**Welcome to Day #3 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 3: Flappy Threads the Needle and Beginning Game Development**

**SCHEDULE:**

* Instructors start the video call
* Going over homework from day 2
* Short review of players and player interactions
* Introduction to Scenes
* Students begin to make their own scenes for Flappy Bird
* Break
* Questions
* Introduce group game
* Play Among us with instructors
* Meeting your team
* GitHub Classroom setup
* Questions
* Introduction to pixel art
* Students break into groups and begin designing characters
* Additional Questions
* Introduce Homework

**INSTRUCTION**: Going over last night's homework assignment of making one change to the game (approximately 5 minutes) or practice screen sharing if no one wants to volunteer

Have each student group go over their homework with the group, explain what change they made to Professor Platypus (2-3 minutes each)

**ACTIVITY**: full camp activity (approximately 15 minutes)

Students demonstrate what they have and ask any questions about player interactions

* **INSTRUCTION**: Add code to move player up and down (15 min)
  + Video instructions part 1: <https://youtu.be/ATUObD9IEsU>
  + Video instructions part 2: <https://youtu.be/oafvLTNNA2Q>
  + Open player.gd in Microsoft Visual Studio Code
  + Replace the contents with below code
    - This will move the player up and down as well as play the swim animation

extends RigidBody2D

var started = false

func \_physics\_process(delta):

    if Input.is\_action\_just\_pressed("swim"):

        if !started:

            start\_swim()

        swim\_speed()

func start\_swim():

    started = true

    gravity\_scale = 5.0

    $AnimationPlayer.play("swim")

func swim\_speed():

    linear\_velocity.y = -200

* + Save the game
  + GitHub: <https://github.com/domini4/ProfessorPlatypus/releases/tag/1.07>
* **INSTRUCTION**: Add limits to where the player can go (10 min)
  + Video instructions: <https://youtu.be/tkjs7on87zk>
  + Modify Player.gd script to sop the player from falling beyond a limit

extends RigidBody2D

var started = false

func \_physics\_process(delta):

    if Input.is\_action\_just\_pressed("swim"):

        if !started:

            start\_swim()

        swim\_speed()

    if position.y >= 500:

        gravity\_scale = 0

        linear\_velocity.y = 0

        $AnimationPlayer.stop()

func start\_swim():

    started = true

    gravity\_scale = 5.0

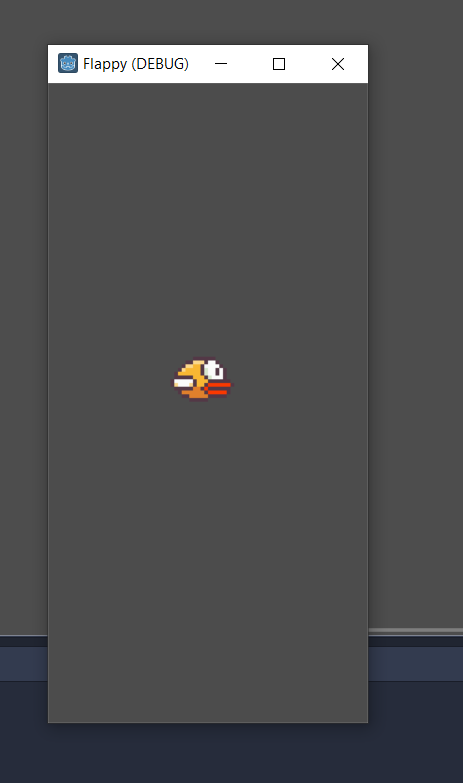
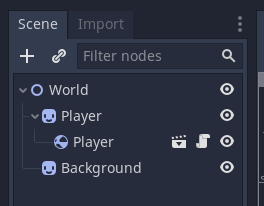
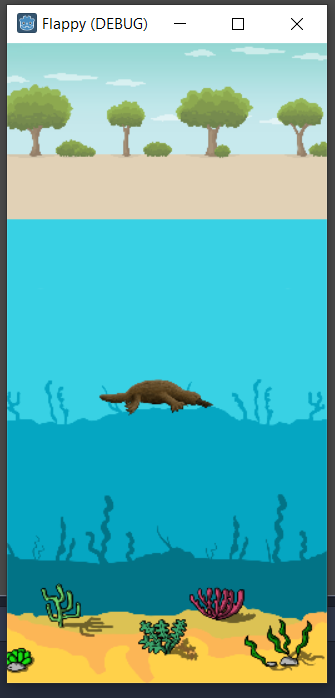
    $AnimationPlayer.play("swim")

