

Game Programming with Stencyl

--from Geek Squad Academy

We're excited to help you ignite your students' creativity through gaming, and we think you'll enjoy it just as much! By combining your student's imaginations, our custom-created content, and Stencyl, your class will be creating games that can be played nearly anywhere.

Stencyl is a drag-and-drop game designer that takes the fear out of coding, allowing your students to learn about variables by snapping gameplay blocks together to program new features or change other elements of the game!

This material has been real-world tested at 60 technology camps nationwide for youth ages 9-18 as part of the Geek Squad Summer Academy program, and is one of our most popular classes. The lesson plans we've offered here provide step-by-step instructions to help guide you through classroom setup, gameplay demonstrations, several example variables, and saving the game for later play on PC, Mac, or anywhere else that accepts Adobe Flash!

Before You Get Started

Here is a quick simple setup that will need to be installed on all computers prior to the beginning of this class. Please review this lesson prior to teaching to the class to familiarize yourselves with the program. The times given at the beginning of every new topic are merely suggestions. Take as much time as you need for your class to understand the functionality of the program. We also strongly recommend each student has their own flash drive to save their finished product.

- Download Stencyl-2.0.0.exe by going clicking [here](#)
- After Stencyl installs, open program
- When in Stencyl, click the button "View Games Folder" – a new windows opens
- Double click "GAMES" folder.
- Copy the "GSA Programming Game.zip" and "Day 2 Programing Game.zip" folders from [here](#) and paste them to the games folder You must unzip the files first, then paste them into the games folder For reference, here is the direct file location:
C:\Users\GSSA\AppData\Roaming\Stencyl\stencylworks\games
- Restart Stencyl
- Verify Software Working

Table of Contents

- Activity One – Do You Want To Play A Game?
- Activity Two – Modifying the Game
- Activity Three – Scripting in Stencyl
- Activity Four – Customizing Your Game
- Activity Five – Advanced Techniques
- Activity Six – Save Your Game!

OBJECTIVE

- An introduction to gaming and what constitutes a great gaming experience.
- A basic understanding of programming variables, cause, and effect.
- A basic understanding of a script and how to create one.
- Create a video game that reinforces the concepts of game development and programming variables.
- Learn advanced techniques in programming variables

Activity One – Do You Want To Play a Game? – 5 minutes

Before the Activity

- Have Stencyl project ready to play.
- If students have flash drives plug their flash drives into their computer's USB port and do not remove them until after work is saved.

During the Activity

- Explain the following Stencyl functions. These can be seen from the “Resources Dashboard” on the left side of the screen.
- At this time the students should be listening to the teacher's instruction and following along only from any projection coming from the teacher's laptop.
- **Actor Types**
 - Teacher: “Actors include the main playable character, the other characters in the level, or anything the player can interact with. There are many things we can do to edit the character.”
- **Tile sets and Scenes**
 - Teacher: “Tilesets are just like puzzle pieces. You can piece them together to create the actual level or scene.”

- **Sounds**
 - Teacher: “Sounds add another level of depth to your game. In Stencyl, you can add sounds to actors, scenes, or when certain events take place
- **Logic**
 - Teacher: “Logic controls how other actors interact with the player, as well as, how the main playable character interacts with the scene. These are also known as Behaviors.”
- **Controls**
 - Teacher: “The Controls editor allows you to edit and add controls, such as jumping, breaking, and walking in your game.”
- After demonstrating the game, state that now all of the students are going to “create and modify their own game” and walk the students through the process.
 - Teacher: “On the desktop, find the red icon with the large white “C.” Double click on the icon and wait for the program to load.”
 - Teacher: “On the welcome page, you will see a few games to select. Double click on the game GSSA Programming Game – Gopher Roman. This will open the game.”
 - Teacher: “Now before we make any changes to the game, we should go ahead and save a new copy of it as to not disturb the original for the other classes! To do that, go to “File,” followed by “Save As,” and then save the game as your name. We’ll be saving often to ensure we don’t lose any of our hard work!”
 - Teacher: “Next, we’ll click on the blue arrow on the top right of the screen to play the game, as is, to see what the game looks like to begin with. This will be the button you will use throughout the game to check your progress by testing the game.”
 - Teacher: “We already have the start of our game so let’s see how it plays by clicking the “Test Game” button in the upper right hand corner.”
- Let the students play the game for around two minutes before transitioning. This will allow them to become familiar with the program

