

Main Camera

RenderCamera_Player1 = Hitbox 1

Dans le template d'Hitbox
(= A00 Debug dans le repo Github UnityTemplate)

ImpactPointControl.cs
From Hitbox template.
Calculate impact position based on
Hitbox pressure data points

Send impact position through event
ImpactPointControlEventArgs()

ImpactManager.cs
Receive impact position and
send instructions based on
game progression

Instantiate

ImpactFeedback.cs

Clic de souris dans l'éditeur Unity
(pour simuler un sac quand on n'en a pas et qu'on développe)

MouseClickPosition

Send impact position
GetInteractPoint(Vector2)

SessionManager.cs

Instantiate / Start
(Send impact position)

Bilan du round
sur scenario 1

Scenario 1 Prefab

scenario1.cs

Instantiate / Start
(Send impact position)

Bilan du round
sur scenario 2

Scenario 2 Prefab

scenario2.cs

Instantiate / Start
(Send impact position)

Bilan du round
sur scenario 3

Scenario 3 Prefab

scenario3.cs

DeviceManager.cs
Receive data from different
devices

Movuino

Polar (cardio)

HTC Vive (position absolue)

Hitbox (impact)

Hitbox (acc + gyro)

UserInterface.cs
Interface utilisateur pour
renseigner le déroulement de la
session

Set session parameters

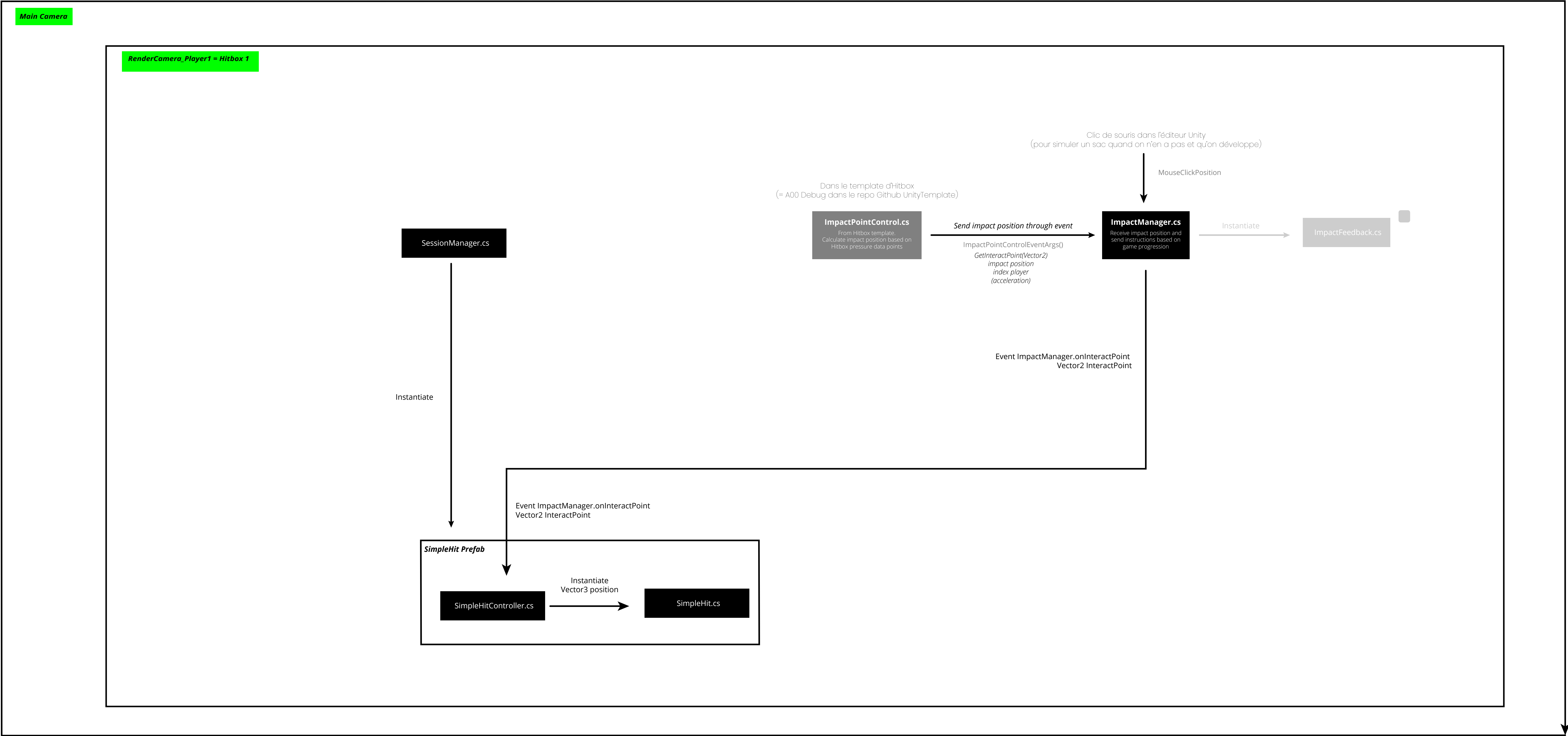
Bilan de la session
(données utiles utilisateur)

