Change the color, size, shape, location and direction of your agent. You can change any of these traits during your game, such as having a character grow in size or change color.
You choose from the default agent traits by using the pulldown menu in the SET MY block.

Once you choose a trait, you can either type in the TO area or drag in a trait value block, such as BUILT IN SHAPE, blocks from the math drawer such as RANDOM.

Set the color by picking from a list of standard colors, choosing a random color, or selecting a specific color using an RGB code. RGB allows you to add values of red, green and blue (the primary colors of light) together to produce any color. Refer to an RGB chart for specific colors.
Create a new breed when you need a new category of agents which can follow different rules than the other breeds.
Now you can create agents of that breed and there’s also a new page in the workspace with that breed’s name where you can program all of its behaviors.

Click Edit Breeds > Add Breed > Breed name, then check that new breed shows up in the list.
Use keyboard controls to move an agent around the 3D world and to control other aspects of your game, such as picking up objects, managing inventory, opening doors, etc.
You can assign any keys on the keyboard to enable basic movement for an agent. Use the KEY HELD block for smooth movement, while KEY TYPED forces a player to keep clicking.

You can also use a modifier key, such as SHIFT or CONTROL with another key by using the AND block. This would be a good way to enable a player to both walk and run.
Heading controls the direction the agent faces. You can also use the heading trait to establish north/south/east/west in your game world.
Setting the heading during setup gives you control over which direction to face when you create new breeds. If you create 1 agent, the agent automatically faces 0°. For 2 agents, they face 180 and 0°. For 3 agents: 120°, 240°, and 0°. See the pattern? Here is the code used to generate the image on the front of this card (see DRAW ON TERRAIN card for explanation).

Changing the heading during your game is more precise than using the RIGHT and LEFT movement blocks. Headings also give a way to compare the direction of two or more agents.
The pen feature allows you to draw shapes and patterns on the terrain, with varying line thickness and color. You can even have several agents drawing at the same time to create patterns.
Any agent can draw on the terrain by using the PEN DOWN block. You can change the line color by changing the color of the agent and the line thickness by changing the agent’s size. Use PEN UP once the drawing is complete.

A few changes to your code can make a more complicated shape, such as a five-point star.

Here we create 25 turtles and scatter them to draw a bunch of stars in random locations.
You can apply a set color to a portion of the terrain (to represent water/fire/etc.) or change the entire terrain color by using the STAMP block.
You can apply a circle of color by choosing the appropriate agent size and using a STAMP and COLOR block. To change the entire terrain color create a giant turtle, stamp the color you want, then delete it. You will usually want to use the CLEAR TERRAIN block during setup to erase any previous colors applied with STAMP or PEN (see DRAW ON TERRAIN).

You can use the TERRAIN COLOR block to check the color under an agent (handy for setting traps).

The STAMP GRID block will just create a 1x1 colored square at the center of the agent regardless of agent size.
Change the point-of-view from overhead to following one of the agents to look like a more traditional 3D action or adventure game.
By default the camera starts from directly above the terrain. For a 3D view place the TAKE CAMERA block inside the CREATE/DO block of the agent you want to follow at the beginning of the game. If you create more than one agent the camera will follow the last one created.

For more precise camera control create a new breed called camera, put the TAKE CAMERA block on it and use keyboard controls to change the view in your game.

You can use the 4 camera buttons above webland to reset your camera to the default top view, zoom in and out, or switch back to agent (3D) view.
Teleporting means that you want the agent to instantly move somewhere else. This is a handy way to skip across your game world or even move to a new level.
The simplest solution is to program a KEY TYPED block with SCATTER to transport you to a random random location.

To simulate a portal, you could have a COLLISION teleport you to a specific place by setting the X and Y traits. (X and Y can both range from -50 to +50. In the center X=0 and Y=0, which is where all agents are initially created during SETUP).

Or shrink/expand the agent for a more slick effect.
Program your agents with the ability to jump. Control how high they go and how many times they jump.
When the spacebar is pressed the agent jumps into the air and then comes back down. The yield block separates the actions by slowing down the movements.

To make faster jumps, remove one or more yield block or increase the number of steps.

Alternatively, if you want to make jumping more realistic, see GRAVITY.
FLYING

Program your agents with the ability to fly above the terrain or even into outer space.
For easy flying just program one key to move up and another to move down.

Or you could simulate “flapping” which just requires tapping 1 key, such as the space bar, to keep flapping your wings, otherwise you will continue to fall.
You can use the default score box to award player progress in your game. You may also use widgets to display an additional score for an enemy or multiplayer.
While the way you update a score is always the same (using a SET DATABOX block), there are several ways to determine how score is kept in your game.

You may have the score increase when colliding with some form of treasure...

...or have the score increase continually as long as the player survives...

You can even create additional databoxes to keep track of an enemy or teammate’s score.
Sometimes it’s helpful to give the player a message in text – such as if you win or lose, directions on how to reset the game, or to enable character dialogue.
Click EDIT WIDGETS then NEW WIDGET, type a name such as “Message” and select the “Label” option.

A collision is often used to trigger a message.

Once you click ADD WIDGET It should appear among the other widgets--while in Edit Widget mode you can move your widgets by clicking and dragging. For dialogue, you may want to run SETUP first so you can position the text near your character. Once it’s in the right place click EDIT WIDGETS again.

You might want to hide your first message until later in the game.

A collision is often used to trigger a message.
A push button is only activated once, such as the Setup button. Toggle buttons can be turned on or off, like a light switch and are used to keep something repeating, like the default PLAY button.
The **WHEN ____ PUSHED** block tells the agent to run the blocks inside of it when the given button (in this case setup) is pushed.