After the Activity Ends

- Teacher: “By show of hands, who is finding the game rather frustrating so far? Who would like to be able to actually play and enjoy this game and not keep dying over and over again? Who can tell me why the game **isn’t** playable in its current state?”
- Teacher: “That’s correct! Our hero can’t quite make the jump, can he? If you will all lend me your attention again for a moment, we will learn to fix this rather broken and rather non-fun game.”

Activity Two – Modifying the Game – 10 minutes

Objective

- Understand and be able to give an example of a programming variable.
- Understand the cause and effect nature of changing variables within a game.
- Understand and be able to explain scripts.
- Become comfortable with testing the game repeatedly for game changing elements.

Before the Activity

- You should have the class's attention prior to starting this activity by having pointed out how unplayable the game is in its current state.
- It is important to ensure that everyone stops what they are doing and listens for instruction all at the same time.

During the Activity

- Teacher: "Since we know our character cannot jump high enough to reach the first platform, let's change his jumping variable (attribute) so that he can."
 - Under "Resources" and "Actor Types" double click on our character called "Gopher" to open it.
 - Click on the "Behaviors" tab in the center and then select the "Jumping" behavior on the left hand side.
- Teacher: "Here we can see the actor's attributes for jumping, but we only need to worry about the "Jump Force." We need to change this to a higher number so he can reach the first platform, so let's try 45."
- Teacher: "Click the "Save Game" button in the top left and then test your game again."
- Teacher: "You will notice that now your character jumps much higher and now you can get on top of the first platform."
 - Show them one more variable. Under the "Walking" behavior change the "Max Walking Speed." Show them how they can make the character run fast or slow.

After the Activity Ends

- Encourage students to ask for help while they are working on their projects.
- Teacher: "Those are the variables that are most commonly edited. Now that we have gone over the basics of the variables, you have time to make all the different changes you want to your game. If you need help with one of the variables we talked about or

you want to tackle something a little harder, raise your hand and we will help as we walk around the room.”

Tips and Tricks

- Save often.
- Move quickly through this section. Students will be anxious to explore and will have questions as they progress through their customizations anyway.

Activity Three – Scripting in Stencyl – 5 minutes

Objective

- Students have a basic understanding of script and its role in a video game.
- Write a line of script from scratch.
- Understand how the script changed the game.

Before the Activity

- Make sure that the students have saved their game recently and their game still opens.

During the Activity

- Teacher: “Up until now, you have been working with pre-created scripts and changing variables. For example, this game was already pre-programmed to jump, you just changed the variable that determines how far the character can jump. Now, you will have the opportunity to actually write code for your game.”
 - Instruct the students to select “Actor Behaviors” under “Logic” on the left side of the screen.
 - Click on the dotted line box on the right that says “Click here to create a new Behavior.”
 - When the new box appears, name the Actor Behavior “Collect Pickup.”
 - Next, click on the Create button
 - Explain to the students that this screen will allow them to create their own script.
- Teacher: “These puzzle pieces are all parts of code. Separate, they cannot do much, but if we connect the puzzle pieces correctly, it will tell the computer what to do.”
 - In the workspace to the left, the students should click on the “Add Event” button and select “Collisions – Any actor collides with... - Member of Group.”
 - The new block now appears in the center. In the dropdown for “Actor Group” select “Players” so it now reads “when Self hits an actor of Players.”

