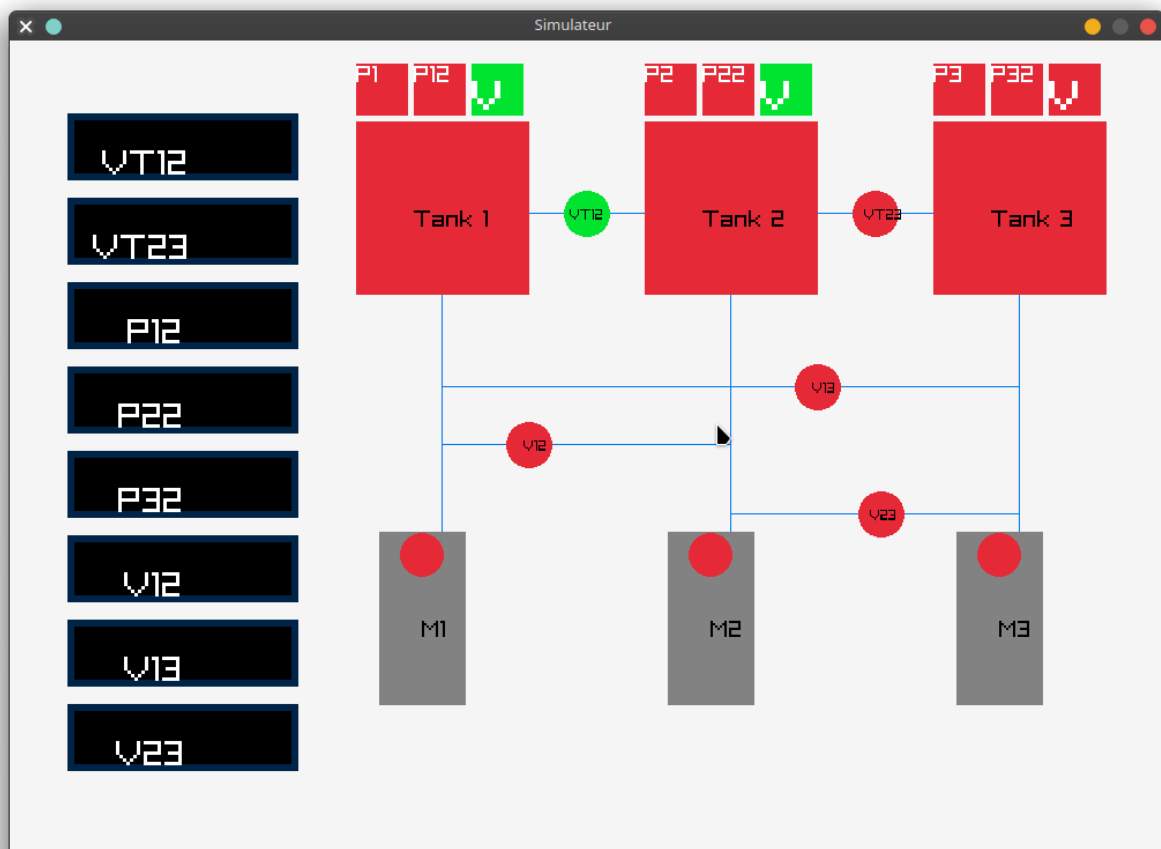


SIMULATEUR



Rappel de l'objectif

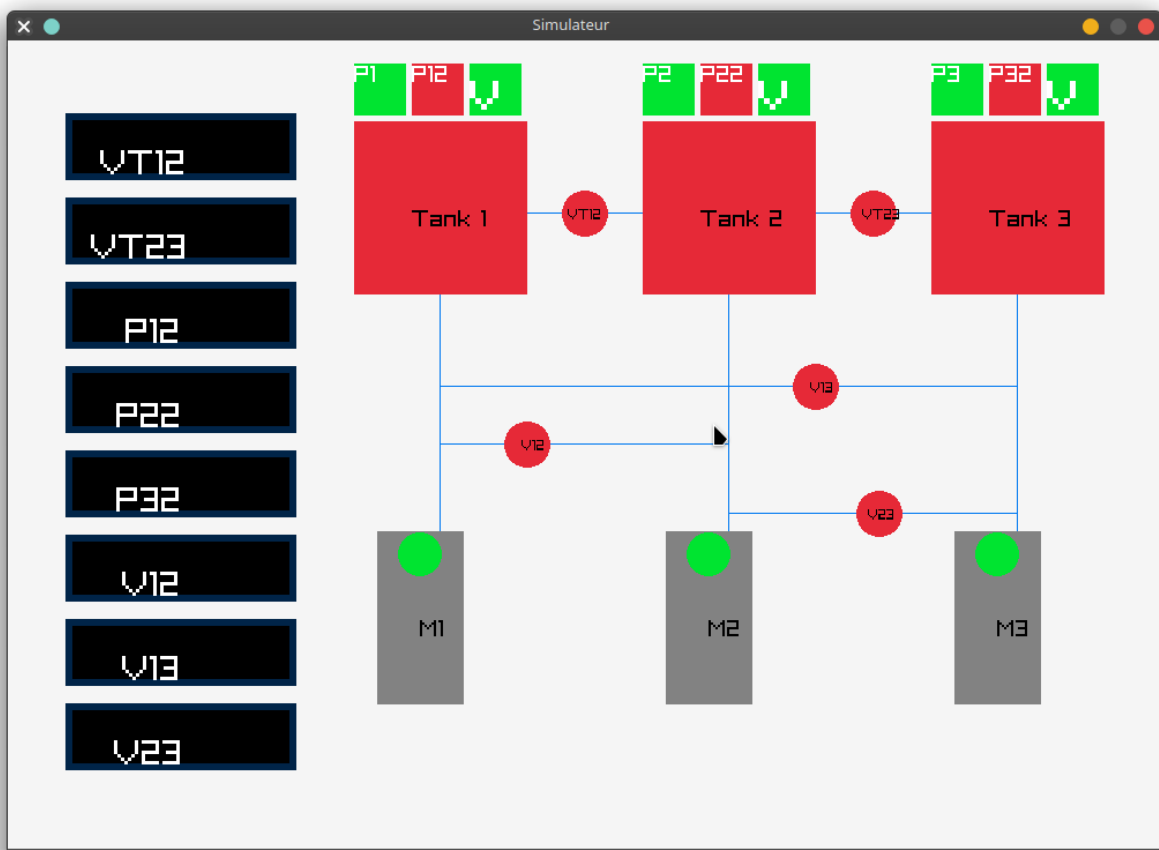
Le but de ce projet est de développer un mini-simulateur du système carburant d'un avion. Ce simulateur a pour but d'entraîner les pilotes à gérer le système carburant en cas de crise.

