

DOSSIER DE PRODUCTION

401, A SPACE ODYSSEE

École de Terrain : Planétarium de
Marseille



FAYET Mateo

mateo.fayet@etu.univ-amu.fr

PITTILLONI – MAESTRACCI Maude

maude.pittilloni--maestracci@etu.univ-amu.fr

TOUZET Arthus

arthus.touzet@etu.univ-amu.fr

WALLERICH Mérové

merove.wallerich@etu.univ-amu.fr

Septembre 2022

401, A Space Odyssey : le Projet

401, A Space Odyssey est un jeu de memory sonore et visuel multijoueur en coopération. Pour communiquer avec les aliens, les quatre joueurs devront recomposer la musique que leur entonne le vaisseau qui surplombe la station spatiale en suivant les instructions de leur ordinateur de bord. En immersion dans l'espace, sous le dôme du planétarium de Marseille, les joueurs entendront et verront s'afficher devant eux des séquences sonores et visuelles qu'ils devront reproduire en utilisant les buzzers à leur disposition. Ils devront se souvenir de leur ordre d'intervention et reproduire la séquence en rythme avec le fond sonore. Le niveau de difficulté croissant permettra aux joueurs de composer petit à petit une bande son complexe pour mener à bien leur mission.

L'équipe

Notre équipe est composée de quatre membres : deux sont issus du parcours ICS (Ingénierie et Conception Sonore) et deux autres du parcours MC¹ (Musicologie et Création). Puisque nous avons tous suivi les mêmes cours, nous avons choisi de nous répartir les tâches en fonction des affinités de chacun avec les différents domaines à exploiter durant ce projet. Ainsi, Maude (MC), cheffe de projet, s'occupera de la partie esthétique musicale, qui comprend les compositions sur Ableton. Mérové (ICS) sera en charge de la composition, des effets audios ainsi que de l'export des informations MIDI pour faciliter le protocole OSC. Matéo sera chargé du protocole OSC sur Max for Live et en C# en lien avec Arthus qui développera le projet Unity et l'intégration des *Buzzers*.



¹ Master Acoustique et Musicologie AMU

Note d'intention

Les premières expérimentations sur le système de projection *fulldome* et l'octophonie du planétarium nous ont aiguillés sur les constats suivants : 1 - Il est difficile de se concentrer sur un objet qui n'est pas diffusé en face du spectateur. 2 - Les objets proches aux géométries régulières sont déformés par le type de projection et l'illusion peut en être atténuée. 3 - La forme de dôme rend la spatialisation précise des sons complexes.

Dans le but de maximiser l'expérience utilisateur dans le cadre d'un projet de création d'un prototype interactif fonctionnel, nous avons donc défini des contraintes de base dans la configuration du projet. Le jeu comportant 4 joueurs sera donc projeté sous la forme de quatre quarts disposés en face du joueur concerné. Toutes les informations nécessaires à la jouabilité du jeu seront diffusées en face des joueurs. Le fond dans lequel se déroulera l'action sera un espace ouvert, pour éviter le manque de réalisme des représentations d'intérieurs à cause de l'effet "*fish eye*" inversé. La spatialisation du son devra être maîtrisée avec précaution et des surprises sont attendues.

Gameplay

Le jeu mis en place pour le projet prend la forme d'un *memory* sonore. L'objectif étant d'entendre une séquence qui devra être reproduite par les joueurs dans le bon ordre et en rythme. Ce jeu pourrait être uniquement sonore, mais la projection apportera un support visuel à l'expérience utilisateur et permettra de créer un *story-telling* immersif qui rendra l'expérience plus totale. Basé sur les séquences et les problématiques célèbre du film *Rencontre du Troisième Type* de Steven Spielberg, (<https://www.youtube.com/watch?v=wZj7gUIO-2k>), le *memory* demandera à chaque joueur de déclencher des sons qui lui sont propres et de compléter le thème qui lui est proposé au préalable en suivant des indications de couleur. Lorsque le thème est présenté le son est accompagné des couleurs correspondantes pour chaque joueur en face de lui et le son sera spatialisé afin que les notes imposées au joueur soient diffusées en face de lui. On peut imaginer plusieurs niveaux de difficultés dans la complexité croissante des thèmes à reproduire mais aussi dans le nombre de boutons impliqués dans la restitution pour chaque joueur.

