The goal of this course is to teach students about different game mechanics found throughout the industry. This course also focuses on learning how to organize your code to make it easier to follow and easier for others to work with. Last, this course introduces coding concepts like functions and local variables that begin the conversation of encapsulation.

By the end of this course, students will be have a working game from start screen to game over screen. They will then be able to use these skills to create their own awesome games.

**Lesson 1-The Player**

In this lesson, we will create a player avatar by using the provided ship and thruster assets. This ship will be able to move and shoot. We will also add a background that places our spaceship in space.

Vocabulary used: Window, Layout, Layer, Sprite, Layout Area, Behavior, Event Sheet, Group, Event, Action

* Begin the lesson talking about the origins of Asteroids. https://en.wikipedia.org/wiki/Asteroids\_(video\_game)
* Open Construct 2 and create a new empty project.
* In the “Projects” window, right-click on the “New Project” folder and rename it “Asteroids”
* Have the students visit [**http://bit.ly/2iX3pRo**](http://bit.ly/2iX3pRo) to download the “freebundles” folder and save it on their desktop. All of the assets used in this game will be in this folder.
* Setting the Layout and Window Sizes
  + First we will change the Window Size. This setting controls the dimensions of the window in which your game is played.
    - Again, select the newly named “Asteroids” folder from the “Projects” window. With that folder selected you can see the properties of the project in the “Properties” window.
    - Find the “Window Size” setting in the “Project settings” section of the “Properties” window. Click on the width and height numbers to the right and set the “Window Size” to 800, 800.
  + Now we will change the the Layout Size. This setting controls how big the game world is. If it is bigger than the window size, then some of your game will be off screen while you play. This works in many games like platformers or shooters, but for Asteroids we want our window and layout to be the same size.
    - Under the “Layouts” folder in the “Projects” window you will see the Default “Layout 1”. This Layout is where we will be creating our main game Layout. For that reason, let’s rename this Layout to “Game”.
    - Right-click on “Layout 1” and rename it to “Game”.
    - With the Game Layout selected you will be able to see the settings of the Game Layout.
    - Find the “Layout Size” setting in the “Layout properties” of the “Properties” window.
    - Click on the width and height numbers to the right and set the “Layout Size” to 800, 800, guaranteeing no scroll while playing.
* **Adding a Background**
  + Underneath the “Projects” window, you should see 2 tabs: “Projects” and “Layers”. If you do not see this, make sure that “Layer Bar” is selected in the View tab.
  + Rename the current Layer to “Game” by double-clicking it. This is where all the assets for the gameplay will go.
  + Set the Transparent property of the Game Layer to “yes”. This allows us to see the Background Layer, once we add it.
  + In the Layers menu click the “+” button to add a new Layer.
  + Name that Layer “Background”
  + Now, reorder the new Background layer so it sits below the Game layer by clicking and dragging it in the Layers menu. This makes sure that anything placed on the Game Layer will not be obscured by the Background Layer.
  + With the Background Layer selected, double-click in the Layout area.
  + Click the “Sprite” button from the list of assets that pops up to add a sprite.
  + Scroll to the bottom of this list and rename your sprite “Background”, then click “Okay”.
  + Click on the Layout area once more after the popup closes. This adds the sprite to your Layout and opens the Sprite Editor.
  + Upload a Sprite from the “freebundles” folder. The students can use any sprite, but using a space sprite will fit the theme of the game.
  + Click the “X” button on the top right hand side of the Sprite Editor to close it.
  + Now resize your Background Sprite so it covers the entire Layout
    - You can either select the “Background” sprite. Then find the “SIze” property in the “Common” section of the “Properties” window.
    - Select the width and height numbers to the right of this property and set it to 800, 800 to match your “Window” and “Layout” size.
    - The other way to accomplish this is to select the “Background” sprite and use the anchors on the edge of the sprite to resize it.
  + Last, click the “Lock” next to the Background Layer in the Layer menu to lock this layer. Locking a layer prevents the editing of that layer. This is important for the background because it takes up our entire screen. In Construct 2 we add assets to the screen by double-clicking our layout. If the Background Layer is not locked, we will only be able to edit the Background sprite once we add it into the project
* **Adding the Player**
