

OBJECTIF : Créer une simulation interactive permettant de comprendre les politiques économiques (monétaire et budgétaire), leurs effets sur la demande globale, ainsi que les arbitrages entre croissance, inflation et chômage.

COMPÉTENCES :

- Identifier une situation conjoncturelle
- Proposer une politique adaptée
- Comprendre les canaux de transmission
- Évaluer les effets des décisions

PRINCIPE DU JEU :

1. Situation initiale : - Récession - Inflation - Hausse des coûts de production. Avec valeurs de départ : taux d'intérêt, dépenses publiques, recettes fiscales.

2. Rôle des élèves :

Décider :

- taux d'intérêt
- dépenses publiques (% PIB)
- recettes fiscales (% PIB)

3. Mécanismes :

- Consommation dépend des impôts
- Investissement dépend des taux
- Dépenses publiques stimulent la demande $\text{Demande globale} = C + I + G$

4. Résultats :

- Croissance
- Inflation
- Chômage

5. Contraintes :

- Inflation cible : 2%
- Déficit < 3%
- Dette < 60%

6. Feedback :

- Explication des mécanismes
- Identification des politiques
- Correction automatique avancée

7. Dimension ludique :

- Score
- Classement
- Graphiques
- Export des résultats

OBJECTIF FINAL : Comprendre les politiques économiques et leurs limites.

```
<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="UTF-8">
<title>Simulation Macroéconomique SES</title>
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<style>
body { font-family: Arial; background:#eef2f7; text-align:center; }
.container { max-width:1200px; margin:auto; background:white; padding:20px; border-radius:10px; }
input, select, button { margin:5px; padding:8px; }
.base { background:#f0f4ff; padding:10px; border-radius:8px; }
</style>
</head>

<body>

<div class="container">

<h1>🎮 Simulation Macroéconomique</h1>

<h3>🎯 Objectifs</h3>
<p>Inflation cible : <b>2%</b></p>
<p>Déficit &lt; 3% | Dette &lt; 60%</p>

<input id="playerName" placeholder="Nom du groupe">

<p id="scenario"></p>
<div class="base" id="base"></div>

<h3>Politiques</h3>
<input id="rate" type="number" placeholder="Taux (%)>
<input id="spending" type="number" placeholder="Dépenses (% PIB)>
<input id="taxes" type="number" placeholder="Recettes (% PIB)>

<br>
<button onclick="simulate()">Simuler</button>
<button onclick="nextTurn()">Tour suivant</button>

<h3>Résultats</h3>
<div id="result"></div>
<div id="details"></div>
<div id="budget"></div>
<div id="score"></div>

<canvas id="chart"></canvas>

<h3>🏆 Classement</h3>
<ul id="leaderboard"></ul>
```

</div>

<script>

```
// =====  
// VARIABLES GLOBALES  
// =====
```

```
let debt = 60;  
let totalScore = 0;  
let scores = {};  
let turn = 1;
```

```
let growthData = [];  
let inflationData = [];  
let unemploymentData = [];
```

```
let chart;  
let current;
```

```
// =====  
// SCÉNARIOS  
// =====
```

```
const scenarios = [  
  {  
    text: "Récession (baisse de la demande)",  
    type: "recession",  
    base: { rate:3, spending:50, taxes:45 }  
  },  
  {  
    text: "Inflation (demande trop forte)",  
    type: "inflation",  
    base: { rate:2, spending:55, taxes:45 }  
  },  
  {  
    text: "Hausse des coûts (inflation + chômage)",  
    type: "supply",  
    base: { rate:3, spending:52, taxes:46 }  
  }  
];
```

```
// =====  
// INITIALISATION  
// =====
```

```
function newScenario(){  
  current = scenarios[Math.floor(Math.random()*scenarios.length)];  
  
  scenario.innerText = "Tour " + turn + " : " + current.text;  
  
  base.innerHTML =
```

