**CONCERTO**

**Plan de test**

**Réalisé par le Groupe W**

Marc-André Bouchard

Émile Désilets

Ghislain Lacombe

Vincent Lalancette

Jeremy Levesque

**Responsable des tests**

Jeremy Levesque

**Jeudi 17 décembre 2020**

# **INTRODUCTION**

Notre projet se résume à un concert interactif mêlant le réel avec le virtuel. Un stream en direct présentera aux spectateurs un décor composé d’une maquette réduite et d’un jeu de lumière. À partir d’un site web, l’utilisateur contrôlera l’audio de la musique à l’aide d’un contrôleur midi virtuel. L’éclairage s'adapte aux hautes et basses fréquences de la musique, eux-mêmes modifiées selon les commandes entrées par l’utilisateur.

Notre produit minimum viable serait de pouvoir mettre tous nos éléments sur un site web local.

Pour le prototype 3, nous allons devoir tester le passage du logiciel OBS vers OSC, la projection de nos clips vidéo dans le studio, l’interaction des lumières et intégrer la totalité de notre projet sur un site web en local.

**Responsable des tests**

Jeremy Levesque, jeremy0levesque@gmail.com

**Équipe de testeurs**

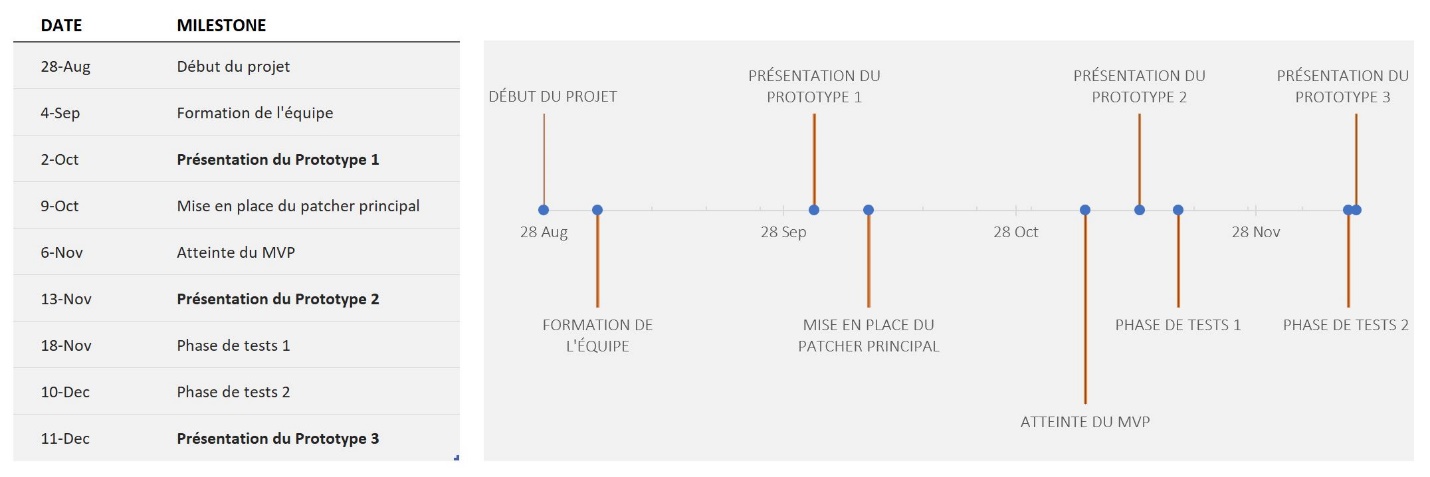
* Marc-André Bouchard
* Émile Désilets
* Ghislain Lacombe
* Vincent Lalancette
* Jeremy Levesque

**GitHub:** https://github.com/Mrm0u/Concerto

**Date de livraison PHASE TEST 1:** 27 novembre 2020

**Date de livraison PHASE TEST 2:** 11 décembre 2020

Notre première phase de test a servi à tester sous forme physique notre projet, en branchant tous nos éléments ensemble, en plus de nous familiariser avec le matériel. Notre deuxième phase de test servira à régler les problématiques rencontrées à la phase de test 1, nous assurant d’avoir la base pour notre MVP.