The blocks inside **WHILE ___ TOGGLED** will keep running in a loop as long as the given toggle button is still on.

Click **EDIT WIDGETS** then double click on a button to rename. Click **NEW WIDGET** to create a new Push or Toggle button.
Some games allow players to choose difficulty before they start. Others offer multiple levels to enhance game play. You can give players these choices by using custom buttons.
For levels, design a large gameworld and have your player start in a different location depending on their selection (see TELEPORT).

One way to vary difficulty is to create more enemies depending on which push button the player selects (see BUTTONS).

Or make your game harder by making the enemies faster (see CUSTOM TRAITS).

For levels, design a large gameworld and have your player start in a different location depending on their selection (see TELEPORT).
CUSTOM SHAPES

Program an agent to look like an object from a library of custom 3D models.
The shape of an agent can be set to either a built-in shape...

...or a 3D model from the online library. *

You can change an agent’s shape and any time and can even create a simple animated object effect by creating a sequence of rapid shape changes.

* If you are linking to a model from the online library, make sure to link to the file that ends in “.obj” [http://slnova.org/shapes/space/moon.obj]
You can create and use a custom traits for any breed. Custom traits are things like health, lives and energy that StarLogo Nova doesn’t include automatically but that you might use in your project.
In the setup of your program be sure to set the starting value to what you want it to be (such as 100 for health or 0 for score). Then use them just like any other traits (size/heading/etc.), increasing or decreasing over time. If you want a trait such as Health to be displayed in a message box see KEEPING SCORE.
Recording sound can be a bit tricky the first time, but it’s worth it for the added joy sound effects bring to game play. You will need both sound blocks to record and play. Click record, then scroll up to WebLand to find the record box.

The first time your record you may need to enable your microphone. Click the Allow button then there may be a 2nd allow button you need to click at the top of the webpage.

Say (or sing) something short, type in the name, then click to save the recording.
Now you can select the new sound under the dropdown menu of the sound block. Place the 2 sound blocks wherever you wish to trigger your sound.

You also have the option of importing sounds. You will find the IMPORT SOUND button near the top right of your web page. You can only import sound from your own projects.

Say (or sing) something short, type a name, then click SAVE. Once you have finished recorded the sound, you may drag the start and end sliders to trim the sound.
Being able to detect when objects collide is essential for most games. You can use collision to simulate eating, to trigger a trap or to collect objects in your game.
Use the collision block to detect when one agent has collided with another agent. You will generally want to change the score (here adding a point) and delete the “collidee,” which is the thing you collided with. See FOLLOW/AVOID to find out about detecting nearby agents.
You will often want to make one agent follow/avoid another agent. We refer to this as AI or artificial intelligence. As your programming skills increase, so will the AI of your agents!
To have one agent follow another, face toward the agent, then move forward. Use the NEAREST block to find the agent you want.

To avoid, face toward the agent, turn around 180 degrees, then move forward. You may need to adjust the number of steps in NEAREST for best results.
A good game should surprise the player. You can set a random chance of something happening, like making obstacles appear 10% of the time or springing a trap just 1% - any chance you’d like.
One way to implement chance is to generate a random number between 1 and 100 and use that value to represent percentage. This code will only create a Turtle 5% of the time because approximately 95% of the time the random number between 1 and 100 will be greater than 5 so the CREATE block will not run.

While this code will create 10 enemies with approximately 10 percent being red and the rest white.

As you design your game or simulation, look for places where chance could make your game more fun to play. Chance makes players want to keep coming back, because each time they play the game is a little different.
Enable real gravity for all of the agents in your game by making them fall faster and faster over time for everything from falling debris to snowfall.
Since you want gravity to affect all agents the same way, putting your code on the Everyone page saves you the trouble of creating code for each breed. Add a breed trait called “fall speed” (see CUSTOM TRAITS) and if an agent is above the ground \( (z > 0) \), add to the falling speed (otherwise set both \( z \) and falling speed to 0). Then make the agent drop according to its falling speed.
Wouldn’t it be great if you could group several programming blocks together, label them, and reduce the clutter in your programming workspace? You can do all that, and more, by using PROCEDURES.

Use the large PROCEDURE block to combine your group of code blocks—be sure to name it, too. It does not matter which tab you put the procedure on. Then use a CALL block wherever you want to pull in the code.
If you add parameters you can further customize how the procedure works. Here we create a parameter called size so you can choose which size circle to draw each place where you use the Draw Circle procedure.
Even if you’ve set the game to only run for a certain amount of time, sometimes you want to end the game early if the player dies.

Be sure to use a toggle button to start your game. (See CUSTOM BUTTON for more about creating buttons).

When Player collides with a Hazard agent, the player toggles the PLAY GAME block off for everyone, which stops the game, and then deletes itself. You could also stop the game if the clock reaches a certain value, if you get a certain score, or if some other condition is met.
STARLOGO NOVA BLOCK CATEGORIES

**AGENTS** create/delete/scatter/take camera

**DETECTION** on collision/nearest agents/count agents

**ENVIRONMENT** pen/stamp/terrain/clock/world traits

**INTERFACE** when/while/toggle/databox/graph/table

**KEYBOARD** key held/key typed

**LISTS** list/splice/insert/get

**LOGIC** if/else/while/repeat/yield/=/<>/and/or

**MATH** random/+/-/remainder/round to/larger/smaller

**MOVEMENT** forward/back/left/right/up/down/face towards

**PROCEDURES** procedure/call/parameter/return-early

**SOUND** record/play/delete

**TRAITS** set/color/size/shape/x/y/z/heading

**VARIABLES** var name/set/value