1.

- There must be a player car
- There must be adversarial cars
- The player must be able to move his car with the mouse
- Cars need to be able to collide with each other
- After a collision, one of the colliding cars should stop moving
- The Player should be able to stop and start the game
- The game should have music playing in the background
- The game should play a sound when two cars crash

2.

- The game should run smoothly (i.e. no noticable slow-downs or speed issues)
- The game should not crash
- The source code for the game should be maintainable
- The game should be compatible with multiple platforms
- User friendliness

3.

As-is scenario: The user opens the game, klicks the start button, and then, attempts to crash as many other cars on the board as possible. After failing to do this for a few times, he gets increasingly angry and changes his cars speed, making it much higher. This gives him a crucial advantage, which he uses to defeat the other cars. Having accomplished this, he gets bored and continues finishing his homework.

Visionary scenario: The player opens the game and starts the first level. After having defeated half of the cars, the player car gets a significant speed boost, which allows him to defeat the rest of the cars. Having collided with all of them, the screen is cleared and a truck chasing the player car appears. Being able to evade this truck and finally colliding with it from the right, the player manages to succeed to the next level. He then proceeds to waste his entire afternoon with the game.

4.

Use case name	Car crash
Participating actors	Player Car, NPC Car
Entry conditions	The Player Car and the NPC Car have collided
Flow of events	After both cars have collided, the car crash sound is played. Subsequently the car having collided from the left will stop moving.
Exit conditions	If the player car was hit from the right, the player loses the game. Otherwise, the player continues the game and the NPC car is halted.
Special requirements	None.

5.

Extends signifies that something belongs to a certain class, and if this relationship does not exist anymore, the object is transformed in its entirety. The Includes relationship is not this essential, but rather accidental, that means if the relationship doesn't hold anymore, this still doesn't fundamentally change the classes of the objects being talked about. For example in the Bumpers game, the GameBoard includes several cars, but would still be a GameBoard if it didn't, but an OldCar extends a Car.

6.

Aggregation is a collection of objects into one shared entity. Composition is a special form of aggregation, in which the parts don't exist on their own, but just as parts of the aggregated entity. One example for aggregation in Bumpers is the aggregation of NPC cars, but the GameBoard is a composition of this group of cars and the player car.