- Now, click on the grey and pink button that reads “Sounds” on the right of the screen.
- Drag the pink puzzle piece that reads “Play Sound” into the work area. Place it inside the yellow piece we created for the collision.
- Instruct the students to click on the sound drop down menu to select the “Pickup Sound” audio clip.
- Find the blue “Kill:Self” puzzle piece on the right, using the search option if necessary and place it under the “Play Sound” puzzle piece.
- Click green “Attach to Actor Type” button in the top right hand corner.
- Teacher: “Now, we need to attach our newly created behavior to any of the pickup items used in the game. Preferably, all three of them.”
 - Instruct the Students to save their games and then give about 2 minutes to test their game out and see what their changes have done.

Activity Four – Customizing Your Game – 10 minutes

Objective

- Customize a game in Stencyl that can leave the room on a flash drive, fully-functional.
- Understand the cause and effect of changing code.
- Make decisions based on the game design concepts.

Before the Activity

- Make sure that the students have saved their game recently and their game is still opening.
- Each of the students should have a partially edited game. Encourage the students to try new things, but always to save the game ahead of time.
- Remind them that they should be editing variables and should not change the coding unless they speak to a teacher beforehand.

During the Activity

- Teachers should walk around the room.
 - Help with variables already taught.
 - Watch/play games with Students
 - Reinforce the concepts of good game design in all questions if possible.
 - Reinforce the concepts of variables during Student interactions.
 - If a student is mastering the topics covered, encourage one of the following variables:

- Walking
 - Walking Acceleration
 - Max Walking Speed
- Jumping
 - Jump Force
 - Max Jump Time
 - Jumping Slowdown
- Slowdown
 - Ground slowdown
 - Moving Ground Slowdown
 - Air Slowdown
 - Moving Air Slowdown
- Scene Editor
 - Show the students how to change the tiles in the first level by going to the Dashboard and selecting Level 01 under Scenes.
 - In the palette on the right use the tiles and the pencil tool on the left to draw new platforms for the characters in the game to jump on.
 - You can also use the actors palette on the right to place more enemies and powerups.

After the Activity Ends

- Request that they save their game one last time.

Tips and Tricks

- Remind students periodically to save their game.

Activity Five – Advanced Techniques – 15 minutes

Objective

- Allow students to tweak the Day 2 game with advanced features.

Before the Activity

- Have your teacher Stencyl project ready to play, if needed.
- Have the class double-click on the icon and wait for the program to load.
- Teacher: “On the welcome page, we will be opening a different game this time. Double-click on the game Day 2 Programming Game.”

- Teacher: “Now, before we make any changes to the game, we should go ahead and save a new copy of it as to not disturb the original for the other classes! To do that, go to “File” followed by “Save As,” and then save the game as your name and group name. Once again, we’ll be saving often to ensure we do not lose any of our hard work!”

During the Activity

- Recap the day’s main points
 - What is programming?
 - How do variables (attributes) work?
 - What other things could we program in our games?
 - Suggest things like power ups, changing gravity, and changing colors.
- Introduce the challenges and help students to add as much stuff to their games as possible through custom behaviors.
 - Do not tell the students exactly how to complete the challenges, but use the below information as a basic guide on one way to achieve the desired results.

Challenge Ideas

Create a powerup that makes your character jump higher

- This should be a new Event within the existing Jumping Behavior (since we need to affect the Jump Force Attribute).
- Create a “Collision – Between Type & Type” Event
- You could also use a “Collision – Between Group & Group” Event with Players and Powerups.

Create a powerup that makes your character run faster

- This should be a new Event within the existing Walking Behavior (since we need to affect the Walking Speed Attribute)
- Create a “Collision – Between Type & Type” Event
- You could also use a “Collision – Between Group & Group” Event with Players and Powerups

Create a powerup that makes your character grow bigger

- This should be a completely new Actor Behavior
- Create a “Collision – Between Type & Type” Event
 - Make sure you drag the “1st actor” tag down into the Actor slot of the grow block

- You could also use a “Collision – Between Group & Group” Event with Players and Powerups.
- Make sure to attach the new behavior to the Actor.