  + Make sure that you have the “Game” Layout selected. The player is part of the game so it makes sense to go on this layer.
  + Now double click on the Layout Area(it should appear that you are double clicking on the background sprite that you added in the above steps).
  + Just like with the Background sprite, find the “Sprite” option in the pop-up menu and click it.
  + Scroll to the bottom of the list and name the new sprite “Player” and click “Okay”.
  + Click on the Layout Area at the location you want the player to be added.
  + Now the Sprite Editor should pop up again.
  + Upload a Spaceship for the player from the “freebundles” folder.
  + We will not be using an animated ship in this project so we do not need to worry about adding additional frames.
  + Click the “X” in the top right of the Sprite Editor to close it.
  + Now we need some Behaviors.
  + Make sure you have the “Player” sprite selected.
  + Click on the blue “Behaviors” link under the “Behaviors” section of the “Properties” window.
  + Click the “+” button on the “Behaviors” pop-up
  + Double-click the “Wrap” behavior from the list. This will allow the player to wrap around the screen.
  + Do the same for “Custom Movement”
* **Adding Thrusters**
  + Now we will add the reverse and forward thrusters. These will give a visual representation of accelerating and decelerating to the player.
  + Double-click the Layout Area and select “Sprite” from the pop-up just like above.
  + Rename it “Forward Thruster”. Then click “Okay”.
  + Click on the Layout Area one more time to place the forward thruster in the Layout.
  + In the “Sprite Editor” pop-up, make sure that your thruster is pointing in the same direction as your ship. If not use the “Rotate” buttons  to adjust their rotation.
  + Once rotated properly, close the “Sprite Editor”.
  + In the Layout Area, position your thruster where you want it to show up on your ship.
  + We need a way to keep the thruster connected to the ship when it moves to do this, we need the “Pin” Behavior.
  + With your thruster selected, click the blue “Behaviors” link in the “Behaviors” section of the “Properties” window.
  + Click the “+” button to add a new “Behavior”.
  + Select the “Pin” Behavior from the pop-up.
  + Now we will be able to control the position of the thrusters easily.
  + If you need more than 1 thruster, select the thruster press CTRL+C and then CTRL+V to copy and paste a new thruster, then select where you want it to be in the Layout Area.
  + Follow these same steps to create your “Reverse Thruster”.
* **Adding Input Detection**
  + Now that there is a player we need to be able to move it.
  + Double-click the Layout Area and choose “Keyboard” from the pop-up options.
  + This adds the keyboard plugin to your game. You can now check for input from the keyboard in your event sheets.
* **Adding a Laser**
  + We need a laser sprite for the player to shoot.
  + There are many options available in the “freebundles” folder for lasers and bullets. Have the students choose from them.
  + Just as before, add a sprite by double-clicking the Layout Area, choosing “Sprite” from the options, name it “Laser” and press “Okay”. Then click on the Layout Area one more time to place that sprite in your Layout.
  + Upload the chosen sprite into the Sprite Editor.
  + Change the origin point to be the end of the Laser to ensure that the Laser’s position is based on the end of the sprite, not the center.
    - Double-click on your Laser sprite to open the Sprite editor. I the tools section find the “Origin” button .
    - Select the “Origin” button and this will bring up the “Image points” pop-up. This pop-up is used to set the position of your origin as well as adding new image points that can be used as spawning locations for other objects.
    - In the “Image points” pop-up, select the “Origin”.
    - In the “Sprite Editor”, use your mouse to click and drag the origin point to the end of the Laser. Now when we spawn this bullet later, it will not spawn in the middle of the sprite.
* **Event Sheets**
  + Event Sheets are where you write code. Any Layout can be controlled by any Event Sheet. To stay organized, let’s rename this Event Sheet to make sure we know which Event Sheet to add code to for the main part of our game.
  + In the “Projects” window, right-click “Event Sheet 1” from the “Events” folder.
  + Select “rename” and change the name to “esGame”: “es” is short for “Event Sheet”.
* **Coding Thrusters Pinned to Player**
  + Make sure you are in “esGame”.
  + Click the “Add Event” button.
  + Double-click the “System”(gear) icon.
  + Double-click “On start of layout” event.
  + Click the “Add Action” button of the new event created.
  + Double-click the “Forward Thruster” icon.
  + Double-click the “Pin to object” option.
  + Click on <click to choose> next to the “Pin to” option in the Pop-up window.
  + Select the “Player” from the next pop-up. Then click “OK”.
  + Make sure that “Mode” is set to “Position & angle”. Click “Done”.