📌 Situation initiale :

```
Taux = ${current.base.rate}% |  
Dépenses = ${current.base.spending}% |  
Recettes = ${current.base.taxes}%`;
```

```
rate.value = current.base.rate;  
spending.value = current.base.spending;  
taxes.value = current.base.taxes;
```

```
}
```

```
// =====
```

```
// SIMULATION
```

```
// =====
```

```
function simulate(){
```

```
let name = playerName.value || "Anonyme";  
let r = parseFloat(rate.value);  
let G = parseFloat(spending.value);  
let T = parseFloat(taxes.value);
```

```
if(isNaN(r) || isNaN(G) || isNaN(T)){  
  alert("Remplir tous les champs");  
  return;  
}
```

```
// ---- DEMANDE GLOBALE ----
```

```
let C = (T > 70) ? -3 : (T < 40 ? 2 : 1);  
let I = (r < 2) ? 3 : (r > 5 ? -2 : 1);  
let G_effect = (G > 65) ? 3 : (G < 45 ? -1 : 1);
```

```
let demand = C + I + G_effect;
```

```
// ---- RÉSULTATS ----
```

```
let growth, inflation, unemployment;
```

```
if(demand > 5){  
  growth=4; inflation=5; unemployment=4;  
} else if(demand > 2){  
  growth=3; inflation=2; unemployment=5;  
} else if(demand > 0){  
  growth=1; inflation=1; unemployment=7;  
} else {  
  growth=0; inflation=0; unemployment=10;  
}
```

```
// ---- DETTE ----
```

```
let deficit = G - T;  
debt += deficit * 0.5;
```

```
// ---- ANALYSE POLITIQUES ----
```

```
let monoExp = r < current.base.rate;
```

```

let monoRest = r > current.base.rate;
let budExp = G > current.base.spending || T < current.base.taxes;
let budRest = G < current.base.spending || T > current.base.taxes;

let policyText = `
Monétaire : ${monoExp ? "Expansionniste" : monoRest ? "Restrictive" : "Neutre"}<br>
Budgétaire : ${budExp ? "Expansionniste" : budRest ? "Restrictive" : "Neutre"}
`;

// ---- CORRECTION AVANCÉE ----
let correction = "";

if(current.type==="recession"){
  if(monoExp && budExp){
    correction="✅ Très bonne politique de relance";
  } else if(monoRest && budRest){
    correction="❌ Politique aggravant la récession";
  } else {
    correction="⚖️ Politique partielle";
  }
}

if(current.type==="inflation"){
  if(monoRest && budRest){
    correction="✅ Bonne politique anti-inflation";
  } else if(monoExp && budExp){
    correction="❌ Inflation aggravée";
  } else {
    correction="⚖️ Politique partielle";
  }
}

if(current.type==="supply"){
  correction="⚖️ Arbitrage inflation / chômage";
}

// ---- SCORE ----
let stability = Math.abs(inflation-2)+Math.abs(unemployment-5);
let score = Math.max(0,10-stability);

totalScore += score;
scores[name] = (scores[name] || 0)+score;

// ---- AFFICHAGE ----
result.innerHTML =
`Croissance: ${growth}% | Inflation: ${inflation}% | Chômage: ${unemployment}%`;

details.innerHTML =
policyText + "<br><b>Correction :</b> " + correction;

budget.innerHTML =
`Déficit: ${deficit}% | Dette: ${debt.toFixed(1)}%`;

```

```

scoreDiv.innerHTML = "Score : " + totalScore;

// ---- GRAPH ----
growthData.push(growth);
inflationData.push(inflation);
unemploymentData.push(unemployment);

updateChart();
updateLeaderboard();
}

// =====
// GRAPH
// =====

function updateChart(){
  if(chart) chart.destroy();

  chart = new Chart(chartCanvas, {
    type:'line',
    data:{
      labels: growthData.map((_,i)=>"T"+(i+1)),
      datasets:[
        {label:"Croissance", data:growthData},
        {label:"Inflation", data:inflationData},
        {label:"Chômage", data:unemploymentData}
      ]
    }
  });
}

// =====
// CLASSEMENT
// =====

function updateLeaderboard(){
  leaderboard.innerHTML = "";
  Object.entries(scores)
  .sort((a,b)=>b[1]-a[1])
  .forEach(([n,s])=>{
    let li=document.createElement("li");
    li.textContent = n + " : " + s;
    leaderboard.appendChild(li);
  });
}

// =====
// TOUR SUIVANT
// =====

function nextTurn(){

```

```
turn++;  
newScenario();  
}
```

```
// =====  
// LANCEMENT  
// =====
```

```
newScenario();
```

```
</script>
```

```
</body>
```

```
</html>
```