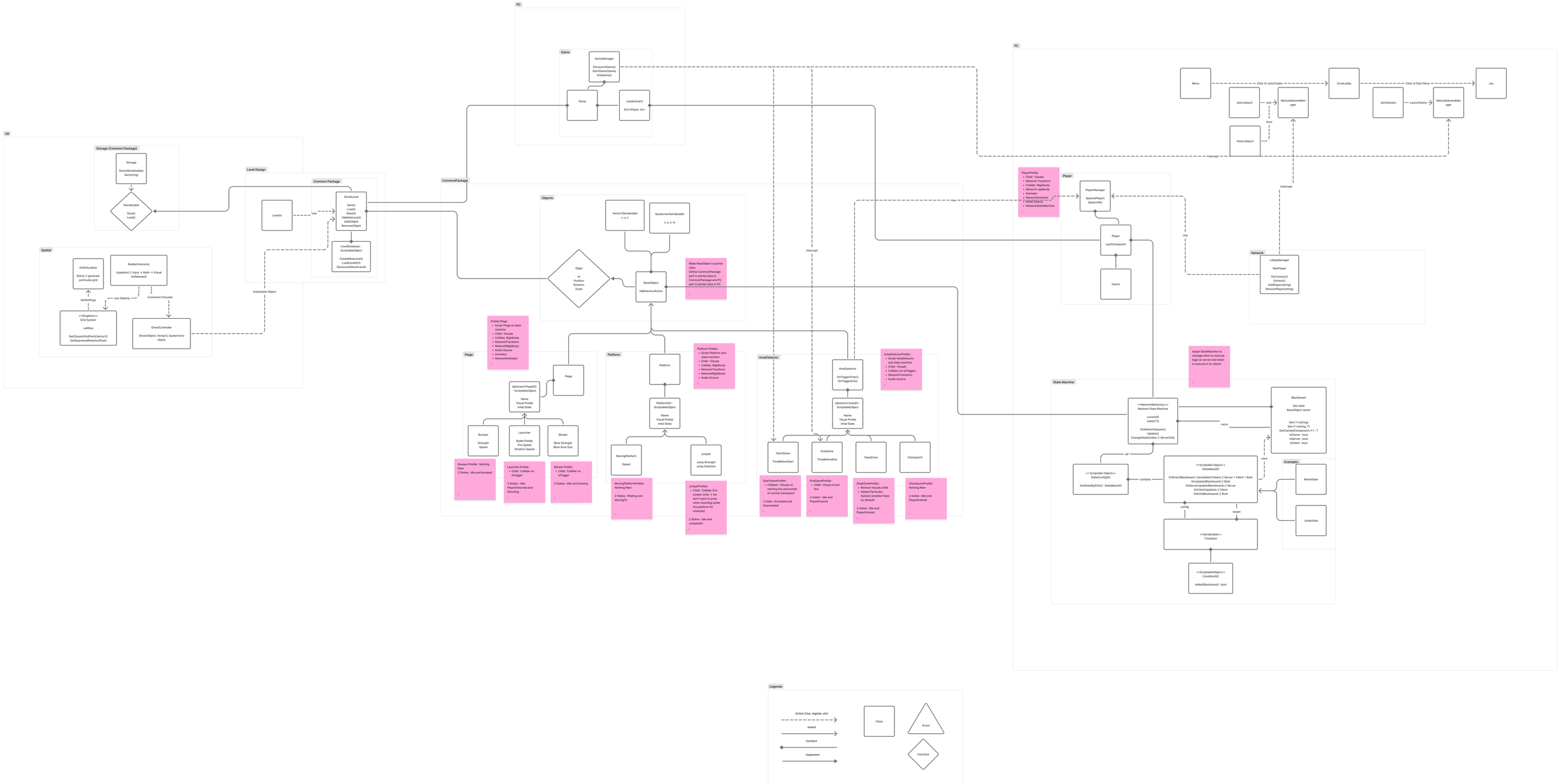
PC
Game
GameManager
OnLaunchGame()
StartGame(Game)
EndGame()
Game
Leaderboard
Dict<Player, int>
VR
Storage (Common Package)
Storage
Store(ISerializable)
GetString()
ISerializable
Save()
Load()
Level Design
Common Package
LevelUI
Use
SlotsLevel
Save()
Load()
Reset()
ValidateLevel()
Add(Object)
Remove(Object)
LevelDatabase : ScriptableObject
CreateNewLevel()
LoadLevel(int)
SaveLevel(SlotsLevel)
Instantiate Object
Spatial
GridVisualizer
Start() // generate particules grid
BuilderInteractor
Update() // Input -> Math -> Visual
OnRelease()
GetSettings
Use (Matrix)
Command (Visuals)
<<Singleton>>
Grid System
cellSize
GetClosestGridPoint(Vector3)
GetQuantizedRotation(float)
GhostControler
Show(Object, Vector3, Quaternion)
Hide()
CommonPackage
Objects
Vector3Serializable
x, y, z
QuaternionSerializable
x, y, z, w
Objet
Id
Position
Rotation
Scale
BaseObject
IsBehaviourActive
Make BaseObject a partial class
Define CommonPackage part in partial class in CommonPackage and PC part in partial class in PC
Intercept
Use
Piege
Prefab Piege
• Script Piege et state machine
• Child : Visuals
• Collider, Rigidbody
• NetworkTransform
• NetworkRigidbody
• Audio Source
• Animator
• NetworkAnimator
Piege
(abstract) PiegeSO : ScriptableObject
Name
Visual Prefab
Initial State
Bumper
Strength
Speed
Launcher
Bullet Prefab
Fire Speed
Rotation Speed
Blower
Blow Strenght
Blow Area Size
Bumper Prefab : Nothing New
2 States : idle and bumped
Launcher Prefab :
• Child : Collider on IsTrigger
3 States : Idle, PlayerDetected and Shooting
Blower Prefab :
• Child : Collider on IsTrigger
2 States : Idle and blowing
Platform
Platform
Platform Prefab :
• Script Platform and state machine
• Child : Visuals
• Collider, Rigidbody
• NetworkTransform
• NetworkRigidbody
• Audio Source
PlatformSO : ScriptableObject
Name
Visual Prefab
Initial State
MovingPlatform
Speed
Jumper
Jump Strenght
Jump Direction
MovingPlatformPrefab : Nothing New
2 States : Waiting and MovingTo
JumperPrefab :
• Child : Collider (For jumper zone + we don't want to jump when touching under the platform for example)
2 States : Idle and JumpedOn