func swim\_speed():

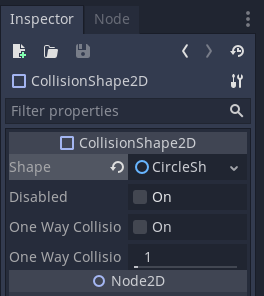
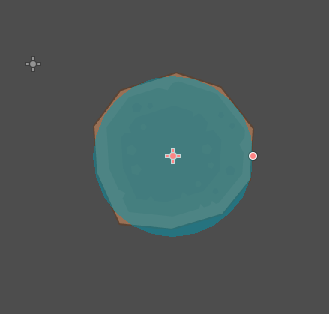
    linear\_velocity.y = -200

* + Save the game
  + GitHub: <https://github.com/domini4/ProfessorPlatypus/releases/tag/1.8>

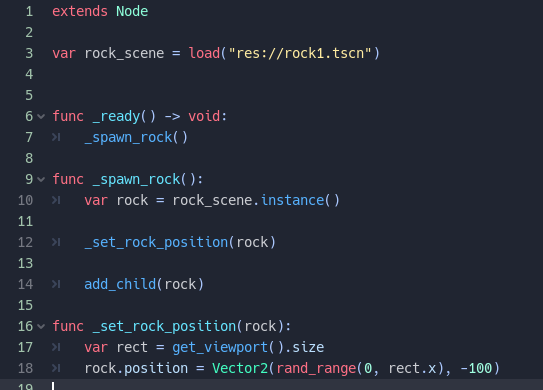
**INSTRUCTION**: Scenes – Walls and Textures (30 min)

* Change the aspect ratio of the game scene
  + Project > Project Settings > Display > Window
  + Width: 256
  + Height: 512
  +  
* In the FileSystem window in Godot engine, ensure that you have an assets folder
  + Assets>textures and there should be several assets there
* Load the background
  + Create a new sprite on World – right click the world node to add a child node and sprite
  + Rename Sprite to “Background”
  + In the inspector tab add a texture from assets>textures>prof\_platty\_background
  + Resize using the orange handles to your scene size
  +  

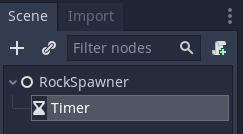
Adding Obstacles

* Adding our first obstacle
  + Click on the + in the Scene window and ass a RigidBody 2D
    - Node > CanvasItem > Node2D > CollisionObject2D > PhysicsBody2D > RigidBody2D
  + Right click on RigidBody2D and rename it to “rock1”
  + Right click on rock1 and add a child Sprite Node
  + Add a texture to the Sprite node from the assets folder same as before with the player Sprite
  + Add the collision shape to the rock
    - Right click on rock1 and add a CollisionShape2D
    - In the inspector tab under “shape” click the drop down and select circle and rightly cover your rock
    -  
  + Finally save this as a scence by right clicking on rock1 and “Save Branch As Scene”
  + Now we can test the game and you will find that the platypus will hit the rock and go off course, we don’t want that! So lets move to the next step to see how we can fix that and make the rocks come from off screen

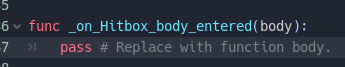
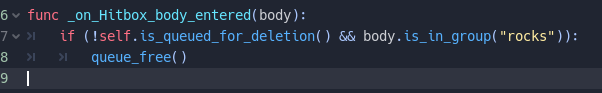
Spawning Rocks

