Topic	Git and Github	
Class Description	**Students will learn how to use different git commands to set up a repository, clone a repository, commit changes etc. remotely through the command line.**	
Class	**C26**	
Class time	**45 mins**	
Goal	• Install git and command line git bash. • Use git commands to manage, work and update a remote repository.	

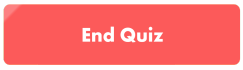
Resources Required	• Teacher Resources
	  ○ Laptop with internet connectivity
	  ○ Earphones with mic
	  ○ Notebook and pen
	• Student Resources
	  ○ Laptop with internet connectivity
	  ○ Earphones with mic
	  ○ Notebook and pen	

Class structure	**Warm Up** **Teacher-led Activity** **Student-led Activity** **Wrap up**	**5 mins** **15 min** **15 mins** **5 min**

CONTEXT

- **Review the concepts from the previous class.**
- **The pain of manually downloading, unzipping, updating a github code base.**

Class Steps	Teacher Action	Student Action
Step 1: Warm Up (5 mins)	Welcome back to the class. Do you remember what we did in the last class?	**ESR:** Yes! - We added images and animation to all the game objects.

		- We learned about class inheritance and how to create new classes which extend from a parent class and inherit all the properties and functions of the parent class.
	Before that I have an exciting quiz for you! Are you ready? Teacher click on the  button on the bottom right corner of your screen to start the In-Class Quiz. A quiz will be visible to both you and the student. Encourage the student to answer the quiz question. The student may choose the wrong option, help the student to think correctly about the question and then answer again. After the student selects the correct option, the  button will start appearing on your screen. Click the End quiz to close the quiz pop-up and continue the class.	**ESR:** **Yes**

	Good. Now, we must create a slingshot with which we can shoot our pigs. The current way of using mouse control for the bird is too easy.	*Student listens.*
	But before we start with creating a slingshot, don't you think we should do something about having to repeatedly go on the Github link, downloading the zipped files, unzipping them and then starting on our projects? This is a pain! Remember the golden saying - DO NOT REPEAT YOURSELF. Today, we will learn how to do all these things and more in an easy and fun way where we don't have to go through all this pain. Let's get started.	-
Teacher Initiates Screen Share		
CHALLENGE • **Install command line git and git bash.** • **Clone the activity repository.** • **Make some changes to the code and commit these changes to a remote repository.**		
Step 2: Teacher-led Activity (15 min)	We have been using GitHub website to host our code online.	

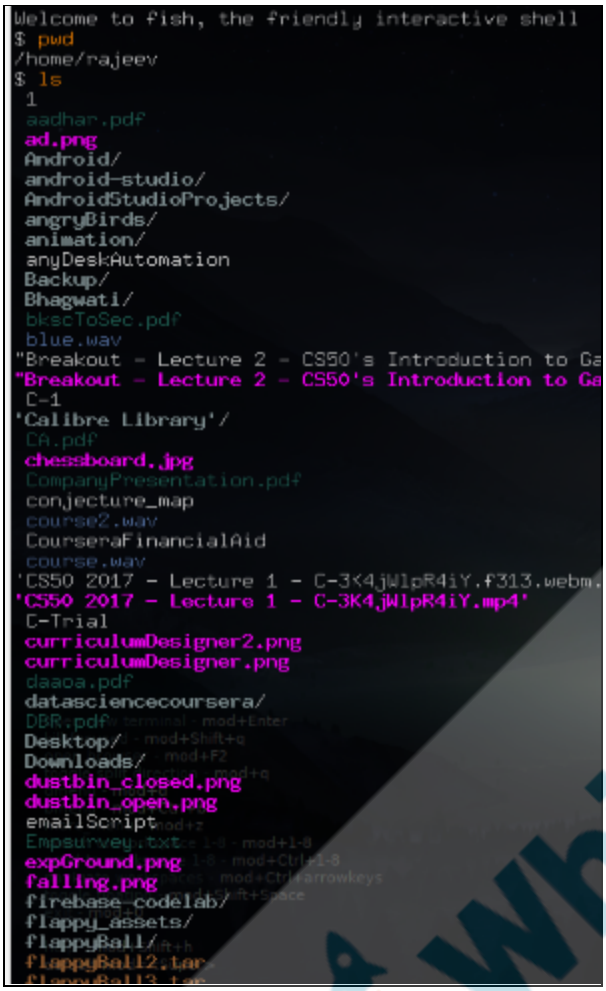
	GitHub uses a tool called Git which is used to keep track of all the changes in the code. It is a very powerful tool for developers to keep track of the changes they are making to their code and also keeping the track of the changes made while collaborating with others. We will learn more about Git in today's class. But before that, we need to install Git into our system. *Guide the student to install Git into their system. Make sure they check the checkbox to install the git bash shell for Windows.* *Also guide them to create a GitHub account if they didn't create it in the previous class. Get them to remember the GitHub credentials (username and password).*	*The student installs Git on their system.*
For Mac (follow the instructions given in the PDF): https://whitehatjrcontent.s3.ap-south-1.amazonaws.com/curriculum/PRO+Asset/git_installation_MAC.pdf after installing Git to check the git version and set git id and password. note-(git version may differ)		