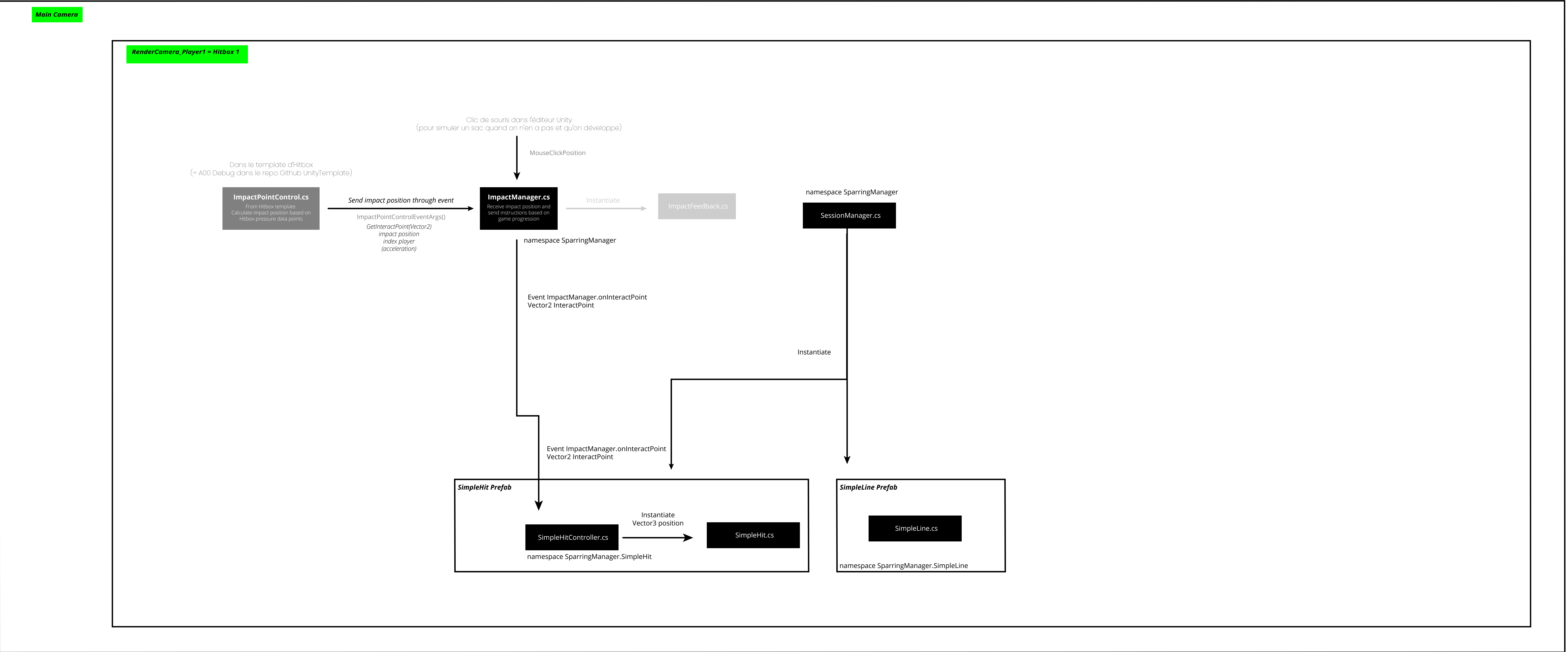
Command export

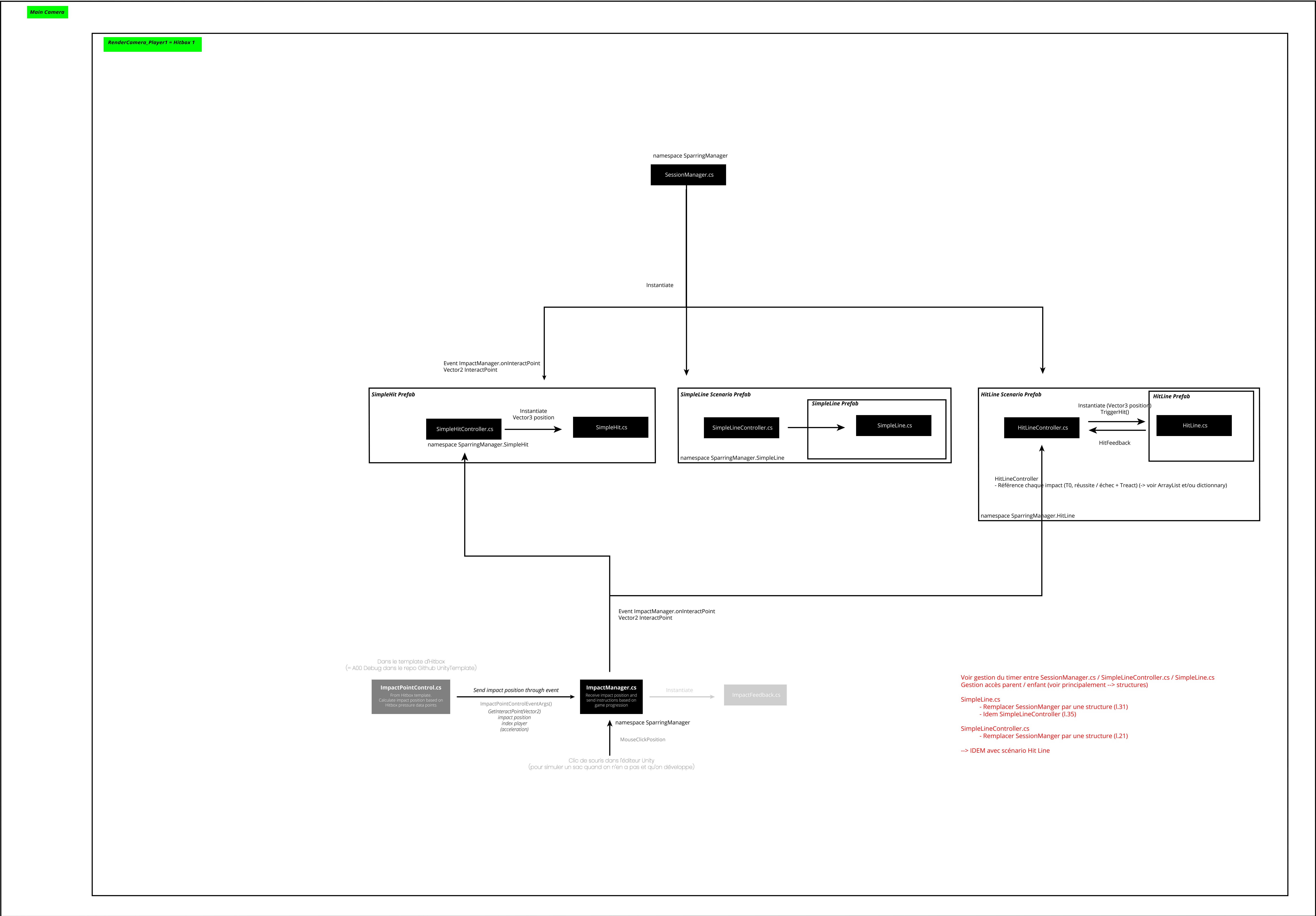
ExportManager.cs
Export les données des différentes
sessions dans un fichier type .csv

Données complètes
(courbes cardio, positionnement boxeur,
positionnement consigne...)

