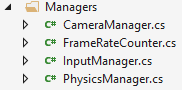
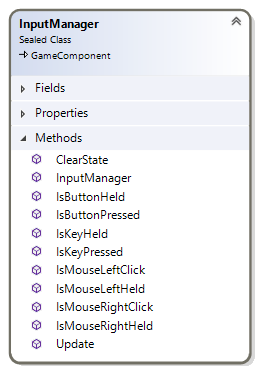
# Engine Project – Managers Folder



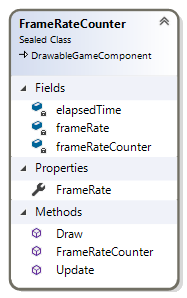
* Create a new folder in the Engine project, call the folder Managers
* Create three new class in this folder
  + CameraManager
  + FrameRateCounter
  + PhysicsManager

# Input Manager

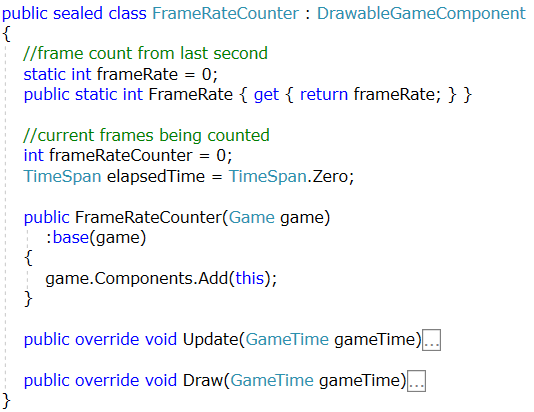
* Download the InputManager from Moodle and add it to the Managers folder.



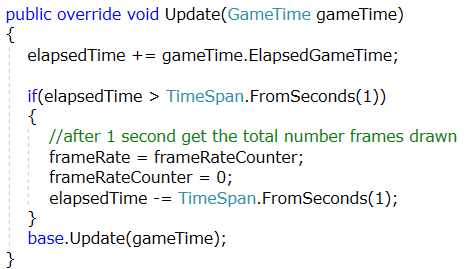
# Frame Rate Counter



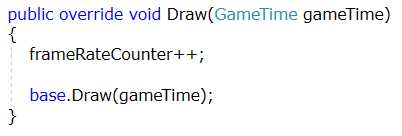
## Code



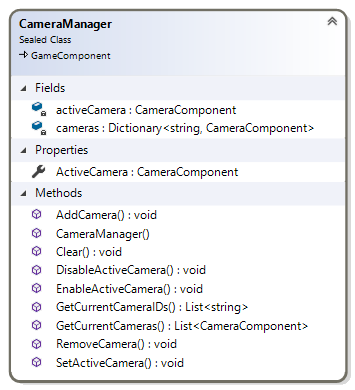
## Update



## Draw



# Camera Manager



## Code



## Set Active Camera (string)

* Accepts a string that should be the ID of the camera we want to make active
* Ensure that the ID of the active camera is not the same as the ID being passed into the method
* Ensure that the passed ID argument is contained within the dictionary of cameras
  + If it is, get the camera from the dictionary, set the active camera to be the retrieved value

## Add Camera (Camera component)

* Accepts a camera component
* If the camera is not already contained in the camera dictionary
  + Add the new camera to the dictionary
* If the newly added camera is the only camera in the dictionary
  + Set this camera as the current active camera

## Clear

* Clear all entries in the cameras dictionary
* Set the active camera to be null

## Remove Camera (String)

* If the dictionary contains the Id, remove the camera from the dictionary
* If the active camera is the camera that was removed
  + Set the active camera to null

## Get Current Camera IDs

* Return all the IDs for cameras in the dictionary

## Get Current Cameras

* Return all camera components in the dictionary

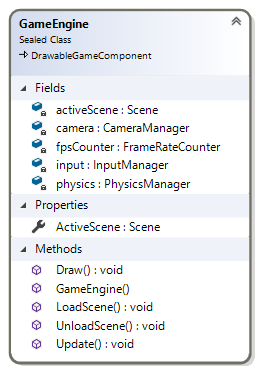
## Enable Active Camera

* If the active camera is not null
  + Set the Enabled property of the active camera to be true

## Disable Active Camera

* If the active camera is not null
  + Set the Enabled property of the active camera to be false

# Game Engine



## Code



## Load Scene (Scene)

* Ensure the new scene is not null
* If there is already a scene set, unload this scene
* Set the active scene to be the new scene
* Initialize the active scene

## Unload Scene

* Set the active scene to be null
* Clear all camera from the CameraManager
* Call GameUtilities.SceneConetent.Unload()

## Update

* Set GameUtilities.Time to be gameTime
* If the active scene is not null, call update on that active scene

## Draw

* If the active scene is not null
  + If the active camera is not null, call draw on the active scene
  + Call DrawUI on the active scene
  + Call GameUtilities.SetGraphicsDeviceFor3D()

# Game 1 – Client Project

* Update the Game1 class in the client project
  + Add a GameEngine variable and instantiate it in the Game1 constructor