Esthétique Visuelle et Sonore

L'esthétique visuelle sera basée sur une esthétique de science-fiction de type *space odyssey*. On peut imaginer un visuel qui simule une soucoupe volante dans laquelle les joueurs seraient placés. Au centre, en haut, une entité (étoile, planète ou OVNI) servira à guider les joueurs en prenant la place de maître du jeu et/ou d'objectif à atteindre. On peut citer la série de jeu d'énigmes *Portal* de la société Valve comme inspiration du rendu visuel 3D. *Portal* et *Portal 2* empruntent déjà de nombreuses références au film *2001, L'Odyssée de l'Espace* de Kubrick qui constitue une inspiration assumée en terme d'esthétique visuelle et sonore.

L'esthétique musicale se voudra éthérée au départ pour se complexifier et se structurer au fur et à mesure des niveaux. Chaque niveau crée une nouvelle boucle qui constitue une voie de la composition totale. Le joueur aura l'impression de participer à la création de cette œuvre composée à l'avance. Concrètement des clips midi s'additionnent jusqu'au résultat final et la victoire de l'équipe. Les références seront piochées dans des œuvres de sciences fictions, *2001 l'Odyssée de l'Espace*, *Ghost In The Shell*...

Composition

Le but du jeu est de créer en coopération la musique au fur et à mesure de l'avancement du jeu. Les séquences jouées sont ensuite intégrées dans une boucle qui participe à l'élaboration finale de la musique du jeu. Afin de dynamiser le *game-play*, nous avons décidé d'une forme de musique évolutive. Afin d'augmenter la difficulté nous avons ajouté différentes variations de rythmes, sans qu'elles ne soient trop complexes pour que le jeu reste accessible à des joueurs non-musiciens.

Choix esthétiques

Le jeu se déroulant dans l'espace, nous avons décidé de créer une nappe sonore latente sans mélodie reconnaissable. Une rythmique est présente dès le début pour permettre à la fois d'aider les joueurs à être en rythme et d'autre part pour donner davantage de dynamique au jeu. Nous avons ensuite choisi des *plug-ins* qui proposent des sons résonnant avec effets semblables à ceux de « lasers » (souvent entendus dans les films de science fiction).

De plus, nous nous sommes inspirés de différents films pour imaginer l'ambiance et le jeu sonore, par exemple : “Rencontre du troisième type” de Steven Spielberg (1977) qui met en

scène une rencontre entre un OVNI et les humains. Afin de communiquer, un jeu musical sous forme de *memory* va se dérouler et permettre la communication par le biais de notes et des rythmes.

Mise en place des niveaux

Niveau 1

Pour ce niveau nous avons choisi de demander aux joueurs de créer 4 accords qui participent au commencement de la boucle. Ils sont joués avec un rythme linéaire de blanches pour chaque note. Chacune de ces dernières créées au fur et à mesure l'harmonie désirée. Les accords sont donc les suivants :

1. Do#/mi/sol/sib
2. Ré/mi/fa/la
3. Si/Ré/Mi/Sol
4. La/do#/mi/la

Répartition : Afin d'avoir un constant équilibre du *game-play* pour chaque joueur nous avons choisi une répartition s'équilibrant dans chaque niveau de manière à ne pas affecter l'expérience de jeu.

