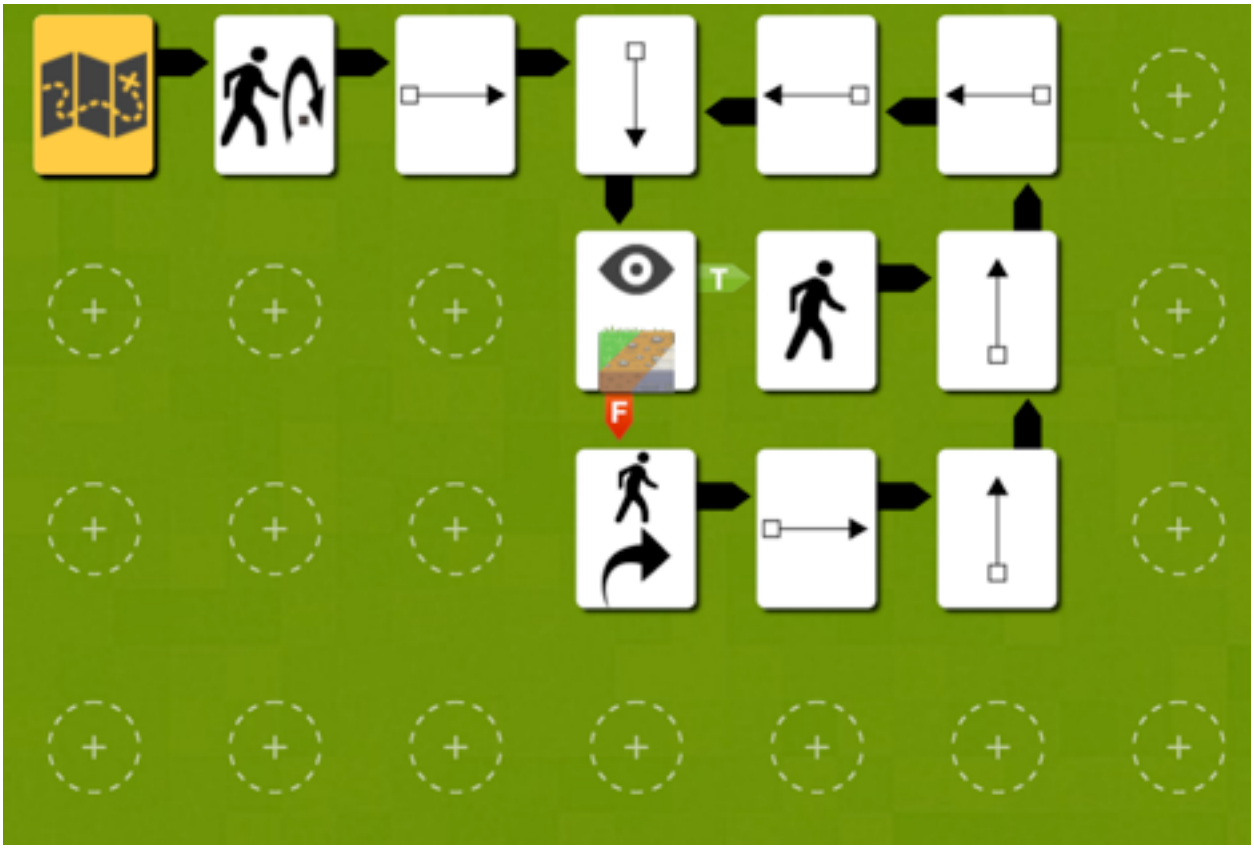
This is a fun program which demonstrates GOTO LABEL's and GOTO's. The idea here is that you can place a label in your code and loop back to it instead of looping back to the start, you can have up to **EIGHT** such labels and jump to any in your code which gives you enormous flexibility!

The program works as follows:

- Jump, this first step is to jump out of where we are, then we change the program after that to do something else
- **GOTO LABEL**, we set a yellow **GOTO LABEL** here, that means that we can start our program here at any time using the yellow **GOTO** card
- Look for ground, if:
  - ground, then walk and start over at the yellow **GOTO LABEL**
  - no ground, then look for ground 2 steps ahead, if:
    - ground: jump, because there is no ground one step ahead and there is two steps ahead we must jump over, then start over from the yellow **GOTO LABEL**
    - no ground: turn right, because there is no ground one or two steps ahead, turn and start over at the yellow **GOTO LABEL**

You can make your programs much more complicated using these labels and goto's mixed in with colored flags for example. Experiment, you'll find it fun!

