Vehicle Creation

A vehicle. How can I do that?

Taking into account all the different concepts that you have already learnt, how will you build a car with Bullet library?

Rigidbodies?
Shapes?
Constraints?
Anything else?

btRayCastVehicle

Kester Maddock created a driving model system! We will use it.

In the video, you can see all the parameters that modify the car's response.

Vehicle info struct

```
struct VehicleInfo
{
    ~VehicleInfo();

    vec3 chassis_size;
    vec3 chassis_offset;

    float mass;
    float suspensionStiffness; // default to 5.88 / 10.0 offroad / 50.0 sports car / 200.0 F1 car
    float suspensionCompression; // default to 0.83
    float suspensionDamping; // default to 0.88 / 0..1 0 bounces / 1 rigid / recommended 0.1...0.3
    float maxSuspensionTravelCm; // default to 500 cm suspension can be compressed
    float frictionSlip; // defaults to 10.5 / friction with the ground. 0.8 should be good but high values feels
better (kart 1000.0)
    float maxSuspensionForce; // defaults to 6000 / max force to the chassis

    Wheel* wheels;
    int num_wheels;
};
```

Wheel struct

```
struct Wheel
{
    vec3 connection; // origin of the ray. Must come from within the chassis
    vec3 direction;
    vec3 axis;
    float suspensionRestLength; // max length for suspension in meters
    float radius;
    float width;
    bool front; // is front wheel ?
    bool drive; // does this wheel received engine power ?
    bool brake; // does breakes affect this wheel ?
    bool steering; // does this wheel turns ?
};
```

PhysVehicle struct

```
struct PhysVehicle3D : public PhysBody3D
{
public:
    PhysVehicle3D(btRigidBody* body, btRaycastVehicle* vehicle, const VehicleInfo& info);
    ~PhysVehicle3D();

    void Render();
    void ApplyEngineForce(float force);
    void Brake(float force);
    void Turn(float degrees);
    float GetKmh() const;
public:

    VehicleInfo info;
    btRaycastVehicle* vehicle;
};
```

Vehicle Creation

To create a vehicle, we need to fill in and tweak the different parameters in the *VehicleInfo* struct.

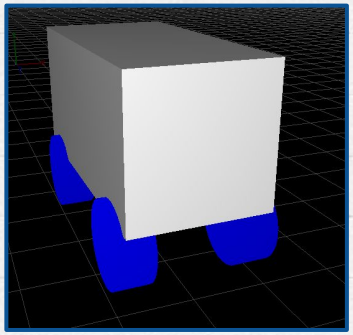
Upon completion, call App->physics->AddVehicle(const &VehicleInfo car);

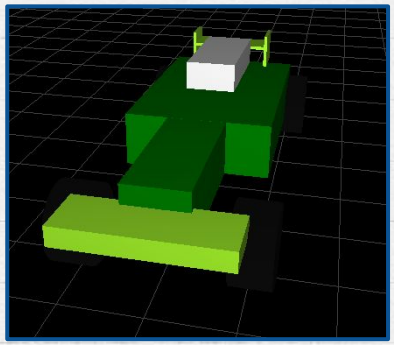
You will control the car in the Player's Update loop.

Improve the car!

Once you understand the code, try to go crazy:

- X Change the chassis of your car: a Formula 1, a jeep, an articulated bus, a van, a forklift?





Improve the car!

Once you understand the code, try to go crazy:

- X Tweak the values: it depends on the type of vehicle that you want

Improve the car!

Once you understand the code, try to go crazy:

- X How many wheels do you need?

The camera

The camera is one of the most important three C's in video games, according to guru Mark Cerny: character, **camera** and control.

How will your camera behave in your game? A static/dynamic panoramic view, a chasing camera...?

The LookAt method on camera will be very useful for this assignment.

Homework (obligatory!!)

On 7th December afternoon, I will evaluate your game and add some notes/advice on your design in order to fulfil the assignment requirements.

Every not checked game can be uploaded as well, but they won't have my approval!

Try to create a first approximation of your game!

Next Week . . .

Sensors