- *Player 1* : La (2) /Do# (2) = (4)
- *Player 2* : Sib (1) /Ré (2) = (3)
- *Player 3* : Mi (4) /Sol (2) = (6)
- *Player 4* : Fa (+La3) (1) /Si (1) = (2)

Déroulement du niveau : Parceque chaque joueur va jouer à tour de rôle une seule note de chaque accords, nous avons défini cette répartition par souci d'harmonie (*P* signifiant *Player*) :

- *Level 1.1* = P3/P2/P4/P1
- *Level 1.2* = P3/P3/P2/P1
- *Level 1.3* = P2/P4/P3/P3
- *Level 1.4* = P1/P1/P3/P1

Contraintes : En raison de l'intégration de la musique vers le jeu ainsi que du *game-play* qui propose deux cubes par joueurs (donc huit au total) nous avons restreint nos possibilités à 8 notes. Cependant le *Player 4* jouera deux notes en activant qu'une seule touche pour nous

permettre de pouvoir créer tous les accords, puisque nous avons en réalité neuf notes à causes du La3 qui octavie l'accord final. Le fa joué dans le niveau 1.3 sera donc joué avec le La3 puisque, se situant à la tierce, cela ne causera aucun dégat harmonique.

Niveau 2

Après avoir créé les accords, le niveau 2 permettra aux joueurs de concevoir la ligne mélodiques principale de la musique. De fait, nous avons décidé de jouer 4 mesures qui donneront lieu à 4 sous-niveaux. Chaque mesure comprend 4 notes :

- Mesure 1 : Ré/Ré/Fa/Ré
- Mesure 2 : Mi/La/Mi/Fa
- Mesure 3 : La/Mi/Ré/Ré
- Mesure 4 : Do#/Mi/Sol/La

De plus, après une rythmique continue au premier niveau, nous avons décidé de complexifier le jeu en ajoutant une rythmique qui sera identique à chaque mesure : deux noirs sont jouées sur le premier temps, une croche suivie d'une noire qui donne un rythme syncopé sur le deuxième et les deux temps restant sont à vide.

Répartition : Pour les mêmes raisons qu'évoqué au niveau précédent nous avons décidé d'une répartition la plus équilibrée possible. Pour ce faire nous avons compté le nombre de notes que jouent chacun des participants :

- *Player 1* : Fa = (2)
- *Player 2* : Mi (4) /Sol (1) = (5)
- *Player 3* : La = (3)
- *Player 4* : Ré = (5)

Ici nous avons seulement six notes différentes qui facilitent l'intégration dans *Unity*.

Déroulement du niveau : Nous avons déterminé dans quel ordre les joueurs doivent actionner les *buzzers* comme pour le niveau précédent :

- *Level 2.1* = P4/P4/P1/P4
- *Level 2.2* = P2/P3/P2/P1
- *Level 2.3* = P3/P2/P4/P4
- *Level 2.4* = P1/P2/P2/P3