  + Add the same Action for the “Reverse Thruster”.
* **Coding the Player Movement**
  + Make sure you are in “esGame”.
  + First, we are going to create a Group. Groups allow us to keep our code organized. This increases readability and makes the game easier to add to and control. Groups can be disabled or enabled, which allows us to prevent code from running when we don’t want it to.
  + Any time you want an event to be contained within this group, it needs to be a subevent of that group. Right click on the group, highlight the “Add” option and choosing “subevent” from the options listed.
  + Right click on some empty space on the Event Sheet.
  + Highlight the “Add” option and choose “Group” from the list of options.
  + Name that group “Player Movement”.
  + Add a subevent to the “Player Handling” group.
  + Click the “Keyboard” from the pop-up list. Click “Next”.
  + Click the “Key is down” event. Click “Next”.
  + Click <click to choose> next to “Key”.
  + Press the “Up Arrow” key. This automatically adds that key to the Event. Click “Ok”. Click “Done”.
  + Click “Add Action” next to your new Event.
  + Click the”Player” from the pop-up and click “Next”.
  + Click “Accelerate toward angle”. Click “Next”.
  + Set “Acceleration” to 100.
  + Set “Angle” to Self.Angle. Click “Done”. Talk about what is happening here. Mention that the player is an “Object” and “Angle” is a property of the Player “Object”.
  + Add another Action to this Event
  + Select the Forward Thrusters Object. Click “Next”.
  + Select “Set Visible”. Click “Next”.
  + Set “Visibility” to “Visible”. Click “Done”.
  + Add a subevent to the “Player Handling” group.
  + Click the “Keyboard” from the pop-up list. Click “Next”.
  + Click the “On key release” event. Click “Next”.
  + Click <click to choose> next to “Key”.
  + Press the “Up Arrow” key. Click “Ok”. Click “Done”.
  + Click “Add Action” next to your new Event.
  + Select the Forward Thrusters Object. Click “Next”.
  + Select “Set Visible”. Click “Next”.
  + Set “Visibility” to “Invisible”. Click “Done”.
  + Add a subevent to the “Player Handling” group.
  + Click the “Keyboard” from the pop-up list. Click “Next”.
  + Click the “Key is down” event. Click “Next”.
  + Click <click to choose> next to “Key”.
  + Press the “Down Arrow” key. Click “Ok”. Click “Done”.
  + Click “Add Action” next to your new Event.
  + Click the”Player” from the pop-up and click “Next”.
  + Click “Accelerate”. Click “Next”.
  + Make sure “Which” is set to “Forwards”.
  + Set “Acceleration” to -100. Click “Done”.
  + Add another Action to this Event
  + Select the Reverse Thrusters Object. Click “Next”.
  + Select “Set Visible”. Click “Next”.
  + Set “Visibility” to “Visible”. Click “Done”.
  + Add a subevent to the “Player Handling” group.
  + Click the “Keyboard” from the pop-up list. Click “Next”.
  + Click the “On key release” event. Click “Next”.
  + Click <click to choose> next to “Key”.
  + Press the “Down Arrow” key. Click “Ok”. Click “Done”.
  + Click “Add Action” next to your new Event.
  + Select the Reverse Thrusters Object. Click “Next”.
  + Select “Set Visible”. Click “Next”.
  + Set “Visibility” to “Invisible”. Click “Done”.
  + Add a subevent to the “Player Handling” group.
  + Click the “Keyboard” from the pop-up list. Click “Next”.
  + Click the “Key is down” event. Click “Next”.
  + Click <click to choose> next to “Key”.
  + Press the “Left Arrow” key. Click “Ok”. Click “Done”.
  + Click “Add Action” next to your new Event.
  + Click the”Player” from the pop-up and click “Next”.
  + Select “Rotate counter-clockwise”. Click “Next”.
  + Set “Degrees” to 125 \* dt. Click “Done”. Explain delta time.
  + Add a subevent to the “Player Handling” group.
  + Click the “Keyboard” from the pop-up list. Click “Next”.
  + Click the “Key is down” event. Click “Next”.
  + Click <click to choose> next to “Key”.
  + Press the “Right Arrow” key. Click “Ok”. Click “Done”.
  + Click “Add Action” next to your new Event.
  + Click the “Player” from the pop-up and click “Next”.
  + Select “Rotate clockwise”. Click “Next”.
  + Set “Degrees” to 125 \* dt. Click “Done”. Explain delta time.
  + 3
  + Add a subevent to the “Player Handling” group.
  + Click the “Keyboard” from the pop-up list. Click “Next”.
  + Click the “Key is down” event. Click “Next”.
  + Click <click to choose> next to “Key”.
  + Press the “Spacebar” key. Click “Ok”. Click “Done”.
  + Click “Add Action” next to your new Event.
  + Click the “Player” from the pop-up and click “Next”.
  + Select the “Spawn another object” option. Click “Next”.
  + Click <click to choose> next to “Object”.
  + Select the Laser object. Click “Ok”.
  + Set “Layer” to your game layer.
  + Keep “Image point” at 0. Click “Done”.
  + Add another Action to this Event.
  + Select the Laser Object. Click “Next”.
  + Select “Move Forward”. Click “Next”.
  + Set “Distance” to 48. This may change depending on your laser or ship.
  + Below is an example of what your events should look like. Ignore the audio lines,we will cover those later.