For Windows:

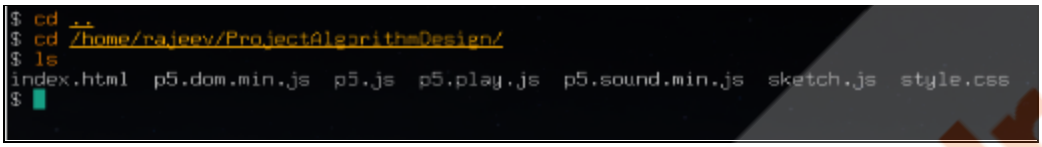
https://gitforwindows.org/
REMEMBER TO CHECK THE INSTALL GIT BASH CHECKBOX.

	Open your git bash shell (for Windows) or a normal terminal (for Mac). You know every computer uses a hard drive which is its memory. A hard drive is divided into sections and each section has a name. Each section can be divided into more sections called folders. Each folder may contain a file or another folder. This is a very powerful window. You	

	can do a lot of things from here. Let's check which section of the computer's memory we are in right now. We do this by typing 'pwd'. pwd stands for present working directory. *Teacher enters pwd on the bash shell and shows the output.*	*Student experiments with pwd in their own bash shell.*
		
	You can see all the files and folders that are present in this location right here in the window. You need to type the command 'ls' to list down all the files in this location.	*Student experiments with 'ls' in their own bash shell.*



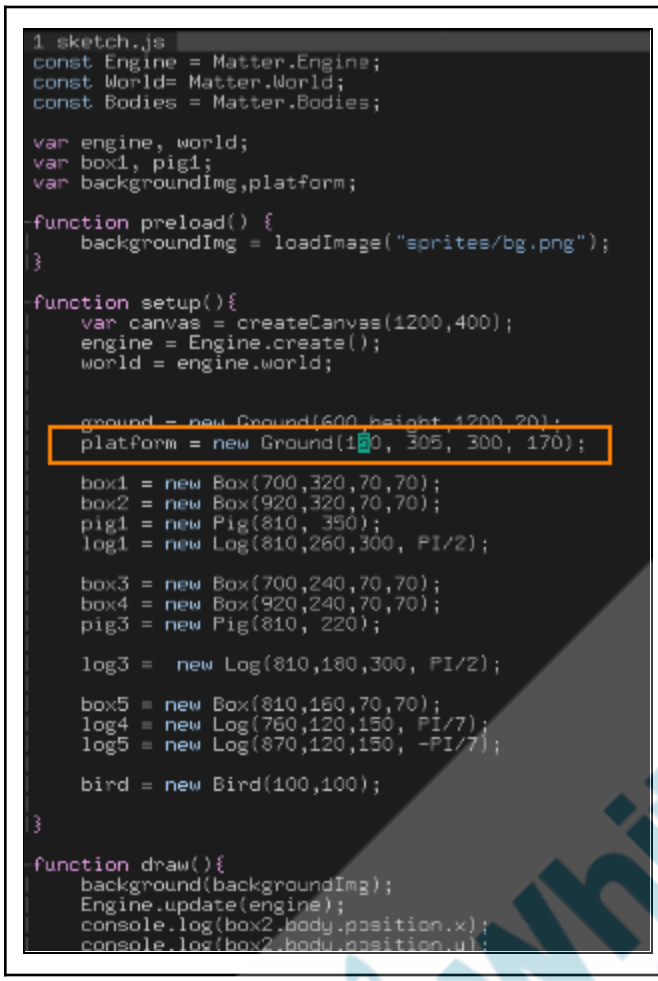
	Amazing! isn't it? You can in fact navigate to any other folder or directory and see the files/folders there. We use 'cd' (change directory) command to change the terminal's location to any other directory location. You need to enter the complete address of the location.	*Student experiments with the 'cd' and 'ls' commands in their own shell.*

	You need to type cd <SPACE><location of the new directory> You can then use 'ls' to list down the contents of that directory.	
		