Intermediate - Looping after Changing Strategies, Advanced Connectors

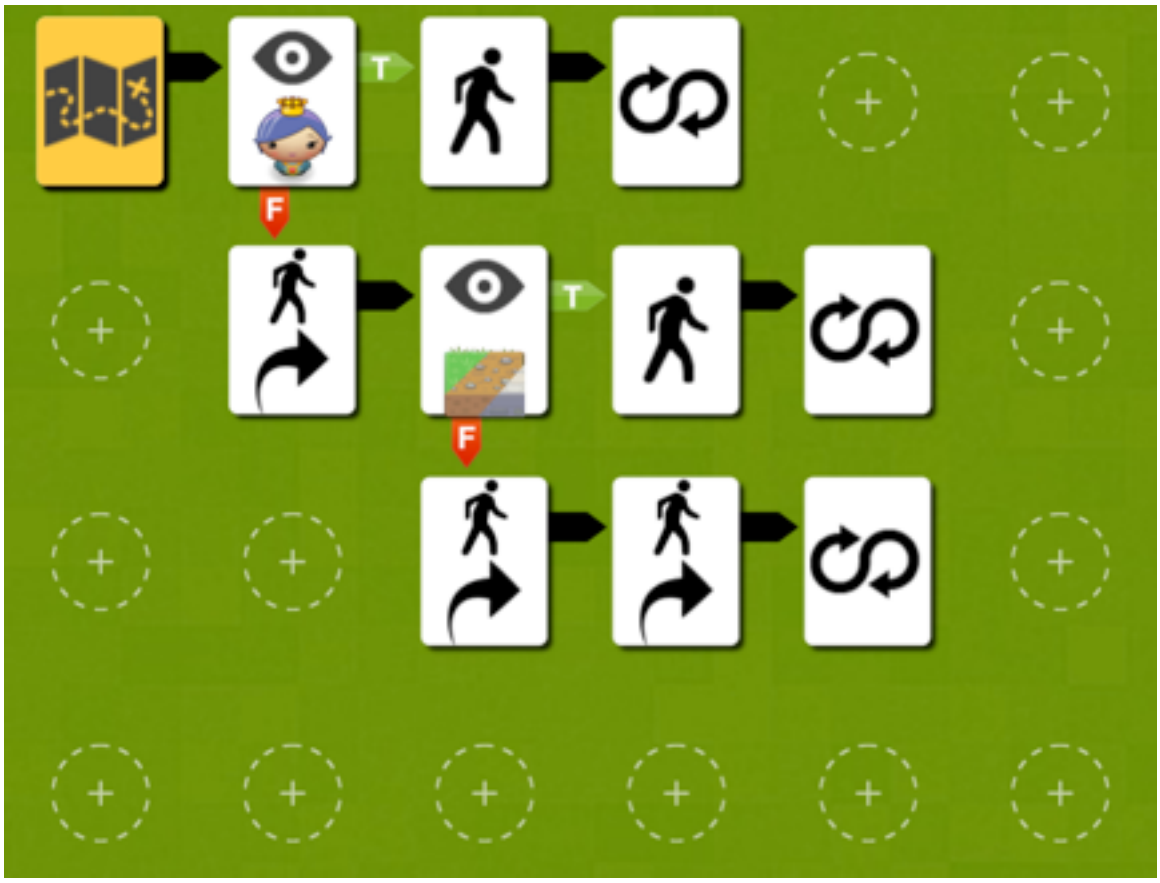


In this program, you learn how to use the connectors. Normally you would finish your program by looping back to start. Sometimes that isn't possible or desirable because it would make your program more complicated.

In this case for example, you need to start off by jumping once. Once you've jumped, you just follow the usual strategy of looking for ground and turning if you can't see walkable ground.

If you had to modify the program to figure out that you needed to jump and not walk through a maze for example on start you would have a much more complicated program. As it stands, you just connect back to the start of your new second program, almost a **subroutine** in programming-speak and you're on your merry way!

Advanced - The Right Way



This is a very complicated program that solves the level of the same name. Not every level is this hard and one of the key things to notice is that there are patterns on the level. The main thing in this level is to first start off by taking a right if you can and step forward if you can. You can see this on the look card that looks for ground. So on startup you turn right and try to take a step, if you can't, turn around to what would be your left and start over. It will take a right and try to take a step.

What this means is, it tries to first turn right and take a step, if it can't, it will turn 2 turns and start over. When it starts over it will turn once making it 4 complete turns. This means that if you can't take a right you will walk straight forward. If you can take a right you will attempt what we just described. Basically it is a program that favors taking rights if it can otherwise walking forward.

The looking for friend at the beginning of the program has no effect on what we just described except to say, if you see your friend, walk towards her, otherwise do what we just described above. She is what one might consider to be an **exit condition** for our program that lets us finish the level.