Student name: Varun Premchandran	Student id: 60001117		
Implemented	Feature	Location of Implementation (wrt solution Directory)	
yes	Game loop	main.cpp	
	Debug Drawing:		
yes	Basic Debug Drawing	Sprite.cpp	
	Object Architecture:		
yes	Object architecture (Component based)	Component class - Abstract class	
yes	Garbage Collection (object management)	Manager/GameObjectInstance.cpp (deletes game Objects that are no longer needed during run time)	
	Communication:		
yes	BasicEvents	Collision Event - Physics Manager.cpp	
yes	Subscription Model	Events only to a particular game object - Event Manager.cpp	
yes	Events & Messaging System	Broadcast events like game over and game win - Event Manager.cpp	
	Data Driven Design:		
yes	Basic Text Serialization	GameObjectManager.cpp	
yes	ObjectFactory (Creation of Objects from data files)	GameObjectManager.cpp - Level Loader and Load Objects	
yes	Level Files	Level1.txt - Level Loader contating object name and file name that its going to load	
yes	Archetypes	GameObjectManager	
yes	advanced serialization/ multiple formats	serializing Game Objects using text file in (GameObjectManager.cpp) and in Shader.cpp for the shaders.	
		Serializing Textures and Vertices of a mesh from text file	
	Frame rate Controller:		
yes	frame rate can be manually controlled	FrameRateController.cpp	
	Input		
yes	Input Manager providing keyboard and mouse key	InputManager.cpp	
yes	and button “IsPressed”, “IsTriggered” and “IsReleased” detection		
	Graphics:		
yes	Hardware accelerated sprite rendering using shaders	Sprite.cpp	
yes	(textured quads/meshes in OpenGL or DirectX)		
yes	Shaders	Shader.cpp	
yes	Texture	Sprite.cpp	
yes	Basic Sprite Transformation	Transform.cpp - Update function using GLM	
No	Sprite Animations		
No	Sprite Z Sorting		
No	Sprite Batching		
Yes	Parallax Scrolling	Background Scrolling	
	Physics:		
yes	Basic collision detection between circles	CollisionManager.cpp	
yes	Basic Collision response	PhysicsManager.cpp	
yes	Collision between different bodies type	CollisionManager.cpp	
No	Impulse base collision response		
No	Advanced physics (rotations, constraint based, stacking, etc…)		
	Game Requirements:		
yes	Human player control of a ship, avatar, or character	Player is stationary according to game requirement. He can shoot from -90 to 90 degrees using right and left key presses and space to shoot.	

yes	Game runs at 30 fps most of the time	Frame time is 16 i.e 60 fps	
yes	Collision between projectiles, player, and enemies	Yes, can be verified using the debug feature	
yes	At least 2 different types of enemies with different behaviors	Homing Behaves in 2 ways, one is a angular velocity and 2nd once the no. of drones reduce its called the kill shot which projects a fast missile onto a drone	
yes	Player has 2 different types of weapons or power ups	Bullets and Kill Shot	
yes	Game must have a win / lose condition	Mission Failure and Success pops. Failure happens if we get killed and our HP goes less than 50. Each drone banging the gunship costs 5 HP damage	
		Mission Success happens on killing any 9 drones	