	We can also create a new directory/folder in any location using 'mkdir' commands. 'mkdir' stands for make directory. You need to type mkdir<SPACE><Name and location of the new directory> If only the name of the directory is given, the directory is created in the default location. Use 'mkdir' and then 'ls' to show the directory created. *Get the student to create a Projects Directory.*	*Student experiments with the 'mkdir' and 'ls' command in their shell.* *Student creates a directory called Projects inside his home directory.*
mkdir Projects/		
	All this is cool! We can do much more but let's first learn to use git.	

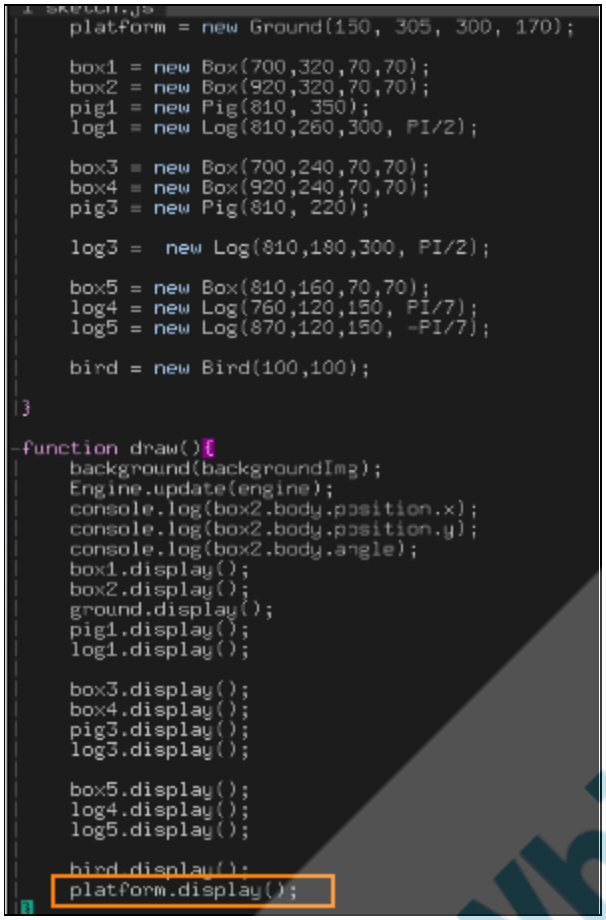
	We had progressed to Angry Birds stage 2 in the last class. There is a GitHub link in your student Activity.	*Student checks the Github Link.*
	Earlier, we used to download the files in a zipped format, then unzipped and opened them in the visual code studio. Now we don't need to do that. We can simply clone the Github repository using the "git clone" command. *Teacher shows how to git clone the repository.* ***[Teacher Activity 1]*** *Teacher uses 'ls' to show the cloned repository into the local system.*	*Student observes and learns.*

```
rocksvishu@Vishals-MacBook-Pro ~ % mkdir Project
rocksvishu@Vishals-MacBook-Pro ~ % cd Project
rocksvishu@Vishals-MacBook-Pro Project % git clone https://github.com/whitehatjr/angryBirdsStage2.git
Cloning into 'angryBirdsStage2'...
remote: Enumerating objects: 38, done.
remote: Total 38 (delta 0), reused 0 (delta 0), pack-reused 38
Unpacking objects: 100% (38/38), done.
rocksvishu@Vishals-MacBook-Pro Project % ls
angryBirdsStage2
rocksvishu@Vishals-MacBook-Pro Project % cd angryBirdsStage2
rocksvishu@Vishals-MacBook-Pro angryBirdsStage2 % ls
BaseClass.js    Ground.js       README.md       p5.dom.min.js   sketch.js
Bird.js         Log.js          index.html      p5.min.js       sprites
Box.js          Pig.js          matter.js       p5.sound.min.js style.css
rocksvishu@Vishals-MacBook-Pro angryBirdsStage2 %
```

