Tasks							
Index	**Title**	**Description**	**Requirements**	**Sub-Tasks**	**Dependencies**	**Member**	**Priority** (1-3) **Due Date**
0	Project setup	Before the project can be started, it will be necessary for a single member to setup the project structure, with the different packages, dependencies GitHub setup, collaborator access, and make sure all the members can effectively cooperator	- Gradle - JavaFX - GitHub - FXProjectTemplate	0.1 Create Project 0.2 Dependencies 0.3 GitHub		Ethan	- Priority: 3 - 10/6/2023
1	Starter screen	This will be the introduction page when the user initially launches the application. It will comprise of a layout of buttons allowing the user to choose what they want to do, and where they can go	- JavaFX - FXML - Scene builder	1.1 FXML 1.2 Buttons	0 Project Setup	Mackenzie Zachary	- Priority: 3 -10/12/2023
2	Drag-and-Drop scene layout	This feature will give the user the ability to customize what structures they are launching the projectile at. It will comprise of a layout of shapes that the user can drag out of the window and into the projectile motion screen, as well as the menu bar with different actions	- JavaFX - FXML - OpenGameArt.org	2.1 Menu bar 2.2 Side panel	1 Starter Screen	Anton Ethan	- Priority: 3 - 10/16/2023
3	Projectile Selection Window	This feature will comprise of a layout of images acting as buttons allowing users to choose what projectile will launch. This scene is a popup modal window that appears when the user clicks on the projectile.	- JavaFX - OpenGameArt.org	3.1 UI elements	2 Drag-and-drop scene layout	Mackenzie	- Priority: 2 - 10/24/2023
4	Sound Customization Window	Similar to the projectile selection window, this feature will be a layout of buttons allowing users to choose what sound effects and background music they want. This scene is accessible from the starter screen.	- JavaFX - Java Sound API - OpenGameArt.org	4.1 Pane	1 Starter Screen	Anton	- Priority: 1 - 10/31/2023
5	Visual Elements Window	This window will give the user to choose to customize the look and feel of the application such as background color, fonts and so forth	- JavaFX - OpenGameArt.org	5.1 Pane and menu bar	1 Screen screen	Zachary	- Priority: 1 - 10/31/2023

6	Projectile Motion Window	This will be the main feature of the application; it will be the place where the user views the projectile launch and sees the collision with the structure built. This is main part of the drag-and-drop window.	- JavaFX - JBox2D - FXGL - OpenGameArt.org	6.1 Projectile implementation 6.2 Collision implementation	2 Drag-and-drop scene layout	Mackenzie Ethan Anton Zachary	- Priority: 3 - 11/11/2023
7	Parameter customization	This will be the window where people can control the conditions of the simulation under which the motion of projectile is affected	- JavaFX - FXGL - JBox2D - FXML	7.1 UI elements 7.2 Parameter functionality	2 Drag-and-drop scene layout	Mackenzie	- Priority: 2 - 11/18/2023
8	Save and Load feature	This will be a feature that will allow the users to save and keep the launches that they attempted and load them back at a different time. This feature can be accessed from the Drag-and-Drop scene ‘File’ button in the menu bar.	- JavaFX Library - JDBC	8.1 Saving 8.2 Loading	2 Drag-and-drop scene layout 3 Projectile Motion Window	Ethan Zachary	- Priority: 1 - 12/4/2023
9	Custom projectile window	This feature will allow users to create their custom projectile, and input multiple customize various characteristics. It can be accessed when the user clicks on ‘Create custom projectile’ from the starter screen. This feature will allow for saving and loading to ensure that users can keep their creations.	- JavaFX - OpenGameArt.org - JDBC	9.1 UI elements 9.2 Picture upload 9.3 Defining traits for each projectile	1 Starter Screen	Anton	- Priority: 1 - 12/4/2023

Sub-tasks		
Index	**Title**	**Description**
0.1	Create project	Download FXProjectTemplate from class notes and use as template as a starter project. Configure settings.gradle file and rename project directory to match the project name. Create packages for controllers, models, UI classes, tests, and anything else necessary.