* + Now you should have a spaceship that moves around the screen and wraps to the other side when it goes off screen.

**Lesson 2-The Asteroid**

In this lesson, we will be creating an asteroid. This asteroid will move across the screen, wrap around the screen.

Vocabulary used: Scope, Encapsulation, Object, Group, Event, Action, Sub-event, Instance Variable, Local Variable, Layout, Event Sheet, Behavior,

* **Adding the Asteroid**
  + Have the students choose an asteroid from the “freebundles” folder. Then have them add it to their “Game” Layer of their “Game” Layout.
  + Name that Sprite “Asteroid”.
  + With the asteroid selected, click the blue “Behaviors” link under the “Behaviors” section of the “Properties” window.
  + Add the “Wrap” Behavior to the asteroid.
  + Add the “Bullet” Behavior to the asteroid.
  + Add the “Rotate” behavior to the asteroid.
  + We want the bullet behavior to control forward motion but not rotation.
  + Select the Asteroid. In the “Properties” window, set the Bullet Property “Set angle” to “No”
  + Next, we need an Instance Variable to control the size of the asteroid when it splits. There are 3 kinds of variables in Construct2: Global, Local and Instance. Global variables can be used by any line of code you write. This is good for a score, level number, or anything else you want multiple sprites to influence. Local variables are very similar to Global variables. They can be used by any sprite. However, they cannot be used by any line of code.These variables can only be used within the event they are added to. This is called Scope. If you are trying to access a Local variable from one event in a different event, you are out of the Scope of that variable. Instance variables are only accessible by the Sprite it has been added to. This is another example of Scope This is called Encapsulation. We use encapsulation to prevent bugs from happening. In this case, the asteroid is the only object that should change it’s size. If, for example, the player changed the size of the asteroid, then the asteroid would change size without being hit by the laser. To prevent this, we encapsulate the Size variable within the asteroid object to guarantee nothing else can change that variable at the wrong time. Instance variables are good for health, size, speed and any other variable you want to be controlled by only one object.
  + Select the Asteroid.
  + In the “Properties” window find the “Instance variables” section.
  + Click the blue “Instance variable” link to add a new instance variable.
  + Click the “+” to add a new variable.
  + Set “Name” to Size.
  + Set “Type” to Number.
  + Set “Initial value” to 1.
  + Click “OK”.
  + This variable stores an size ratio. Every time the asteroid gets hit, this ratio should change.
* **Coding the Asteroid Handling**
  + Go back to the “esGame” Event Sheet.
  + Add a new Group called “AsteroidHandling”.
  + Add a sub-event to this Group.
  + Select the Asteroid Object then Click “Next”.
  + Select “On collision with another object” then Click “Next”.
  + Click <click to choose>.
  + Select the Laser object from the pop-up. Click OK. Click Done.
  + Add an action to this new Event
  + Select the Laser Object. Click “Next”.
  + Select “Destroy”. Click “Done”.
  + Local Variables
    - Now we get to use local variables. Remind the class about the importance of encapsulation.
    - Right-click on the “Asteroid On collision with Laser” Event.
    - Highlight the “Add” option and select “local variable”.
    - Set “Name” to CurrentUID.
    - Set “Type” to Number.
    - Set “Initial value” to 0.
    - Click “OK”.
    - Follow the above steps to create local variables called “CurrentAOM” and “CurrentSize”.
  + Right-click on the “Asteroid On collision with Laser” Event.
  + Highlight the “Add” option and select “Add blank Sub-Event”.
  + Add an Action to the new blank Sub-Event.
  + Select the System object. Click “Next”.
  + Select “Set value”. Click “Next”.
  + Set “Variable” to CurrentUID.
  + Set “Value” to Asteroid.UID. Click “Done”.
  + Add an Action to the new blank Sub-Event.
  + Select the System object. Click “Next”.
  + Select “Set value”. Click “Next”.
  + Set “Variable” to CurrentAOM.
  + Set “Value” to Asteroid.Bullet.AngleOfMotion then Click “Done”.
  + Add another Sub-Event to the AsteroidHandling Group.
  + Select the Asteroid Object then Click “Next”.
  + Select “Compare instance variable” then Click “Next”.
  + Set “Instance variable” to “Size”.
  + Set “Comparison” to “Less or equal”.
  + Set “Value” to 2.5 then Click “Done”.
  + Now we need a Global Variable to keep track of our score.
  + Right-click on any empty space inside the “esGame” Event Sheet.
  + Click “Add global variable”.
  + Set “Name” to “Score”.
  + Set “Type” to “Number”.
  + Set “Initial value” to 0. Click “OK”. Now we can add to this variable.
  + Back in the “Asteroid Size <= 2.5” Event, add a new Action.
  + Select the System Object then Click “Next”.
  + Select “Add to” then Click “Next”.
  + Set “Variable” to Score.
  + Set “Value” to 1 then Click Done.
  + Right-click on the Asteroid Size <= 2.5 event. Highlight “Add” and select “Else”.
  + Else always comes after an if statement. IF a condition is met run code ELSE run other code. Elses cannot be added without an initial if statement.
  + Add an Action to this new “Else” Event.
  + Select the Asteroid Object and click “Next”.
  + Select “Destroy” and click “Done”.
  + Add another Action to this event.
  + Select the System Object and click “Next”.
  + Select “Add to” and click “Next”.
  + Set “Variable” to Score.
  + Set “Value” to 10. Click Done.