AreaDetector
AreaDetector
OnTriggerEnter()
OnTriggerExit()
AreaDetectorPrefab :
• Script AreaDetector and state machine
• Child : Visuals
• Collider (on IsTrigger)
• NetworkTransform
• Audio Source
(abstract) AreaSO : ScriptableObject
Name
Visual Prefab
Initial State
StartGame
TimeBeforeStart
EndGame
TimeBeforeEnd
DeadZone
Checkpoint
StartGamePrefab :
• Children : Visuals of starting line and prefabs of normal checkpoint
2 state : Activated and Deactivated
EndGamePrefab :
• Child : Visual of end line
2 states : Idle and PlayerEntered
DeadZonePrefab :
• Remove Visuals child
• Added Particules System (enabled false by default)
2 states : Idle and PlayerEntered
CheckpointPrefab : Nothing New
3 states : Idle and PlayerEntered
PC
Menu
Click UI Join/Create
ZoneLobby
Click UI Start Race
Jeu
JoinLobbyUI
Join
NetcodeSceneManager
JoinGameUI
LaunchGame
NetcodeSceneManager
Host
HostLobbyUI
Intercept
Intercept
PlayerPrefab
• Child : Visuals
• Network Transform
• Collider, Rigidbody
• Network rigidbody
• Animator
• NetworkAnimator
• Added Scripts
• NetworkStateMachine
Player
PlayerManager
Spawn(Player)
SpawnAll()
Player
LastCheckpoint
Inputs
Use
Network
LobbyManager
MaxPlayer
OnConnect()
OnHost()
AddPlayer(string)
RemovePlayer(string)
Adapt StateMachine to manage when to execute logic on server and when to execute it on clients
State Machine
<<NetworkBehaviour>>
Network State Machine
currentID
stateCTS
OnNetworkSpawn()
Update()
ChangeState(index) // ServerOnly
owns
Blackboard
Dict data
BaseObject owner
GetT<T>(string)
SetvT<T>(string, T)
GetCachedComponent<T> : T
IsOwner : bool
IsServer : bool
IsClient : bool
uses
uses
contains
<< Scriptable Object>>
StateConfigSO
GetStateBy(Dict) : StateBaseSO
<<Scriptable Object>>
StateBaseSO
OnEnter(Blackboard, CancellationToken) // Server + Client + Both
OnUpdate(Blackboard) // Both
OnServerUpdate(Blackboard) // Server
OnClientUpdate() // Client
OnExit(Blackboard) // Both
config
target
<<Serializable>>
Transition
<<ScriptableObject>>
ConditionSO
IsMet(Blackboard) : bool
Examples
MoveState
JumpState
Légende
Action (Use, register, etc)
Inherit
Contient
Implement
Class
Enum
Interface