Niveau 3

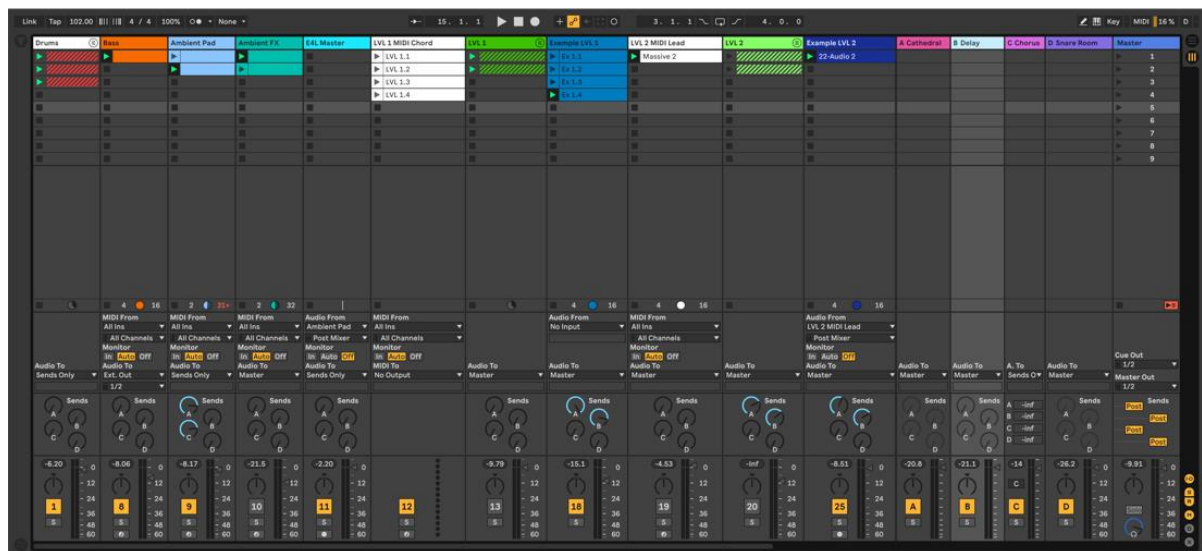
Pour le 3ème et dernier niveau, nous avons imaginé une augmentation de tempo avec d'avantage d'éléments rythmiques. Comme décrit précédemment, ce niveau comporte une répartition équitable du temps de *game-play*. Ayant auparavant équilibré le temps d'interaction de chacun des joueurs, nous complexifieront les interactions avec le *buzzer* en alternant la rapidité rythmique. De plus, il est intéressant de surprendre les joueurs en les faisant jouer tous en même temps le même rythme pour terminer le niveau. Cela nécessite donc plus de mouvements de la part des joueurs avec une complexité plus importantes correspondant à celle d'un niveau final.

Fin du jeu

Une fois le jeu achevé la musique est enfin complète. Nous allons *looper* l'intégralité de la musique et finir sur un accord final en partie en guise de réussite du jeu.

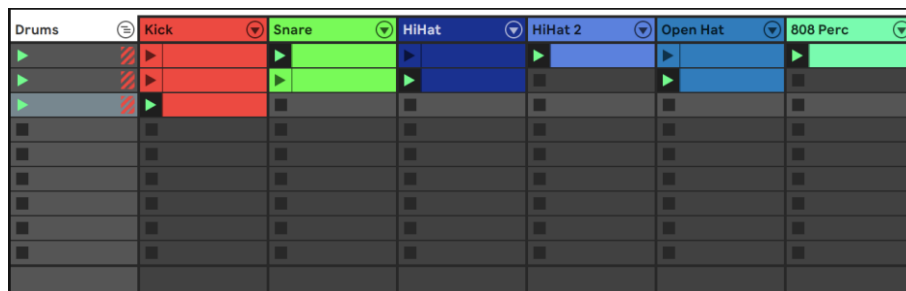
Ableton

La vue d'ensemble du projet :



L'ambiance initial se compose de :

-Drums



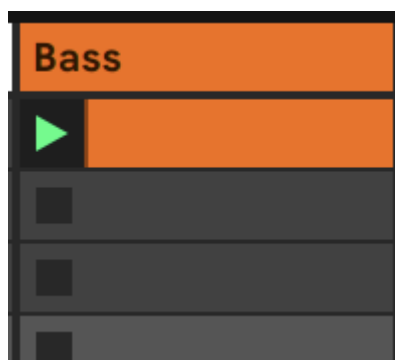
Il s'agit d'une rythmique breakbeat constitué de différents samples issus d'ableton.

-Pads/Fx



Ils ont été réalisés à l'aide du vst massive.

-Bass



Elle a également été créé avec massive. Il s'agit du sub bass contenant peu d'harmonique. Elle sert à ajouter de la profondeur.

Nous avons également utilisé différents effets en piste de retour (en insert, afin de mixer et donner une place à chaque élément.