On utilise une interface graphique nommée l'installation est précisée dans le Readme

Nous verrons les différentes étapes de la réalisation du projet

La fenêtre est séparée entre 2 parties :

- celle de gauche où l'utilisateur peut réparer les pannes du système
- celle de droite où l'on voit en direct l'état du système par des voyants (c'est aussi dans cette partie où l'on insère des pannes)



On crée une classe Launcher ayant comme attributs les objets Vannes,Pompes,Moteur,Tank qui ont chacun un état (booléen true ou false)

```
class Launcher
{
private:
    Moteur M1,M2,M3;
    Tank T1,T2,T3;
    Vanne V12,V13,V23,VT12,VT23;
    Pompe P1, P1s, P2,P2s,P3,P3s ;
    bool panne;
public:
    Launcher();
    Launcher(const char*);
    void Principal();
    void Alimentation();
    void Transfert();
    bool appui();
    void drawTank();
    void drawInterface();
    void drawMoteur();
    void clic();

    ~Launcher();
};
```

Tout d'abord on définit les conditions pour que chaque moteur fonctionne (ex:le moteur 1 a besoin que au minimum l'une de pompes correspondant à son réservoir soit fonctionnel ou qu'une vanne partant d'un autre réservoir actif fasse la liaison)

```
if(
    (( P1.getEtat() || P1s.getEtat() ) && T1.getEtat()) ||
    (( P2.getEtat() || P2s.getEtat() ) && T2.getEtat() && V12.getEtat()) ||
    (( P3.getEtat() || P3s.getEtat() ) && T3.getEtat() && V13.getEtat())
)
M1.marche();
else M1.marchepas();

if(M1.getEtat())DrawCircle(Moteur_M[0].x+Moteur_M[0].width/2, Moteur_M[0].y+20,19,GREEN);
else DrawCircle(Moteur_M[0].x+Moteur_M[0].width/2, Moteur_M[0].y+20,19,RED);
```

On définit aussi le cas de transfert de carburant suite à une vidange (on ne fait le remplissage que si l'un des reservoirs est plein)

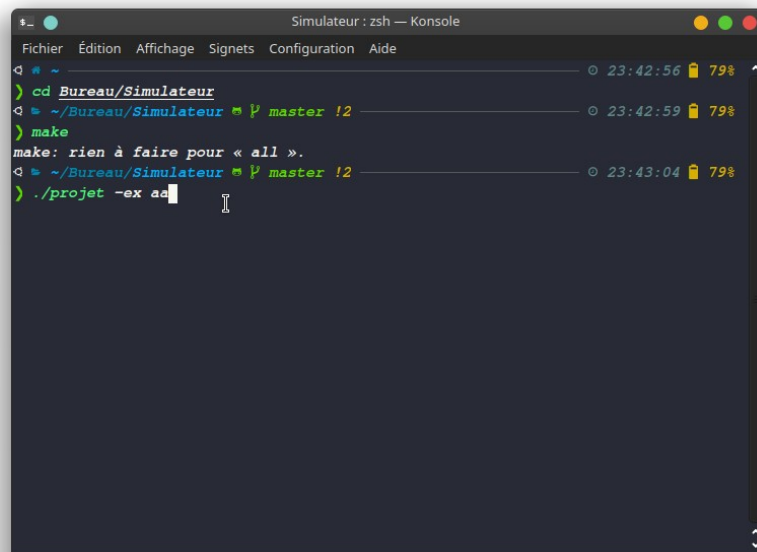
```
void Launcher::Transfert(){  
    if(VT12.getEtat()){  
        if(T1.getEtat())    T2.setEtat(T1.getEtat());    //verif et remplissage  
        else if (T2.getEtat())    T1.setEtat(T2.getEtat());  
    }  
    if(VT23.getEtat()){  
        if(T2.getEtat())    T3.setEtat(T2.getEtat());    //verif et remplissage  
        else if (T3.getEtat())    T2.setEtat(T3.getEtat());  
    }  
}
```

Pour lancer la partie 2 il faut rajouter une option à l'exécutable tel que :

`./projet -ex <login>`

Pour l'instant les login disponibles sont « aa,azerty,q »

Suite à l'authentification le programme cherche le fichier d'exercice(exo modification possible) et crée l'exercice qui correspond au fichier



```
Simulateur : zsh — Konsole  
Fichier  Édition  Affichage  Signets  Configuration  Aide  
~  
> cd Bureau/Simulateur  
~/Bureau/Simulateur % master !2  
> make  
make: rien à faire pour « all ».  
~/Bureau/Simulateur % master !2  
> ./projet -ex aa
```