data.csv

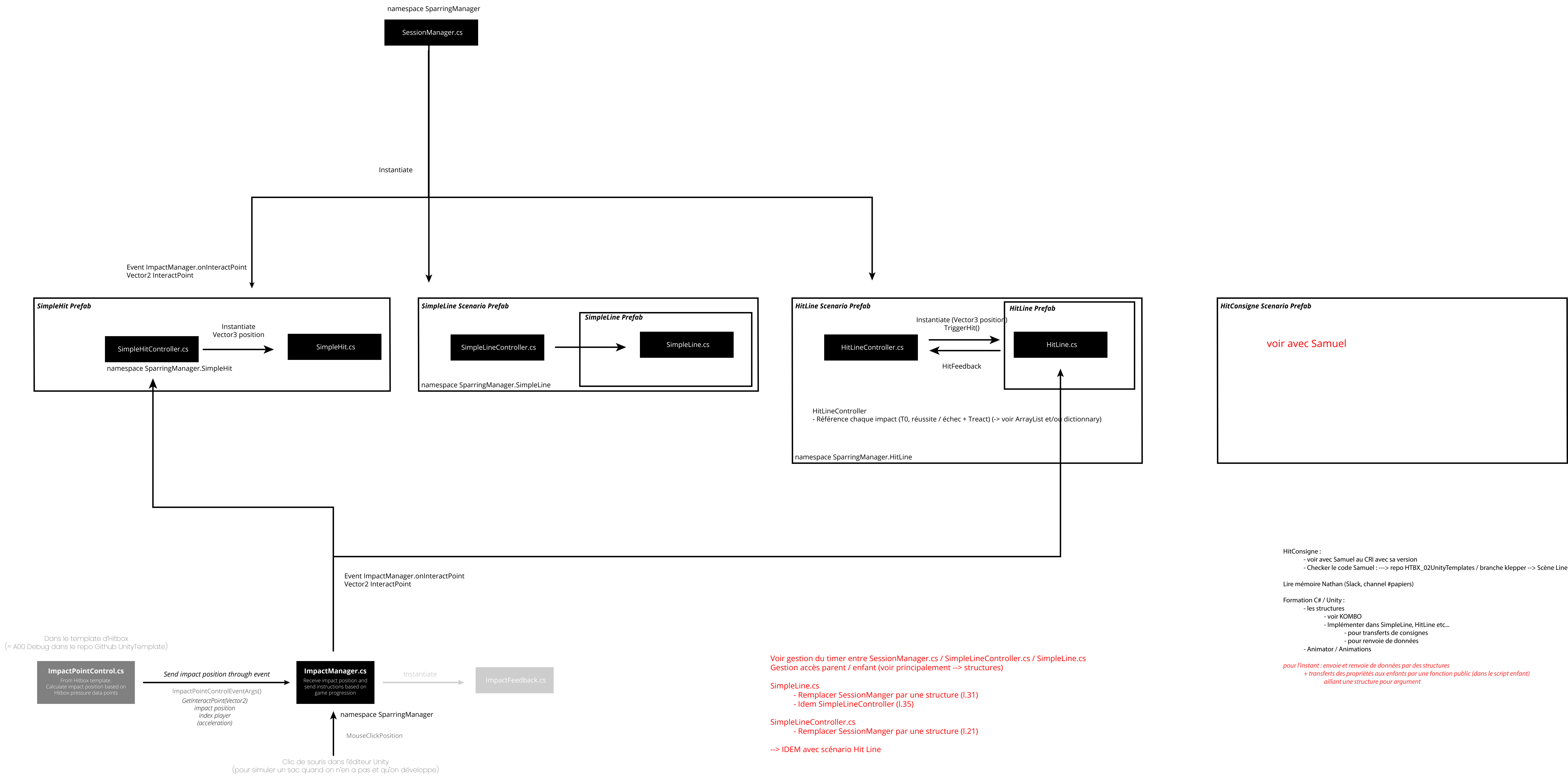






Main Camera

RenderCamera, Player1 = Hitbox 1



HitConsigne :
- voir avec Samuel au CRI avec sa version
- Checker le code Samuel : --> repo HTBX_02UnityTemplates / branche klepper --> Scène Line Polar voir prefab BagDisplay / MovingBar

Lire mémoire Nathan (Slack, channel #papiers)

Formation C# / Unity :
- les structures
- voir KOMBO
- Implémenter dans SimpleLine, HitLine etc...
- pour transferts de consignes
- pour renvoi de données
- Animator / Animations

*pour l'instant : envoi et renvoi de données par des structures
+ transferts des propriétés aux enfants par une fonction public (dans le script enfant)
alliant une structure pour argument*

Voir gestion du timer entre SessionManager.cs / SimpleLineController.cs / SimpleLine.cs
Gestion accès parent / enfant (voir principalement --> structures)

SimpleLine.cs
- Remplacer SessionManger par une structure (I.31)
- Idem SimpleLineController (I.35)

SimpleLineController.cs
- Remplacer SessionManger par une structure (I.21)

--> IDEM avec scénario Hit Line