A Reverb Cathedral	B Delay	C Chorus	D Reverb Room

Nous avons aussi utilisé des effets en insert comme de la compression ou de l'égalisation.



Ici, par exemple, nous avons placé un glue compressor sur notre groupe de drums

La diffusion se faisant sur un système 8.1, nous avons modifié le Source panner de Enveloppe for Live afin de spatialiser les éléments dans le dôme.



Il s'agit de la chaine d'effet de la piste de pad d'ambiance. Nous avons souhaité obtenir un effet de rotation dans le dôme, d'où l'utilisation d'un LFO mappé sur l'azimut.

Ensuite pour spatialiser chacun des éléments, notamment les notes reçues par chacun des joueurs suivant leur orientation, nous avons placé un *Source Panner* dans la direction souhaitée. Finalement, il suffisait de créer une piste audio avec un master bus et de router les pistes à spatialiser en *send only*.



Ici, il est réglé sur octo pour rediriger chaque voix vers chacune des 8 enceintes.

Chaque niveau s'articule selon autour de 3 sections :

- La section MIDI

LVL 1 MIDI Chord	
▶	LVL 1.1
▶	LVL 1.2
▶	LVL 1.3
▶	LVL 1.4

Elle contient les informations MIDI qui ont servi à enregistrer l'audio, et à être envoyé dans *Unity* pour les comparer avec celle du joueur. Ici, la piste contient 4 clips MIDI contenant chacune une voix de l'accord du niveau 1.

- La section audio ex

Ex LVL 1	
▶	Ex 1.1
▶	Ex 1.2
▶	Ex 1.3
▶	Ex 1.4

Elle contient les clips audios qui sont bouclés à chaque validation de niveau/sous-niveau. Ici les 4 clips audios, contiennent la construction de l'accord (1 voix à 4 voix)

-La section audio player

LVL 1	Player 1	Player 2	Player 3	Player 4
▶	▶ LA	▶ RE	▶ MI	▶ SI
▶	▶ DO#	▶ Sib	▶ SOL	▶ FA (LA3)
■	■	■	■	■
■	■	■	■	■
■	■	■	■	■
■	■	■	■	■
■	■	■	■	■
■	■	■	■	■

Elle contient un groupe de 4 pistes correspondant aux notes que joueront les 4 joueurs. Ce sont ces clips que nous allons activer à l'aide de *Max4live* lorsqu'un joueur appuie sur un bouton.

Protocole technique :

