

# Dive-PP

This guide explains how to **recreate this project from scratch** in WebGPU Studio (without loading an example).

## 1) Goal and principle

We will create the buffers, paste the WGSL helper functions, write the compute shaders, then configure the Pass.

Steps (in order):

- **Pipeline 1**

## 2) Create a new project

1. Launch WebGPU Studio.
2. Click **New**.

## 3) Create the buffers (Buffers tab)

Create the following buffers (names must match exactly):

- **mat\_param\_model**: size `64×32×1`, type `float`, fill `empty`
- **mat\_param\_simu**: size `16×1×1`, type `float`, fill `empty`

After each change, click **Apply**.

## 4) Add the helper library (Functions tab)

For each entry below:

1. Click **+Add**.
2. Set the name.
3. Paste the WGSL.
4. Click **Apply**.

### ModellIndex

```
// Declaration des positions des parametres de la matrice
//mat_param_model
// ****
const mat_param_model_SX: u32 = 64 ;
const vent: u32 = 0; // débit ventilatoire
```

```

const Vaw: u32 = 1;// volume des voies aériennes
const Valg: u32= 2;// volume du gaz alvéolaire
const Valb: u32 = 3;// volume de sang alvéolaire
const Q:u32 = 4;// débit cardiaque
const Va:u32 = 5;// volume artériel
const Vv:u32 = 6;// volume veineux
const patm: u32 = 7;// pression ambiante
const fn2: u32 = 8;// fraction d azote dans le gaz respiré
// Constantes
const alpha_n2: u32= 9;// coef solubilité azote
const ph2o:u32= 10;// pression partielle de vapeur d eau
const R:u32 = 11;// constante des gaz parfaits
const T:u32 = 12;// température (K)
const kn2:u32= 13;// coef solubilité azote
// Variables (état)
const K1: u32 = 14;// coef de diffusion respiratoire
const K2: u32 = 15;// coef de diffusion alvéolo-capillaire
const K3: u32= 16;
const Vc: u32 = 17;// volume capillaire
const Vt: u32 = 18;// tissue volume
// Pressions partielles
const pp_N2_c_t0: u32= 19;// pression partielle initiale capillaire
const pp_N2_c_t1:u32 = 20;// pression partielle courante capillaire
const pp_N2_ti_t0: u32 = 21;// pression partielle initiale tissus
const pp_N2_ti_t1: u32 = 22;// pression partielle courante tissus
// Air paramètres
const pp_N2_air: u32 = 23;
// Airways paramètres
const pp_N2_aw_t0: u32 = 24;
const pp_N2_aw_t1: u32 = 25;
// Alvéoles gaz paramètres
const pp_N2_alv_t0: u32 = 26;
const pp_N2_alv_t1: u32 = 27;
const pp_N2_alb_t0: u32 = 28;
const pp_N2_alb_t1: u32 = 29;
// Sang veineux paramètres
const pp_N2_v_t0: u32 = 30;
const pp_N2_v_t1: u32 = 31;
// Sang artériel paramètres
const pp_N2_a_t0: u32 = 32;
const pp_N2_a_t1: u32 = 33;

```

## SimulationIndex

```

// Declaration des positions des parametres de la matrice mat_param_simu
// ****

```

```

const mat_param_simu_SX: u32 = 16 ;
const tmp_t: u32 = 0;
const tmp_s: u32 = 1;
const time: u32 = 2;
const dtINI: u32 = 3; // dt global (Euler/RK4/RK6/RK8 selon choix)
const dt: u32 = 4;
const diving_stage: u32 = 5; // itérateur profil de plongée
const iteration: u32 = 6; // pas de simulation en cours