	Isn't this nice and amazing! Half of our pain is gone! But we can do more. We can actually make some changes to our file. Let's create a big platform for our slingshot. We will do this by creating a new Ground object using the Ground class in our sketch.js file. *Teacher writes code to create a platform for the slingshot.*	*Student observes.*



```
1 sketch.js
const Engine = Matter.Engine;
const World= Matter.World;
const Bodies = Matter.Bodies;

var engine, world;
var box1, pig1;
var backgroundImg,platform;

function preload() {
    backgroundImg = loadImage("sprites/bg.png");
}

function setup(){
    var canvas = createCanvas(1200,400);
    engine = Engine.create();
    world = engine.world;


    ground = new Ground(600,height,1200,20);
    platform = new Ground(150, 305, 300, 170);

    box1 = new Box(700,320,70,70);
    box2 = new Box(920,320,70,70);
    pig1 = new Pig(810, 350);
    log1 = new Log(810,260,300, PI/2);

    box3 = new Box(700,240,70,70);
    box4 = new Box(920,240,70,70);
    pig3 = new Pig(810, 220);

    log3 =  new Log(810,180,300, PI/2);

    box5 = new Box(810,160,70,70);
    log4 = new Log(760,120,150, PI/7);
    log5 = new Log(870,120,150, -PI/7);

    bird = new Bird(100,100);

}

function draw(){
    background(backgroundImg);
    Engine.update(engine);
    console.log(box2.body.position.x);
    console.log(box2.body.position.y);
```



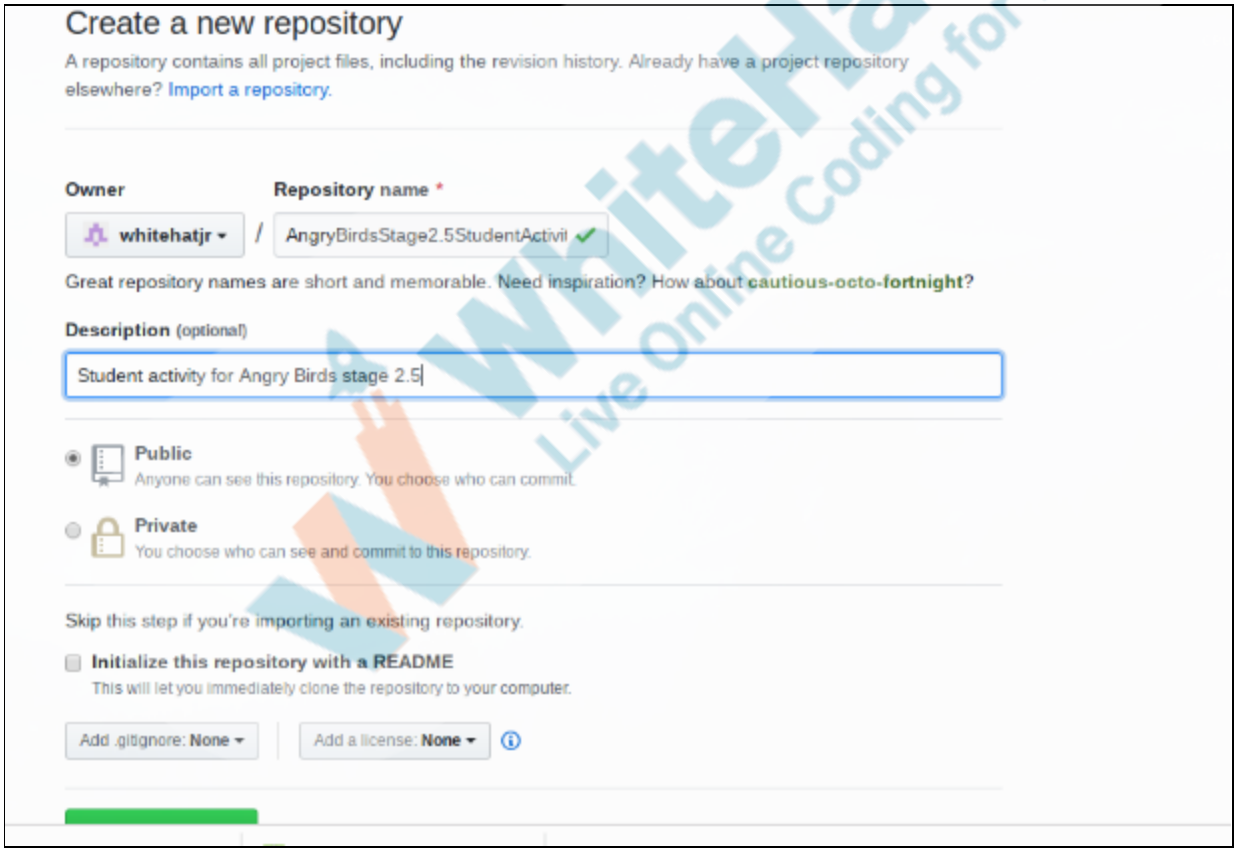
```
sketch.js
    platform = new Ground(150, 305, 300, 170);

    box1 = new Box(700,320,70,70);
    box2 = new Box(920,320,70,70);
    pig1 = new Pig(810, 350);
    log1 = new Log(810,260,300, PI/2);

    box3 = new Box(700,240,70,70);
    box4 = new Box(920,240,70,70);
    pig3 = new Pig(810, 220);

    log3 =  new Log(810,180,300, PI/2);

    box5 = new Box(810,160,70,70);
    log4 = new Log(760,120,150, PI/7);
    log5 = new Log(870,120,150, -PI/7);

    bird = new Bird(100,100);

}

function draw(){
    background(backgroundImg);
    Engine.update(engine);
    console.log(box2.body.position.x);
    console.log(box2.body.position.y);
    console.log(box2.body.angle);
    box1.display();
    box2.display();
    ground.display();
    pig1.display();
    log1.display();

    box3.display();
    box4.display();
    pig3.display();
    log3.display();

    box5.display();
    log4.display();
    log5.display();

    bird.display();
    platform.display();
}
```