Logiciels et Organisation fonctionnelle

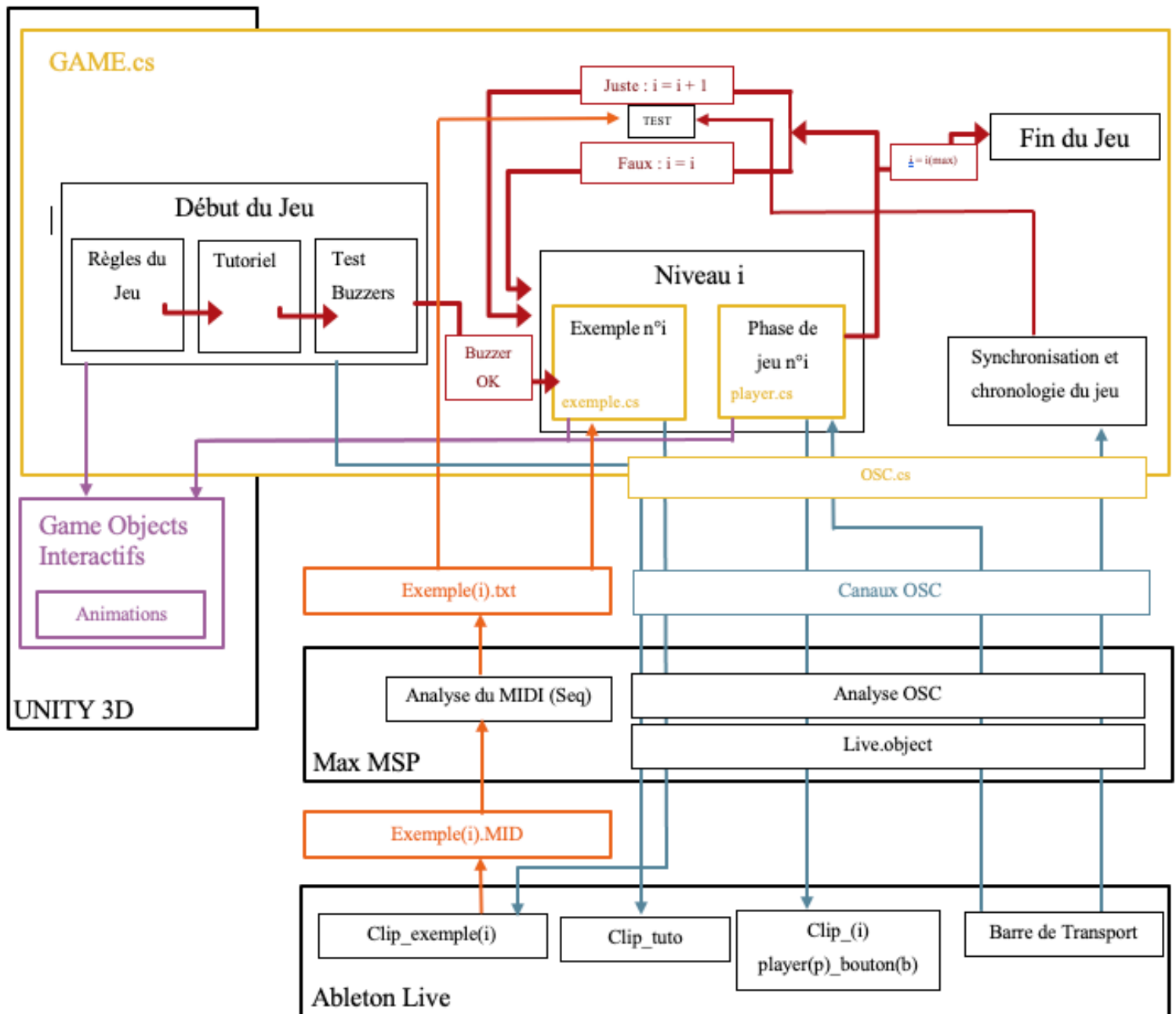
Le diagramme ci-dessous décrit l'architecture du projet. Les trois logiciels – Unity, Max for Live et Ableton Live – contiennent chacun des objets et des fonctions propres nécessaires au déroulement de la phase de jeu complète. Les différents scripts C# sont indiqués dans des boîtes jaunes titrées .cs.

GAME.cs constitue le script maître d'une session de jeu. Il intègre les différentes phases de jeu (Début, niveaux, fin) et contient des méthodes qui lui permettent de tester la validité d'une condition (en rouge) à partir des interactions avec les utilisateurs pour maîtriser le déroulement du jeu. Pour se faire il se base sur deux canaux d'information : la barre de transport qu'il reçoit en OSC via OSC.cs depuis Max MSP, et des assets qui contiennent l'ensemble des données de synchronisation midi de chaque niveau et de chaque joueur (exemple(i).txt en orange). Game.cs compare le tableau de valeurs qu'il reçoit avec le type et le minutage des événements déclenchés par chaque joueur indiquées par les scripts player(p).cs.

Player(p).txt permet de récupérer les interactions de chaque joueur avec sa manette, d'envoyer en OSC les clips à déclencher pour chacune de ses interactions. En parallèle il contrôle le feedback visuel des actions du joueur.

Exemple.cs lit exemple(i).txt pour permettre le feedback visuel de la séquence exemple et envoie à Max l'ordre de jouer le son correspondant.