**Project timeline:**

**ENVIRONNEMENT DE TEST**

L'installation se fait dans un grand studio situé au Collège Montmorency. La maquette sera située dans le milieu du cyclo. La projection des clips vidéos se fera sur le mur derrière la maquette, prenant une largeur de quatre mètres sur 2.56 m de hauteur. Quatre caméras filmeront la scène sous quatre angles différents (plan large, rapproché, hauteur du sujet et vue du haut de la maquette). Deux piliers seront fixés au plafond sur les extrémités de la zone de projection. Sur chacun d’eux, trois spots y seront attachés et projetteront sur les murs leur lumière. Un autre spot projettera sur la maquette. Tous les spots seront branchés en série. Le projecteur, les caméras et les spots seront tous connectés sur le même ordinateur utilisant Windows. Sur l’ordinateur, le patcher Max8, OBS et QLC+ seront ouverts.

**Contexte de réalisation:**

Nous allons présenter le prototype 3 sous forme d’une simulation du projet final. Sur place, nous allons présenter notre installation physique installée dans le grand studio au Collège Montmorency. La présentation se fera le 11 décembre 2020.

**Contexte de tests:**

Nous allons réaliser en grande partie nos tests avec notre matériel physique dans le grand studio, pour nous habituer à la présentation finale et nous familiariser avec le matériel. De plus, travailler ensemble en physique nous permet de mieux visualiser notre travail, nous permet de mieux communiquer nos idées et nous assure de toujours rester à jour avec le déroulement du projet.

**Logiciels utilisés dans la chaîne de réalisation:**

* Max 8
* OpenStage Control
* OBS
* QLC+
* ffmpeg
* VB-CABLE

**Logiciels de déploiement:**

* OpenStage Control
* Teams

**Bibliothèque:**

* Youtube

**Plateformes ciblées:**

* Firefox
* Google Chrome

**MATRICE DE RISQUES**

**Risques techniques:**

* Compatibilité entre les logiciels utilisés
* Compatibilité entre les équipements utilisés
* Compatibilité entre les équipements et les logiciels utilisés
* Utilisation de technologies non maîtrisées
* Bris d’équipement

**Risques humains:**

* Blessure liée à l’installation d’équipement
* Incompétence des membres à accomplir leurs tâches attitrés
* Contraction du Covid-19 par un membre de l’équipe

**Risques juridiques:**

* Perte de l’autorisation à l’accès du studio
* Perte de l’autorisation à l’accès à l'équipement
* Interdiction d’utilisation des médias sélectionnés

**Risques délais:**

* Manque de communication entre les membres de l’équipe
* Manque d'intérêt envers le projet
* Perte intégrale des fichiers et des sauvegardes

Les risques de compatibilité sont les plus essentiels à régler, car dès qu’une seule compatibilité logiciel-logiciel, logiciel-équipement ou équipement-équipement tombe, le projet au complet s’écroule aussi. Les conséquences des risques liés à un manque d’expérience sur un logiciel ou à un bris d'équipement sont plus simples à régler, car ils sont soit réversibles, soit corrigibles. Le second plus grand facteur de risque sont les risques humains. Considérant la situation actuelle de la pandémie, il reste possible malgré toutes les précautions que l’un d’entre nous soit mit en arrêt de travail pendant deux semaines. Nous devons aussi nous assurer que les musiques et les vidéos que nous allons prendre dans notre projet soient libres de droits. Finalement, les risques de délais sont peu probables à avoir, puisque nous faisons tous des sauvegardes fréquemment et régulièrement des réunions d’équipe pour rester dans le projet.

**TESTS À RÉALISER - SCÉNARIO**

|  |  |
| --- | --- |
| **Scénario 1** | (Partie de Ghislain) |

|  |  |
| --- | --- |
| **Identification** | **Cas de test 1.1.1** |
| **Priorité** | 3 |
| **Date limite** | 15 Février |
| **Description** | Le but du test est de s’assurer que le site de diffusion soit compatible avec le projet. Ces tests se feront en ligne, sur le site provider par le professeur. |
| **Contraintes** | Le testeur devra avoir accès au site. Il devra s’assurer du bon fonctionnement des options du projet en ligne. |
| **Dépendances** | Tester la liaison entre le site et le projet.    Tester l’efficacité de l’interface utilisateur sur le site web. |
| **Procédure de test** | Données d’entrée : Glisser les slider sur l’interface en ligne.    Résultats attendus : Voir l’utilisateur changer les valeurs sur l’interface.    Critères de validation : Est-ce que le projet fonctionne correctement en ligne? |
| **Résultat** | Effectué par : Personne  Validation : NON  Site inexistant pour l’instant |