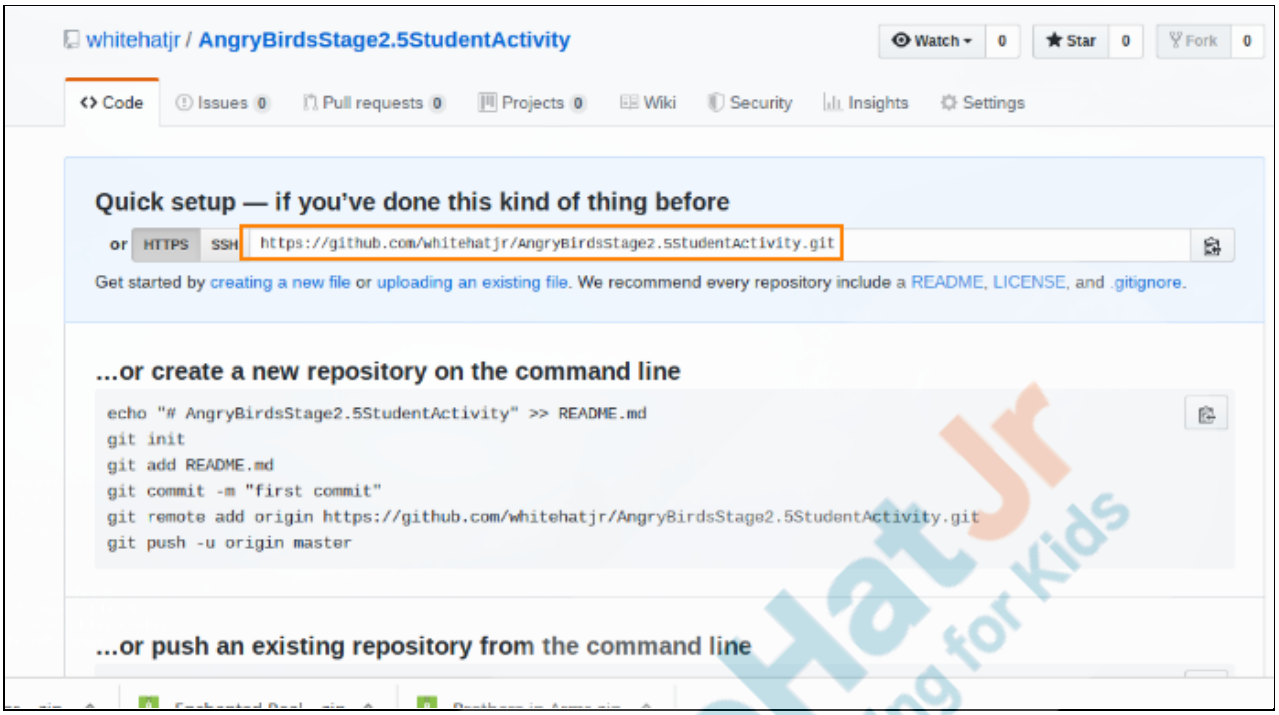
	We have the code ready for the platform. git knows that we have made some changes. You can use 'git status' to see that git knows that there have been some changes made.	*Student observes and asks questions.*
git status		
	One needs to get familiar with the following three areas while working with git: - there is a working directory. - there is a staging area. - there is a repository (local and	*Student observes and asks questions.*

