

Vehicle

Code

```
enum VehicleType
{
    VT_NOSQUAD, // kein Einsatzfahrzeug
    VT_THW_FGRR_BKF, // Bergekran
    VT_THW_FGRB_BLF, // Brückenleger
    VT_THW_FGRI_EKW, // Transportfzg. Ingenieur
    VT_FIREFIIGHTERS_ASF, // Abschleppfzg. TODO: ist jetzt THW
    VT_FIREFIIGHTERS_DEKONP, // Dekontaminationsfzg.
    VT_FIREFIIGHTERS_DLK, // Drehleiterfzg.
    VT_FIREFIIGHTERS_RW, // Rüstwagen
    VT_FIREFIIGHTERS_TLF, // Tanklöschfzg.
    VT_FIREFIIGHTERS_LF, // Löschflugzeug
    VT_FIREFIIGHTERS_FLB, // Feuerlöschboot
    VT_FIREFIIGHTERS_LPF, // Löschanpanzer
    VT_FIREFIIGHTERS_FMB, // Feuerwehrmotorboot
    VT_FIREFIIGHTERS_TFMB, // Transportfzg. für FMB
    VT_POLICE_SW, // Sonderwagen
    VT_POLICE_MTW, // Mannschaftstransportwagen
    VT_POLICE_PHC, // Polizeiheli
    VT_POLICE_STW, // Streifenwagen
    VT_POLICE_WAW, // Wasserwerfer
    VT_POLICE_GETAWAY, // Fluchtwagen im Besitz der Polizei
    VT_AMBULANCE_ITW, // Intensivtransportwagen
    VT_AMBULANCE_NEF, // Notarztfzg.
    VT_AMBULANCE_RHC, // Rettungsheli
    VT_AMBULANCE_RHF, // Rettungshundefzg.
    VT_AMBULANCE_RTW, // Rettungstransportwagen
    VT_TAXI,
    VT_BUS,
    VT_DRIVERCAR,
    VT_GANGSTER_GETAWAY, // Fluchtwagen im Besitz von Gangster
    VT_TV_HELI,
    VT_THW_FGRR_RL = 100, // bulldozer
    VT_THW_FGRR_TRL = 101, // Transportfzg. für RL
    VT_THW_FGRT_BH = 102,
    VT_FIREFIIGHTERS_GTF = 200,
    VT_POLICE_GTW = 300,
    VT_AMBULANCE_TRANSEVAC = 400,
    VT_ALL_SQUAD_VEHICLES = 500, // nur der Komplettheit halber ...

    VT_NUMTYPES = VT_TV_HELI + 7
};
```

Alles anzeigen

Code

```
enum DoorActionType
{
    DAT_NONE = vehicle 0x00, doors
    //
    DAT_EQUIPMENT = 0x01,
    DAT_PERSON = 0x02,
    DAT_SPECIAL = 0x03
};
```

Code

```

class Vehicle : public GameObject
{
    Vehicle();
    Vehicle(const Vehicle& Obj_);
    Vehicle(const Actor* Act_);
    Vehicle(const GameObject* Obj_);
    ~Vehicle();
    virtual bool IsValid() const;

    void SetEnergy(float energy_);
    float GetEnergy() const;
    float GetMaxEnergy() const;
    bool IsDestroyed() const;
    bool IsSmoking() const;

    void Destroy(); //!< destroys the vehicle as in an explosion but without the force and pa
    void SetSmoking(bool smoking_); // it starts to smoke and loses energy and will explode
    void SetSmokeLevelDuration(float smokeLevelDuration_);
    float GetSmokeLevelDuration();

    int GetNumPassengers() const;
    PersonList GetPassengers() const;
    bool AddPassenger(Person &Passenger_);
    bool RemovePassenger(Person &Passenger_);
    int GetFreePassengers() const;
    void SetMaxPassengers(int max_, bool discard_ = false);

    bool IsInstalled();
    bool IsUplifted();
    bool IsUplifting();
    bool IsBasketEmpty();
    bool IsCannonInUse();
    bool IsFGRBEmpty();
    bool IsBucketMoving();
    bool IsBucketUp();
    int GetInstallTargetID() const;
    bool HasBoatUploaded();
    bool HasVehicleUploaded();
    bool IsParking() const;
    bool IsDecontaminating() const;

    void SetParking(bool parking_);

    int GetNumTransported() const;
    PersonList GetTransports() const;
    bool AddTransport(Person &Transported_);
    bool RemoveTransport(Person &Transported_);
    int GetFreeTransports() const;
    void SetMaxTransports(int max_, bool discard_ = false);
    const VehicleType GetVehicleType()const;
    void SetVehicleRole(VehicleType type_); //!< Handle with care!

    bool CheckUnloadPossible(GameObject *obj_, const Vector &V_) const;

    bool CheckUnloadPossible(GameObject *obj_, float _x, float _y, float _z);

    bool HasEnclosedPerson() const;
    bool SetEnclosedPerson(const char* name);
    bool RemoveEnclosedPerson();

    float GetValidLandingAngle(GameObject *target_, const Vector &Pos_) const;

    float IsValidLandingPosition(GameObject *target_, float x_, float y_, float z_) const;

    bool IsValidRopeDropPosition(GameObject *target_, const Vector &Pos_) const;

    bool IsValidEngineerDropPosition(GameObject *target_, const Vector &Pos_) const;
https://em-hub.de/lexicon/entry/37-vehicle/
    bool IsValidRopeDropPosition(GameObject *target_, float x_, float y_, float z_) const;

    bool FindReachablePosition(GameObject* obj_, Vector &Pos_) const;

```

Alles anzeigen