Change the gravity when you press a button

Jumping when not on ground ("flying" or "double jumping")

**This removes the requirements that the gopher has to be on the ground to jump, so he can double jump midair. **

- Logic > Actor Behaviors > Motion > Jumping > Updated
- Below the comment "Detect the jump key press, and initiate the jump
- Change: "If Jump Key is down and Can Jump? and Key released? and get actor value On Ground for Self as boolean"
- to: "If Jump Key is down and Key released?"
- - Time-sensitive powerups
 - e.g. to change jump force for 10 seconds
- This should be a new Event within the existing Jumping behavior (since we need to affect the Jump Force Attribute)
- Create a "Collision - Between Type & Type" Event
- Within the collision, enter this code:
- [set Jump Force to [[Jump Force] + [<how much you want to add to the Jump Force>]]
- Do after [<however long you want the powerup to last>]

Change the color of your character or other actors

- This should be a completely new actor behavior
- Create a “Basic – When Creating” Event
- Set it up similar to the above image.
- Make sure to attach the new behavior to the Actor.

Create a second level and have it change levels when you enter a region

Making scene change correctly

- Open Level 1 scene
- Click "Add Region" on the left
- Drag your mouse to create a region box
- On the right, click "Inspector" and name it "Finish Level"

- Go back to the Dashboard, right click Level 1, click duplicate
- Open Copy of Level 1, click Properties tab at top, change name to Level 2, edit this level in any way you choose
- Click the dashboard tab again on the left
- Go to scene behaviors - create new scene behavior, name it "Finish Level 1"
- Click Add Event -> Enters or Leaves a Region -> Member of Group
- Choose "Actor Group" and select "Players"
- At the bottom right, click the Attributes tab, click Add Attribute
- Name it "Finish Level 1" and select type "Region"
- Click "Attach to Scene" at the top right (green box)
- Select "Choose Region" and on the map, select the "Finish Level" region we created earlier
- Click the "Finish Level 1" tab at the top where the code we began is. Click the region dropdown box in the code, click Choose Region,
- Then click Finish Level 1.
- Add the following code to the code box:
- [if [not [scene is transitioning]]]
- [switch to [Level 2] and crossfade for [1] secs]
- Test Game

Tips and Tricks

- The students should be encouraged to experiment with their games, but be careful to make sure that their expectations are realistic. Stencyl is a very powerful program that can create some high quality games, but they will need to keep their behaviors fairly simple or things will likely become “buggy” and difficult to fix. The more complex the behavior the more likely they are not work as intended. If you cannot find the blocks you are looking for make sure to always use the search bar to find blocks more easily.”

After the Activity

- At 10 minutes until the end of class, it's time to have the students start saving their work!

Activity Six – Save Your Game! – 5-7 minutes

Objective

- Publish the student's game to his/her flash drive.

Before the Activity

- Make sure that the students have saved their game recently and their game still opens.

During the Activity

- Click on the tab that says "Dashboard"
- In the menu at the top of the screen, select "Publish."
- Select "Export as Flash SWF."
- A save window should pop up on the screen within seconds.
- Select "Computer" on the left of the window.
- Select Removal Storage (Flash Drive)
- Another window will prompt to let the Student know that it exported correctly.

After the Activity

- Summarize the class.
- Teacher: "I want to congratulate you on some really great games. Just remember that these games did not happen by accident. When creating these games, you successfully wrote code and used variables to make a base game completely unique. You should be very proud of yourselves!"

We hope that you have found this activity to be fun and engaging with your students! If you have any questions or comments regarding the game or would like information on any of our program offerings please send an email to academy@geeksquad.com.