	remote). Files are first modified in the working directory - just like we did right now when we modified the sketch.js file. Staging area hosts all the files which are changed before they are committed. A repository hosts all the different committed versions of the files. Every modified file travels from working directory to staging area to a repository. We need to add our sketch.js file to the staging area and then commit these changes (just like we did on Github after making changes to a file). Every commit is written with a commit message - so that later you can see the changes you made to the files. *Teacher shows how to make a git add and git commit the files with a message.* git add <file_name> git commit -m "<message>" (-m stands for message)	

```
$ git add sketch.js
$ git commit -m "Add Platform"
[master 02d825a] Add Platform
 1 file changed, 5 insertions(+), 3 deletions(-)
$
```

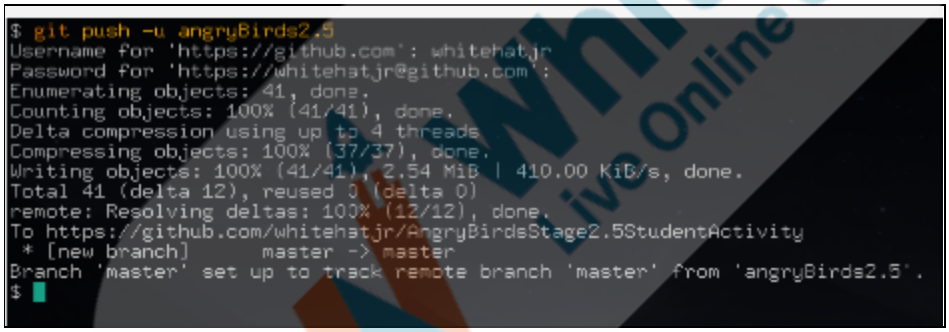
	Now that we have made some changes into our code, we can push this code directly to our remote github repository. For this, we have to create an empty remote Github Repository. Create an empty remote GitHub repository. **Note**: DO NOT INCLUDE A README.md file.	*Student observes and asks questions.*