* Spawning rocks
  + Delete the Rock1 node from the world scene (its okay! It is saved – that was our last step)
  + Create a new node of type “Node” and name it RockSpawner
    - Save this node as its own scene using the “save branch as scene”
  + Attach the following script to the spawner by right clicking on the node and going to attach script
    - 
  + If we run the game now, the rock will bonk our platypus in the head as though it is falling out of the sky – not quite the right direction
  + Back into your rock1.tscn and look in the inspector
    - Change Gravity Scale to 0
    - Linear X velocity to -100 so that it moves from left to right on our screen
  + Now the rock should be moving across the top of the game screen

Spawning MORE Rocks

* We need the rocks to spawn not only at the top of the screen so change the rock position function to
  + rock.position = Vector2(250, rand\_range(-10, 400))
  + 250 so they spawn off screen, -10 to 400 so they spawn somewhere above or below the platy
* Now run the game and see that the rocks spawns in different places with each refresh of the game
* Now we need more rocks!
  + To do this we will need a timer that will spawn our rocks at intervals
  + Go into the RockSpawner Scene and add a timer node at the top
    - 
  + Rename this to “SpawnTimer”
  + In the Inspector Window check “autostart”
  + Now we need to connect it to a signal
    - Go to the Node tab (next to the inspector)
    - Click on “timeout” and connect it to “RockSpawner” node
  + It will add a new function line named “\_on\_SpawnTimer\_timeout():”
    - Add the line: \_spawn\_rock()
  + Run the game and TOO MANY ROCKS
    - In the SpawnTimer node change the Wait time 3 and run the game again
  + We will want to optimize our game some to despawn rocks that have left the viewport
    - Go to the Rock scene add a script
    - Then add a Visibility Notifier2D Node
    - Connect it to the Rock1 node
    - In the script add the line: queue\_free()
* Now we want to group the rocks
  + Open the Rock scene and go into the Node tab (next to Inspector) and select groups

The Disappearing Platypus

* Now we need to make the Platypus disappear when hit with the rocks and end the game
* Go to your Player Scene and create a new Area2D node and name it hitbox
* Add a collision shape 2D to the hitbox
  + Go to the first collisonshape2D and click the drop down on shape, then copy
  + Go to your new collisionshape drop down and hit paste
* Click on the hitbox node and go into the node tab and then signals
  + Select “body\_entered” signal and connect it to the player node
  + This creates the following script in the player.gd script
    - 
  + Add the following line:
    - 

10 MINUTE BREAK

**~~INSTRUCTION~~**~~: Instructors introduce the game “Among Us” (approx. 5 min)~~

**~~ACTIVITY~~**~~: full camp activity (approximately 20 minutes)~~

~~Break into groups of no more than 10 with random number assigner~~

~~Play “Among Us”.~~

**ACTIVITY**: Meeting your team (30 min) [Paige]

* Breakout rooms assigned by Makayla: each team + 1 TA. These are new teams assigned after Day 1.
* Introductory Interview is in a Google Doc. TA shares link with students in the breakout room.
* Activity: Meeting your team. (One student at a time, each student goes for 30 minutes / number of students, TA to facilitate handoff). Student with earlier alphabetical first name goes first.
* Introductory Interview with Project Partner
* Name
* What is your contact info (e.g. phone or email or Snapchat or Insta or Discord)? For backup purposes in case technology doesn't work.
* When can you work on the project outside of class?
* Negotiate mutually exclusive times to edit code
* Which grade are you in?
* What is your background with computers? What skills do you have?
* What do you find most interesting about computers and how does this impact your future educational plans?
* How do you manage your time when you get busy with a lot of tasks?
* Have you worked on a team project before? If yes,
* How often did your team meet together?
* Did your team have a leader? What did that leader do?
* What was your role on the team?
* How well did you get along with your teammates related to work, or related to non-work?
* Who are the other members of your family? Do you live with them?
* What are your hobbies/interests/passions that are not related to this class?
* Do you have any unique skills/tricks that you can show us now?
* Who is your favorite teacher in school? What do they teach? Why are they your favorite?
* What was the last non-software-related book you read and describe what it was about in 2 sentences.
* What is the best movie you saw in last 3 years, and why was it your favorite?
* Do you have a personal hero in your life? Who is it and why are they your hero?
* What kind of programming experiences have you had in the last 3 years?
* Tell me about a trip you took that was far away from where you live?