0.3	Dependencies	Add all project dependencies, such as FXGL, JBox2D, Java Sound API, and any other external libraries to the build.gradle file so that they can be used in the following tasks
0.2	GitHub	Initialize git repository inside the root of the project directory, set up main branch, add all files to staging area, and push an initial commit with everything inside the directory. Following this, give collaborator access to all team members on GitHub and ensure that all members can properly work on it together.
1.1	FXML	Layout the UI in scene builder to create the FXML file. Link the FXML file to a Java controller. It should comprise of a title, a list of buttons of where the user can navigate to: Drag and drop scene | Music Selection | Visual Elements | Create custom projectile |
1.2	Buttons	Implement the logic for the buttons inside the controller. Each button should create a new scene and switch the stage to the corresponding scene.
2.1	Menu bar	Layout the UI for the menu bar that will stick to the top of the stage window. It should include the following buttons which will be used for different feature inside the drag and drop builder: File | View | Undo | Redo |. The menu bar should also include a back button that will allow the user to navigate back to the starter screen.
2.2	Side panel	Create the UI for the side menu with the different blocks that will be draggable to the main window of the simulator. It should the following blocks: Cubes | Rectangles | Pyramids | Circles, and 2 sliders: Scale | Rotate. The scale and transform sliders will apply after the user has dragged the blocks onto the projectile motion window.
3.1	UI elements	Layout the UI for the popup modal scene with a block that contains different projectiles as an animal's heads and buttons in the window: Save | Add | Default | Close. Add button will permit a user to add new projectile and default button will return to initial type of projectile.
4.1	Pane	Layout the UI for the popup that will contains buttons and sliders. It should include the following buttons which will be used for different features : Save | Add | Default | Close.. Sliders: Sound | Effects that will sets the strength of sound (from 0 to 100%)
5.1	Pane and menu bar	Layout the UI for the popup modal scene with a block that contains buttons in the window: Save | Add | Default | Close and a pane with UI elements that will permit to set the color of the background etc.
6.1	Projectile implementation	This will be the implementation of Newtonian physics into our project. This is what will define the path of projectile motion. This subtask will not feature many UI components as it is more the backend of the project which will be used for defining the animation.
6.2	Collision implementation	The implementation of collision into our project. This is a backend part of our project and will not specifically have any UI elements but will instead determine how UI elements behave through animation and values attained based on collision data.
7.1	UI elements	Implement the UI elements will allow the user to control the settings of simulation. It will includes sliders for changing settings such as gravity, air drag, speed of simulation, etc.
7.2	Parameter functionality	Implement the functionality for the parameters, so that they affect the simulation.
8.1	Saving	Implement the functionality for a user clicking on 'Save' from the dropdown menu from the 'File' button in the drag-and-drop scene. When the user clicks on 'save', use JDBC to create, or update (if file already exists) an SQL file that contains data relevant to each object present on the projectile motion window (such as its position and velocity direction/magnitude, and acceleration).
8.2	Loading	Implement the functionality whereby the user can click on 'Load' from the dropdown menu from the 'File' button in the menu bar. It should pop up a file selection window, where the user can choose which SQL file to load from the user's computer. Once the file is chosen, implement the functionality to load all the different objects onto the projectile motion window.
9.1	UI implementation	Layout the UI from Projectile Selection Window with will contains a block that will change the characteristic of projectile, a block that will set the form of projectile and a block with buttons: Load | Remove | Cancel. Load button will allow the user to search for the picture that will cover the shape. It should have a button for uploading an image, a slider to change the scale (size) of the projectile,
9.2	Picture upload	The implementation of projectile will allow the user to apply a picture on the given shape. User can chose with part of picture will be put on projectile.
9.3	Defining traits for each projectile	A VBox with UI like sliders, text fields and buttons will be provided to modify projectile size, weight, shape and custom trail (check box). User will be able to set the color of the trail.