```

## ModelParam

```

fn mat_param_model_init ( ) { // ligne 64
    // Initialisation (uniquement la 1ere ligne : les lignes du dessous
    sont generees automatiquement)
    mat_param_model[vent]= 8.1;// débit ventilatoire
    mat_param_model[Vaw]= 1.5;// volume des voies aériennes
    mat_param_model[Valg]= 1.0;// volume du gaz alvéolaire
    mat_param_model[Valb]= 0.5;// volume de sang alvéolaire
    mat_param_model[Q]= 4.209;// débit cardiaque
    mat_param_model[Va]= 1.7;// volume artériel
    mat_param_model[Vv]= 3.0;// volume veineux
    mat_param_model[patm] = 101325.0;// pression ambiante
    mat_param_model[fn2] = 0.79; // fraction d azote dans le gaz respiré
    // Constantes
    mat_param_model[ alpha_n2 ]= 0.0000619; // coef solubilité azote
    mat_param_model[ ph2o ] = 6246.0; // pression partielle de vapeur d eau
    mat_param_model[ R ] = 8.314;// constante des gaz parfaits
    mat_param_model[ T ] = 310.0;// température (K)
    mat_param_model[ kn2 ] = 0.0000619; // coef solubilité azote
    // Variables (état)
    mat_param_model[ K1 ] = 0.00267;// coef de diffusion respiratoire
    mat_param_model[ K2 ] = 0.00748;// coef de diffusion alvéolo-capillaire
    mat_param_model[ K3 ] = 0.00267;;
    mat_param_model[ Vc ] = 0.5; // volume capillaire
    mat_param_model[ Vt ] = 70.0; // tissue volume
    // Pressions partielles
    mat_param_model[ pp_N2_c_t0 ]= 75112.41;// pression partielle initiale
    //capillaire
    mat_param_model[ pp_N2_c_t1 ]= 0.0;// pression partielle courante
    //capillaire
    mat_param_model[ pp_N2_ti_t0 ]= 75112.41; // pression partielle initiale
    //tissus
    mat_param_model[ pp_N2_ti_t1 ]= 0.0;// pression partielle courante
    //tissus
    // Air paramètres
    mat_param_model[ pp_N2_air ] = 0.0;;

```

```

// Airways paramètres
mat_param_model[ pp_N2_aw_t0 ]= 75112.41;//
mat_param_model[ pp_N2_aw_t1 ] = 0.0;//
// Alvéoles gaz paramètres
mat_param_model[ pp_N2_alv_t0 ] = 75112.41;//
mat_param_model[ pp_N2_alv_t1 ] = 0.0;//
// Alvéoles sang paramètres
mat_param_model[ pp_N2_alb_t0 ]= 75112.41;//
mat_param_model[ pp_N2_alb_t1 ] = 0.0;//
// Sang veineux paramètres
mat_param_model[ pp_N2_v_t0 ] = 75112.41;//
mat_param_model[ pp_N2_v_t1 ] = 0.0;//
// Sang artériel paramètres
mat_param_model[ pp_N2_a_t0 ] = 75112.41;//
mat_param_model[ pp_N2_a_t1 ] = 0.0;
}

```

## SimulationParam

```

fn mat_param_simu_init ( ) {
mat_param_simu[ tmp_t ] = 0.0;
mat_param_simu[ tmp_s ] = 0.0;
mat_param_simu[ time ] = 0.0;
mat_param_simu[ dtINI ] = 0.0009; // dt global (Euler/RK4/RK6/RK8 selon
choix)
mat_param_simu[ dt ] = 0.0009;
mat_param_simu[ diving_stage ] = 1; // itérateur profil de plongée
mat_param_simu[ iteration ] = 0; // pas de simulation en cours
mat_param_simu[ tmp_t ] = 0.0;
mat_param_simu[ tmp_s ] = 0.0;
mat_param_simu[ time ] = 0.0;

}

```

## 5) Create the compute shaders (Compute Shaders tab)

For each shader:

1. Click **+Add**.
2. Set the name.
3. Click **Apply**.
4. Paste the WGSL.

### Shader **Initialize**

Workgroup: **8x8x1**

```
@compute @workgroup_size(1, 1, 1)

fn Initialize(@builtin(global_invocation_id) gid : vec3<u32>) {
    if ( step == 0 ) {
        let index = gid.y * mat_param_model_SX + gid.x;
        if ( index == 0 ) {
            mat_param_model_init( );
            mat_param_simu_init( );
        }
    }
}
```

## 6) Configure the Pass (Pass tab)

Create the pipelines/steps in the following order:

- **Pipeline 1**: dispatch  $1 \times 1 \times 1$

## 7) Compile and run

1. In the **Buffers** tab, select **mat\_param\_model**.
2. View it in **2D** or **3D**.
3. Click **Compile**.
4. Click **Run** (or use **Step**).

## 8) Quick checks (if it doesn't work)

- **Console** tab: read WGSL errors.
- Check buffer **names** match the WGSL code.
- Check buffer sizes (X/Y/Z) and Pass dispatch.

## 9) Save

Click **Save** to export the project as a **.wgstudio** file.