**INSTRUCTION**: Introducing Game Design (20 min)

* What are some of your favorite games, what are their genres?
  + Students unmute and talk about their fav game
* (bring this up a little later in the camp, talk about at the start of each day on the second week of camp 🡪 self assessment of where they think they are and what do they think they can get done that day and by the end of the week)Scope – we only have a week to design a game, so we need to think about what will be realistically possible within the time that we have
* Show some examples of Godot Game Jam examples
* Elements of Game Design
  + High level goals – What do you want to happen?
  + Genre? RPG, Racing, Puzzle, platformer, shooting, racing, walking
  + What do you want to be main mechanics of the game?
  + What incentives do you want for your player?

**ACTIVITY**: group activity (25min-35min)

Students break into their teams to talk about what they think their game genre might be. Come up with 3 game mechanics or aspects they would want to have in the game. And have one be an object so they can draw it later.

Instructors can go around and be a sounding board for the students

**INSTRUCTION**: Introducing Making your own pixel art with (35min)

* Go over where to find free pixel art
  + Kenny.nl
  + Itch.io
  + Camp coordinator made assets?
* Download some of these that you think are useful or cool looking to get an idea of what you may want to pull inspiration from
  + Ask 3-4 campers what they picked and why they thought it was cool
* Begin instructions on character and game development
  + Think about the type of game you want to make!
  + Say my game is a 2D top-down RPG where the player goes on a quest to slay a dragon that’s been threatening the local village for years. I now want to prototype some of the aspects of the game like combat.
  + **Prototype example plan for 2D RPG attack prototype**
    - Start by creating a basic scene and playing the play sprite inside.
    - Add keyboard controls to the player sprite.
    - Create a collide-able object and have it work correctly with the player (the player should not be able to move through a rock for instance
    - Create a weapon sprite on character.
    - Create attacking hitbox / button control (animations can be a different beast that we’ll save for a different day, for now, just have it play a sound and change from one sprite to another like a “2 frame animation”)
    - Create an enemy sprite and add collision.
    - Create a means to defeat an enemy (if the player hits the attack and the collision of the sword hits the collision of the enemy, remove the enemy object)
  + Use steps similar to these to create a basic prototype for your game.
    - Think about the high-level overview of the game
    - Sketch up your main characters
    - Sketch up a mob or enemy
  + Be sure to ask questions when needed.
* Now that you have a character or some things for a scene in mind how do you make it a reality?
* <https://www.piskelapp.com/>
* Hilight features of the app
  + Lighten/darken
  + Mirror tool
* Go over basic drawing concepts
  + Shading and highlights with a sphere
  + Not everything will be perfect!
  + 
* Exporting images
* Animations

**ACTIVITY:** Think-Team-Share Brainstorming characters for game (approximately 25 minutes)

Students break into their groups to begin brain storming character ideas for their final game

(Have them have their phone ready and have a way to capture any drawings that are done on paper. )

**INSTRUCTION**: Ask if there are any final questions for the day before introducing homework

**HOMEWORK:** Continue Brainstorming characters and game

* Continue to think about your characters, game, non-player characters or objects that you would want in your final game
  + Explain what your game will be about with broad detail (e.g. do you want it be an action game, a puzzle game, about farming or going on a quest for riches)
  + Write out some game elements that you might want to have, things like:
    - Goals
    - Storylines
    - Rules
    - Players