ACTIVITY **4.2.1**

Code Overview

INTRODUCTION

In this activity, you will work with mostly functional source code for the Emu On The Loose game. The source includes functionality for loading level data, **rendering** the game objects to the screen, and moving the player character with touch interactions. You will work with partners to describe the code's functionality and to create a UML class diagram.

render
Display graphical output to a screen.

Materials

- Computer with Android™ Studio
- Android™ tablet and USB cable, or device emulator
- Tools with which to create a UML class diagram

Procedure

Part I: Set Up LibGDX

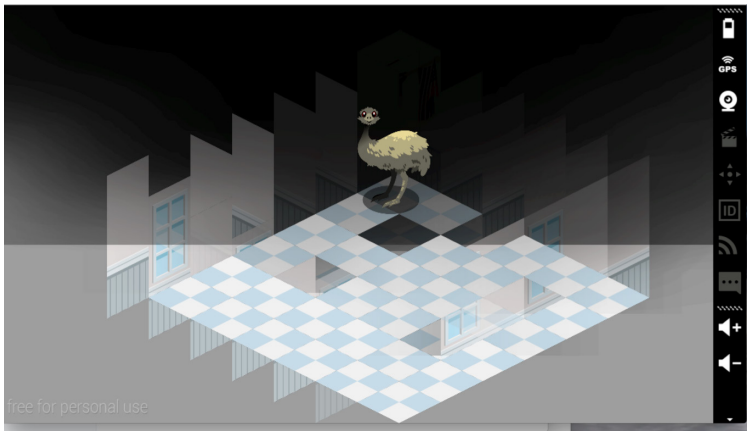
1. Obtain and import *`4.2.1_EmuOnTheLoose_StarterCode`* as instructed by your teacher.
2. On your own, read the comments for the following classes, taking notes as you go. In the next step, you will be asked to work with a team to fully describe the source code:
   a. Model
   b. View
   c. Controller
   d. LevelLoader
   e. Assets
   f. GameObject
   g. GridObject

h. FloorTile
i. WallTile
j. Desk
k. Emu
l. Character
m. PlayerCharacter
n. NonPlayerCharacter

3. Form groups.
4. Share your findings as instructed by your teacher.
5. Within your group, create a UML class diagram for Emu On The Loose.
6. On your own, summarize what each class does as instructed by your teacher.

CONCLUSION

1. The following scenarios (a - f) occurred during the development of the source code for this project. Based on the symptoms described, list the class or classes most likely to need code fixes to solve the problem in each scenario.
   a. When the player character attempted to move into a location with a desk, the desk disappeared and the player character moved into that space.
   b. The player character moved *away* from the location of the touch event.
   c. Instead of moving based on which tile was touched, the player character moved in a direction based on the quadrant where the touch event occurred. Note: the origin is in the middle of the screen, so for example, if the user touched the upper-left corner, the player character moved northwest.
   d. The screen looked as shown in this image:



   e. The player character moved too far on touch events.
   f. The player character moved in the wrong direction on touch events.