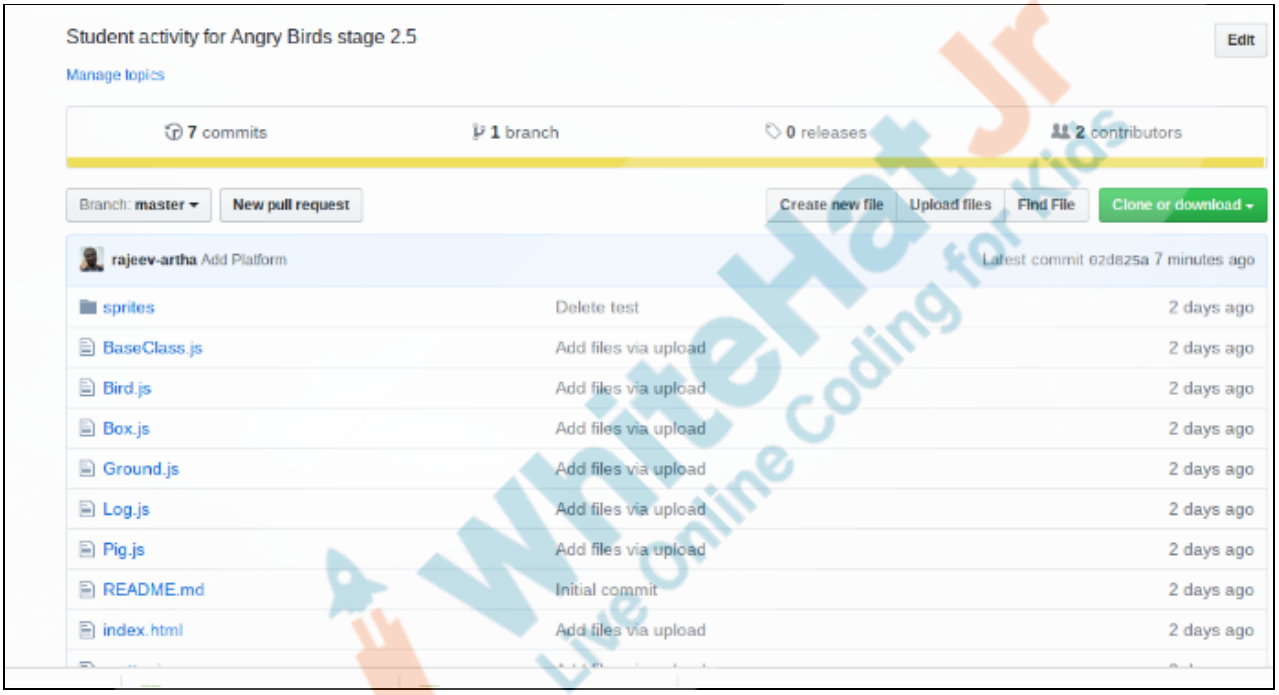


	Now we need to associate the working directory on our git bash shell to this remote repository. We do this using the 'git remote add' command. git remote add <remote_name> <remote_github_url> You can use any name for the <remote_name>. You need to use the url of the github repository you just created. *Teacher shows how to add the remote repository to the project.*	*Student observes, learns and asks questions.*

```
$ git remote add test https://github.com/whitehatjr/AngryBirdsStage2.5StudentActivity
```

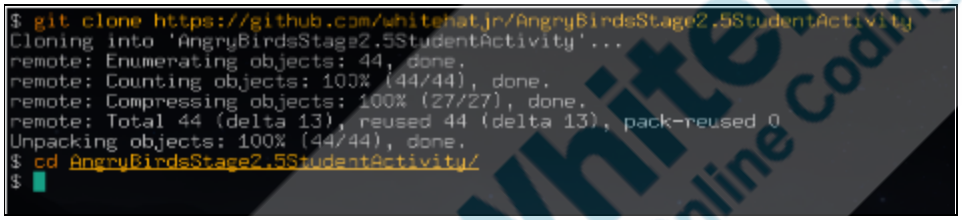
	Finally we have the committed changes stored in our local repository. We want to push these files and changes to our remote repository we just created. *Teacher shows how to push files to a remote repository.* git push -u <remote_name> (-u stands for username which is promoted by this command) **Note:** A username is needed so that the codebase gets uploaded in your relevant account.	*Student observes, learns and asks questions.*
		
	Check the remote repository url now by visiting github.com. Login and you will see the files have been updated there. *Teacher shows the updated remote url.* You can see all the commits ever made on this repository by typing the	-

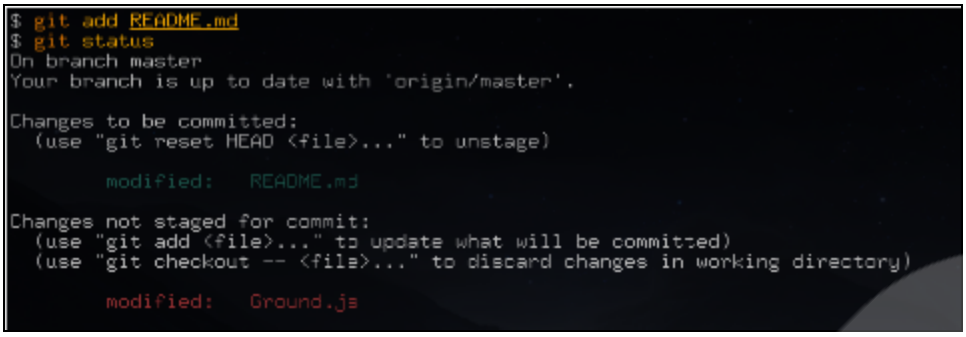
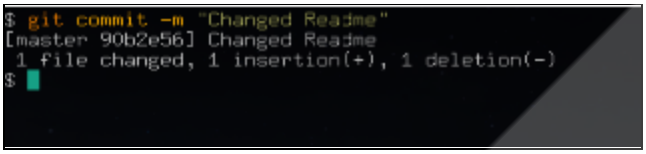
	command "git log". *Teacher shows how to use the git log command.* In fact, you can go back to any of the previous versions of the code anytime - but we will see that later.	



	There is some learning involved to do this but isn't this interesting? Why don't you do this on your own now. I can assist you.	**ESR:** Yes!
Teacher Stops Screen Share		
	Now it's your turn. Please share your screen with me.	