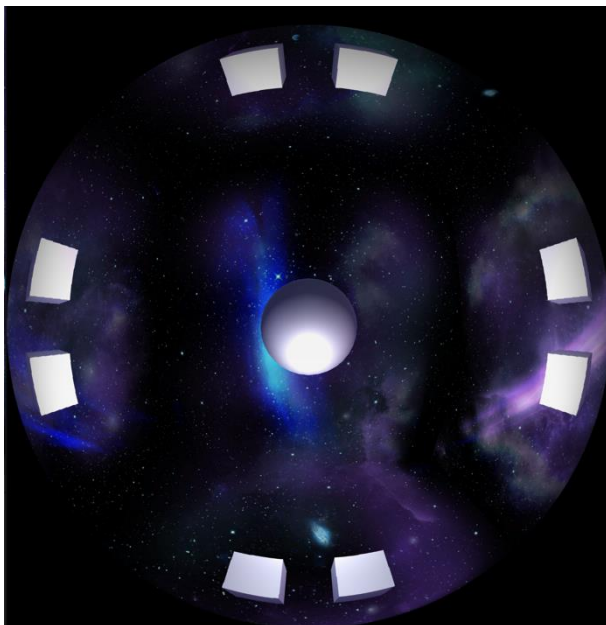
Diagramme de l'architecture fonctionnelle



Design 3D dans Unity

Le projet Unity est pensé pour simuler une immersion dans l'espace. Le dôme est divisé en quatre quarts : un pour chaque joueur. Sur chaque quart, les joueurs visualisent deux objets (les cubes dans le prototype) qui sont les feedbacks visuels correspondant aux boutons de leur manette respective. Un dossier *material* contient les textures nécessaires au retour visuel. Ces cubes sont animés de telle manière à simuler un effet de flottement.

La caméra *Dôme* est située au centre de la station spatiale. Le package nécessaire à son bon fonctionnement a été fourni par l'équipe pédagogique lors de la phase de prototypage. Il est stocké dans un dossier *DomeProjection* dans les assets du projet. À son zénith, le dôme est



surplombé par une entité alien (la sphère dans le prototype) qui constitue l'objet de la mission. L'horizon est constitué à partir d'une skybox (*DeepSpace Skybox Pack*) disponible gratuitement sur le Unity Assets Store. Toute la station spatiale (camera comprise) et contenue dans un objet parent en rotation lente et constante sur y et z. L'horizon profond semble donc en mouvement constant. Toutes les animations et les animationControllers sont stockées dans un dossier *animations* dédié dans les assets du projet.

Les tâches à accomplir dans unity sont les suivantes :

- Design et animation de l'entité alien (inspiration *The Arrivals* de Denis Villeneuve).
- Création et animation d'un objet permettant la personnification de l'ordinateur de bord (inspiration GLaDOS dans *Portal*, ordinateur de bord de *2001, A Space Odyssey*).
- Design des feedback manette (inspiration hologrammes de *Cowboy Bebop*, 1998).

Max for live

Max for Live pourra ainsi jouer différents rôles dans le protocole OSC. Premièrement, l'analyse des informations envoyées par Ableton telles que la métrique, le tempo et la localisation au sein de la composition seront envoyées à Unity par l'intermédiaire de l'outil

transport. Ce dernier analyse les informations de l'horloge interne d'Ableton et permet de créer des messages contenant les informations en temps réel à destination de Unity.

L'outil *seq* nous permet d'éditer des listes de valeurs contenant les informations MIDI des clips Ableton. Ce dernier sera généré au démarrage du programme, et les adresses MIDI fournies à Unity permettront de déclencher les clips audios en fonction de la validité des tests vérifiés par GAME.cs.

Enfin l'outil *live.object* réalise des opérations dans Live avec l'aide de *live.path* qui permet de naviguer au sein des objets Live. En fonction de l'état du jeu et des informations reçues par Unity, cet objet pourra donc déclencher des clips dans Live.

