**Welcome to Day #6 of CGCC!**

Every day we will have a GitHub repository page that outlines each day and the activities that we will complete. We will also provide all homework on these pages.

Feel free to browse the other days to see what is coming up!

As always, let us know if you need any help or have any questions.

*Link to Camp GitHub*:<https://github.com/paigerodeghero/ClemsonGameCodingCamp/tree/master/2021>

**Day 6: Brainstorm & Work on Your Game**

**SCHEDULE:**

* Instructors start the video call
* Go over video and recording and start recording
* Campers can turn video and audio on or off at any time
* Delegating and Planning work
* Scrum!
* Get together in groups and work to complete your game

**INSTRUCTION**: Delegating and Planning Work (15 minutes)

It’s typically best to break apart your project into smaller, independently actionable, chunks. “Independently actionable” in this context means that one person is able to work on one piece of the project without their work affecting another person’s work.

If we were planning on building something like the original Mario game, we could start by separating it into a few small pieces

* Level Design/Story
* Art/Sound
* Game Physics
* Player Interactions

After these larger pieces are separated, you can take them and break them into even smaller chunks. For instance

* **Level Design / Story**
  + There should be a main surface for the player to run on
  + Platforms should be made for the player to jump up on
  + There should be characters that can take lives from the player
  + There should be an award system that gives the player score
  + The game should start with a narrative about getting to a castle to save a princess
  + The character should be awarded at the end with a narrative about saving the princess
* **Art**
  + The ground needs to be designed
  + Platforms need to be designed to look like the ground, but different and floating. Bricks?
  + The character sprite needs to be made
  + The boxes that hold coins need to be designed
  + The coins need to be designed
  + A sound needs to be made when a coin is picked up
  + The characters that can take lives from mario need to be designed
  + The characters that can take lives from mario need to have an animation for them being crushed
* **Game Physics**
  + There should be gravity
  + The player should be able to jump
  + The player shouldn’t fall through the ground
  + The player shouldn’t fall through the platforms
* **Player Interactions**
  + When the arrow keys are pressed the player should move
  + When the player jumps on an enemy character the character should be crushed
  + When the player jumps in to a box the box should create a coin
  + When the player overlaps with a coin that coin should be collected
  + When the player collects a coin there should be a sound that plays.

Each one of these steps can be completed independently by a team member, but you may notice that there is somewhat of a dependency between some of these. For instance, if we were to play a sound when a coin is picked up, first we need to be able to pick a coin. In all software projects there is typically some sort of dependency like this, but we’ll talk about that later.

Always try to play to your groups’ strengths. A typical software development team is composed of backend engineers, frontend developers, and designers. Similarly, you will need people to make art, program, and write stories. If you have someone who enjoys art, let them work on the parts that involve art; same with programming and stories. That doesn’t mean you should dedicate roles to people, but **your end product will always come out best when people are able to express their passions through their work.**

**INSTRUCTION**: Scrum (15 minutes)

Earlier, we mentioned some of the tasks that we made when we were designing mario had an interdependence on one another. In the software engineering world, we like to use a project management practice called [scrum](https://www.scrum.org/) to manage and address these tasks. In scrum, team participants are each encouraged to take a chunk of work, usually called a “work item”, work on it, and report in at a specified amount of time to report their progress. If someone has a work item that is being blocked by another person’s progress, they are encouraged to talk about this with their teammate to help reprioritize that work item.

We call these meetings “Standups” people take turns talking, and answer three questions:

1. What have you done since the last meeting
2. Are you currently stuck on something / is there anything blocking you
3. What are you planning on doing next

We will add a fourth one for this project

1. Is there anything that you think you need to get help from with a TA

**INSTRUCTION**: Start working on your game (Remainder)

Get back into your groups and start working on making your game a reality. Remember to do regular scrum meetings, and frequently push your work to GitHub.

What do we already know how to do?

What are we excited about learning or tinkering with?

Are there any common issues that are arising that the instructors should address to the cohorts by tomorrow. The main goal is to be able to ask about a problem, table it, and allow someone else to look into it.

**How to Scrum**

1. Have a three-minute meeting with your team
2. Commit your code to your repository
3. You should ask your teammates four questions
   1. What did you do?
   2. Where are you stuck?
   3. What are you going to do next?
   4. Is there something that we should get the instructor’s help with?