Game	
• Start the game in its Main function • Hold a Level instance • Plan and execute the update and render functions of the Level	• SuperLevel

SuperLevel	
• Hold, update and render a Level • Be able to restart its Level	• Level

Abstract Level	
• Hold a Player instance • Hold and manage an ArrayList of all the Entities • Call the update and render methods of the Player and Entity instances • Be able to return a new instance of itself	• SuperLevel • Player • Entity

Interface Entity	
• Having the functions update and render • Be able to be initialised with a level • Be able to return its BoxCollider ArrayList • Doing the Camera render calls	• Level

Player	Entity
• Reacting to inputs by moving its BoxCollider using euler integration based physics	• BoxCollider • Keyboard • Mouse

Camera	
• Render a Line3d object by projecting it to a Line2d using the functions of its Projector instance and render it on a Graphics instance	• Line3d • Line2d • Projector • Graphics

Projector	
• Project a world relative Vector3d to a Vector2d containing the pixel coordinates using the following 3 steps: • 1. Transform from a world relative Vector3d to a Camera relative Vector3d • 2. Transform from a Camera relative Vector3d to a 2d-plane relative Vector2d • 3. Transform from a 2d-plane relative Vector2d to a Vector2d holding the pixel coordinates	• Matrix3x3d • Vector3d • Vector2d

BoxCollider	
• Manage a static BoxCollider ArrayList • Add itself to the ArrayList on construction • Have a remove function for removing itself from the ArrayList • Have a boolean named Immovable for collision reaction • Have a function called collision returning a boolean (collision detection) • Have a function called move (collision reaction)	• Vector3d

Keyboard	KeyListener
• Update its static variables for the states of all the keyboard keys	

Mouse	MouseListener
• Update its static variables for the mouse x and y position and the state of its buttons	

Vector3d	
• Hold its x, y and z coordinates as doubles • Calculate its magnitude on demand	

Vector2d	
• Hold its x and y coordinates as doubles • Calculate its magnitude on demand	

Matrix3x3d	
• Transform a Vector3d by itself • Statically create Matric3x3d instances for 3d rotation	• Vector3d

Line3d	
• Hold 2 Vector3d instances	• Vector3d

Line2d	
• Hold 2 Vector2d instances	• Vector2d