Github

Puisque nous serons amenés à travailler en parallèle autour de l'arborescence des scripts C# et de Unity, nous avons paramétré un Github qui contient l'intégralité du projet Unity (et donc des scripts) et le Max for Live en lien avec le protocole OSC. Ceci nous permet premièrement de mettre à jour les changements réguliers au sein du projet, mais aussi de sécuriser les sauvegardes, et de pouvoir revenir en arrière si nous commettons des erreurs. Nos dossiers sont donc clonés sur nos ordinateurs via Github Desktop et la manipulation des *commits* peut se faire en ligne de commande *via* le Git². Ce paramétrage permet ainsi de mettre à jour le projet Unity sans avoir à se communiquer tous les changements à la main. Ce dernier constate des changements dans son arborescence et propose ainsi de recompiler les scripts et les changements s'effectuent automatiquement : nous gagnons du temps et évitons les erreurs. Enfin, puisque le projet Github est publique, nous pouvons facilement le partager et le mettre à disposition de personnes qui en auraient besoin !

Première Expérimentation

Nous avons eu la possibilité d'expérimenter jeudi 8/09 les diffusions audio et visuelles dans le planétarium. Nous avons donc préparé un prototype Unity afin d'observer si la disposition des différents modules de jeu était claire au moment de l'expérience utilisateur. De plus, nous doutions que le câble de chaque *Buzzer* soit suffisamment long pour que les

² Git Bash nécessaire pour windows.

utilisateurs se disposent aux quatre coins du planétarium. Nous avons donc noté que nous allions devoir disposer des chaises plus proches du centres que les fauteuils installés. En effet, un joueur se tenant debout serait ébloui par la diffusion vidéo et cacherait une partie de l’image à ses coéquipiers.

Nous avons ensuite testé le routage d’Ableton en octophonie. Le but était donc de faire fonctionner les patches de spatialisation E4L modifiés par nos soins pour le planétarium. Ainsi, nous avons pu expérimenter la spatialisation contrôlée par LFO.

Gestion de projet

Planning de production

Tableau 1: Planning de Production

Tache	Qui	Check	Début	Fin	12/9	13/9	14/9	15/9	16/9
Composition	Maude		12/9	12/9					
Composition	Mérové		12/9	12/9					
Enregistrement voix	Maude		13/9	13/9					
Enregistrement voix	Mérové		13/9	13/9					
Mixage & Spat	Mérové		14/9	14/9					
Export info MIDI	Mérové		15/9	15/9					
Programmation Gameplay	Arthus		12/9	13/9					
Fin Design Unity	Arthus		14/9	15/9					
OSC	Matéo		12/9	13/9					
Max for Live	Matéo		14/9	15/9					
Rédaction Documentation	Maude		14/9	15/9					
Test & Restitution	Tous		16/9	16/9					

Tableau : Planning de Production

Ce tableau partagé en interne inspiré du diagramme de Gantt, permet à chacun de repérer les échéances correspondant à chaque tâche et de connaître l'évolution des tâches réalisées par nos camarades.

Evaluation préalable des compétences du groupe

Nous avons, à première vue, une équipe composée d'éléments motivés et intéressés dans les tâches à réaliser tout en étant polyvalents. En effet, notre organisation interne lors de la répartition des tâches a permis à chacun de prendre en charge uniquement des éléments du projet lorsqu'ils étaient souhaités. La polyvalence de chacun permet ainsi d'être force de proposition et de venir en aide aux autres camarades lorsqu'elle est nécessaire.

Nous avons tous conscience que notre motivation nous pousse à vouloir réaliser un projet plus ambitieux que ce que nous permettent les cours délais. Cet avantage doit donc être régulé par un planning précis mettant en avant les priorités du projet : un prototype fonctionnel qui répond au cahier des charges.