- **Ask Student to press ESC key to come back to panel**
- **Guide Student to start Screen Share**
- **Teacher gets into Fullscreen**

<table>
<tr><td colspan="3"><u>ACTIVITY</u>
• Clone a remote repository.
• Add the code to create a platform for the bird.
• Check the status of the new repository.
• Create a remote repository.
• Commit the change to a remote repository.</td></tr>
<tr><td>Step 3:
Student-Led
Activity
(15 mins)</td><td>Guide the student to clone the code from the remote repository (<u>Student Activity 1</u>).</td><td>Student uses git clone to clone the code from the remote repo to the local working directory.</td></tr>
<tr><td colspan="3">  </td></tr>
<tr><td></td><td>Guide the student to navigate to this directory.</td><td>Student uses "cd" to move to this directory.</td></tr>
<tr><td colspan="3">  </td></tr>
<tr><td></td><td>Guide the student to modify the Readme.md file to add more details.</td><td>Student makes some changes to the Readme file.</td></tr>
<tr><td colspan="3">  </td></tr>
<tr><td></td><td>Guide the student to add the files to the staging area.</td><td>The student uses 'git add' to add the files to the staging area.</td></tr>
</table>

<table>
<tr><td colspan="3"> 
</td></tr>
<tr><td></td><td>Guide the student to add it to the local repository with a commit message.</td><td>Student writes a commit message and adds the file to the local repository.</td></tr>
<tr><td colspan="3"> 
</td></tr>
<tr><td></td><td>Guide the student to create a remote Repository on Github without a ReadMe file.</td><td>Student creates a remote repository on github and copies the link.</td></tr>
<tr><td></td><td>Guide the student to add the remote repository to the working directory with a remote name and URL.</td><td>Student adds the remote repository to the local repository.</td></tr>
<tr><td colspan="3"><Use the repository link here>
 
</td></tr>
<tr><td></td><td>Guide the student to push the changes to the remote repository.

Students can see git status to check the status of the staging area.

Students can see git log to check the log of commits to the repository.</td><td>Student pushes the changes from the local repository to the remote repository.</td></tr>
</table>

```
$ git push -u angryBirds2.5
```

Teacher Guides Student to Stop Screen Share

FEEDBACK

- **Encourage the student to make reflection notes in markdown format.**
- **Complement the student for her/his effort in the class.**
- **Review the content of the lesson.**

Step 4: Wrap-Up (5 min)	How was the class today? How are you feeling?	**ESR:** varied
	Can you try to summarize what we learned today?	**ESR:** - We learned the use of git commands to - clone a repository. - stage and commit changes in a file in a repository. - push the changes to a remote repository.
	Git is exciting and very powerful. Developers use git to collaborate on their work. In fact, two developers can often work on the same code base using git. Git promotes collaboration. We will keep learning about the more powerful features of git as we progress through the lessons.	*Make sure you have given at least 2 Hats Off during the class for:*   

	You get Hats Off for your excellent work! Next class we will start working on the slingshot.	
Project Name: Masterchef Junior	*** This Project will take only 30 mins to complete. Motivate students to try and finish it immediately after the class.** **Goal of the Project:** In this class you have learned how to use different git commands to set up a repository, clone a repository, commit changes remotely through the command line(terminal). In this project you will apply what you have learned by publishing your most favorite recipe online! **Story:** Joshua is a world famous chef! And he wants you to do the task of taking his most favorite recipe and publishing it online for the world to see. I am very excited to see your project solution and I know you will do really well. Bye Bye!	**Note: You can assign the project to the student in class itself by clicking on the Assign Project button which is available under the projects tab.**

Teacher Clicks 		
Additional Activities	*Encourage the student to write reflection notes in their reflection journal using markdown.* Use these as guiding questions: • What happened today? - Describe what happened - Code I wrote • How did I feel after the class? • What have I learned about programming and developing games? • What aspects of the class helped me? What did I find difficult?	*Student uses the markdown editor to write her/his reflection as a reflection journal.*

Activity	Activity Name	Links
Teacher Activity 1	Angry Birds Stage 2 Github link	https://github.com/whitehatjr/angryBirdsStage2
Student Activity 1	Angry Birds Stage 2.5 Github link	https://github.com/whitehatjr/AngryBirdsStage2.5StudentActivity
Project Solution	Masterchef Junior	As it is an open-ended project, there is no specific solution