**Lesson 3 - The Asteroid: Splitting and Waves**

Up to this point, we have created a player that can be controlled. We built an asteroid that can move. We can shoot the asteroid, but it will not be destroyed. The next step is to make the asteroid split if it is hit by the laser. After being split enough times, the asteroid will just be destroyed. Then the player’s score will increase. After we have one asteroid behaving properly we will add in a wave system to make the game progressively harder.

Vocabulary: int, Function, Modulus, Event, Action, Sub-event, Object, Instance Variable, Local Variable, Global Variable

* Splitting the Asteroid
  + First, we will need a Global Variable.
  + Right-click on some empty space inside your esGame Event Sheet.
  + Click “Add global variable”.
  + Set “Name” to AsteroidMaxSpeed.
  + Set “Type” to Number.
  + Set “Initial value” to 50.
  + Click “OK”.
  + Right-click the “Asteroid Size <= 2.5” Event then add a new Sub-event.
  + Select the System object then click “Next”.
  + Select the “Repeat” option then click “Next”.
  + Set “Count” to 2 then click “Done”.
  + Add an Action to the “Repeat 2 times” Sub-event.
  + Click on the System object and then click “Next”.
  + Select “Create Object” then click “Next”.
  + Select <click to choose> then double-click on the Asteroid object.
  + Set “Layer” to your “Game” Layer.
  + Set “X” to Asteroid.X.
  + Set “Y” to Asteroid.Y.
  + Click “Done”.
  + Add another Action to the “Repeat 2 times” Sub-event.
  + Select the Asteroid object then click “Next”.
  + Select “Set angle of motion” then click “Next”.
  + This is a little mathy so make sure the kids get this exactly correct. Set “Angle” to int((CurrentAOM + random(125))%180). This is a good time to go over modulus, int, and functions.
  + Add another Action to the “Repeat 2 times” Sub-event.
  + Select the Asteroid object then click “Next”.
  + Select “Set size” and click “Next”.
  + Set “Width” and “Height” to 96 / CurrentSize then click “Done”.
  + Add another Action to the “Repeat 2 times” Sub-event.
  + Select the Asteroid object then click “Next”.
  + Select “Set value” then click “Next”.
  + Set “Instance variable” to Size.
  + Set Value to CurrentSize then click “Done”.
  + Add another Action to the “Repeat 2 times” Sub-event.
  + Select the Asteroid object then click “Next”.
  + Select “Set speed” under the “Rotate” section then click “Next”.
  + Set “Speed” to random(20, 180) then click “Done”.
  + Add another Action to the “Repeat 2 times” Sub-event.
  + Select the Asteroid object then click “Next”.
  + Select “Set speed” under the “Bullet” section then click “Next”.
  + Here is where we will use the global variable we createdSet “Speed” to random(AsteroidMaxSpeed - 10, AsteroidMaxSpeed).