Team Name: Road Killer

List of class personnel:
Victor Shu: Advanced Graphics
Weihao Yan: Advanced Physics
Yun Jiang: Audio

Description of Engine:
Our engine is an engine specified in driving simulator game. There are several topics that we are interested in. The physics system is an important part of a driving simulator game. We need a physics system that is particularly aimed at vehicles or cars. Audio is also essential in a driving simulator game. We need to take things like 3D audio and Doppler Effects into account. The car engine sound is also going to be based on this. Other than that, we can have a PBR rendering system that looks good.

Description of the Game:
We are going to build a small driving simulator game with our engine. The game will include a small circular track and a single vehicle that the player can drive. The physics, audio, and graphics will be as realistic as possible.

Details on core systems:
Basically, everyone will be responsible for all parts of the core systems. However, members of the team have different specialties and it will be better if we have different focuses.
Victor is responsible for the rendering system and low-level interactions with APIs such as Direct3D, XInput or XAudio2. The engine will be built on the framework. Victor is also going to implement the camera control, which is really a part of the rendering system. Weihao is going to implement the simple rigid body physics, input system, scene graph, and file representation. Yun is responsible for event management and the game loop.

Details on the additional system:
Victor is responsible for advanced PBR rendering system. Ideally, the game will look like any other realistic racing games. Weihao is responsible for advanced physics. The advanced physics is aimed at vehicle movement. Yun is responsible for the audio system. The audio system will handle things like car engine sound modulation and other 3D audio effects. We might also have a particle system if we have enough time for it.

Milestone 1:
We will have our core systems ready. By saying ready, we mean that we can use what we have to make a very simple game with very simple interaction like roll a ball game.

Milestone 2:
We will have our engine and the game ready. We will complete implementing all of the advanced systems that we need. During Milestone 1 and Milestone 2 we will keep testing the

functionalities of the engine by building our desired game concurrently and some small testing games.