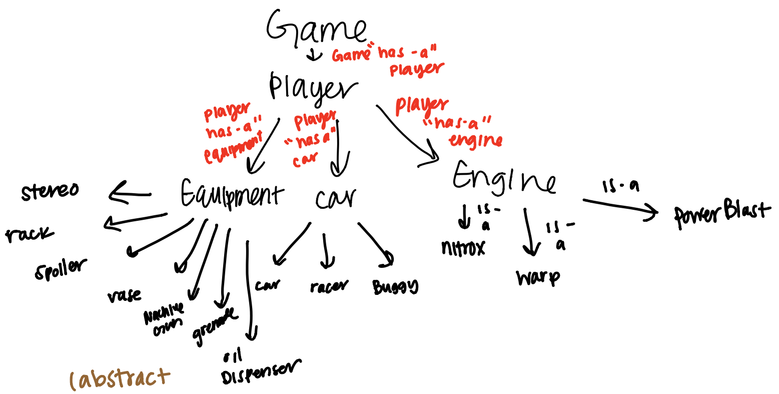

class Game (Game class will keep track of # of players; a simulation of a game)

```
private:
vector<players> player_list;
```

class Player (player stores stats of player. Each player has a vehicle. We have a list of equipment so that we can add to it as new items get acquired)

```
private:
   int experience;
   int strength;
   int control;

   Car vehicle;
   list<Equip> equipment
```

(abstract
Class Engine

```
public:
virtual specifications()=0;

private:
```

(specifications() is virtual so that each child class can override it with their own specifications)

class Nitrox: public Engine

```
public:
  specifications();
```

class Warp: public Engine

```
public:
  specifications();
```

class PowerBlast : public Engine

```
public:
  specifications();
```

(abstract)
class Car

```
public:
virtual   capabilities()=0;

private:
    std::string color;
    Engine engine;
```

((capabilities() are pure virtual b/c their subclasses can override them with their own functions according to car type)

class Truck: public Car

```
public:
  capabilities();
```

class Racer: public Car

```
public:
  capabilities();
```

class Buggy: public Car

```
public:
  capabilities();
```

(abstract)
class Equipment → (with equipment class we can add more weapons or vehicle add-ons)

```
Private:

Public:
virtual bool weaponOrNot ()=0;
  virtual action() =0;
  virtual   carValue()=0;
```

(action() will be overwritten by dif weapon classes)

class Harpon: public Equipment

```
Private:

Public:
  bool weaponOrNot ();
    action();
```

class machineGun: public Equipment

```
Private:

Public:
  bool weaponOrNot ();
    action;
```

class grenadeLauncher: public Equipment

```
Private:

Public:
  bool weaponOrNot ();
    action()
```

class oilDispenser: public Equipment

```
Private:

Public:
  bool weaponOrNot ();
    action();
```

class Stereo: public Equipment

```
Private:

Public:
  bool weaponOrNot ();
   carValue ();
```

class Rack: public Equipment

```
Private:

Public:
  bool weaponOrNot ();
   carValue();
```

class Spoiler : public Equipment

```
Private:

Public:
  bool weaponOrNot ();
   CarValue();
```

class Vase : public Equipment

```
Private:

Public:
  bool weaponOrNot ();
   CarValue();
```