|  |  |
| --- | --- |
| **Scénario 2**  Vincent | Valérianne-Rose entre dans le site où est diffusé Concerto. Elle appuie sur “btn\_nuisance” et passe au mode “Bric-à-brac”. Le son passe donc dans une série de paramètres qu’elle peut changer en temps réel dans les contrôles du mode. Les contrôles du mode de base ont effet dans ce mode.  Valérianne-Rose appuie ensuite sur “btn\_overdrive” et passe au mode “Hapsichord Overdrive”. Le son quitte Bric-à-brac et est modifié avant d’être modulé par un “adsr”, contrôlé par le clavier qu’elle a accès dans les contrôles du mode. Les contrôles du mode de base ont effet dans ce mode. Valérianne-rose appuie finalement sur “btn\_base” et passe au mode de base à nouveau. Le son quitte Hapsichord Overdrive et sort directement dans les haut-parleurs de son ordinateur. |

|  |  |
| --- | --- |
| **Identification** | **Cas de test 2.1** |
| **Priorité** | 1 |
| **Date limite** | 11 décembre |
| **Description** | Le but du test est de vérifier si le bouton “btn\_nuisance” change en effet le son selon les paramètres de Bric-à-brac testés dans le cas de test 4.3. Les contrôles du mode de base devraient également fonctionner. |
| **Contraintes** | Le testeur devra avoir sur son poste d’ordinateur les logiciels Max8 et Open Stage Control. Il devra travailler en local avec le patcher\_principal\_v5.maxpat sur Max 8 et test\_osc.json sur Open Stage Control. |
| **Dépendances** | Tester la liaison entre le patcher\_principal\_v5.maxpat et le fichier test\_osc.json avant de commencer le test. |
| **Procédure de test** | Données d’entrée : L’utilisateur appuie sur “btn\_nuisance”  Résultats attendus : Le son continue à jouer, mais différent de l’original.  Critères de validation : Le son réagit-il aux contrôles du mode Bric-à-brac tel que mentionné dans le cas de test 4.3?  Le son **ne réagit-il pas** aux touches du clavier du mode Hapsichord Overdrive tel que mentionné dans le cas de test 4.2? Le son réagit-il aux contrôles du mode de base tel que mentionné dans le cas de test 3? |
| **Résultat** | Effectué par: Vincent  Validation : OUI |

|  |  |
| --- | --- |
| **Identification** | **Cas de test 2.2** |
| **Priorité** | *Selon la matrice de risques* |
| **Date limite** | 11 décembre |
| **Description** | Le but du test est de vérifier si le bouton “btn\_overdrive” change en effet le son selon les paramètres de Hapsichord Overdrive testés dans le cas de test 4.2. Les contrôles du mode de base devraient également fonctionner. |
| **Contraintes** | Le testeur devra avoir sur son poste d’ordinateur les logiciels Max8 et Open Stage Control. Il devra travailler en local avec le patcher\_principal\_v5.maxpat sur Max 8 et test\_osc.json sur Open Stage Control. |
| **Dépendances** | Tester la liaison entre le patcher\_principal\_v5.maxpat et le fichier test\_osc.json avant de commencer le test. |
| **Procédure de test** | Données d’entrée : L’utilisateur appuie sur “btn\_overdrive”  Résultats attendus : Le son se tait.  Critères de validation : Le son réagit-il aux touches du clavier du mode Hapsichord Overdrive tel que mentionné dans le cas de test 4.2?  Le son **ne réagit-il pas** aux contrôles du mode Bric-à-brac tel que mentionné dans le cas de test 4.3? Le son réagit-il aux contrôles du mode de base tel que mentionné dans le cas de test 3? |
| **Résultat** | Effectué par: Vincent  Validation : OUI |

|  |  |
| --- | --- |
| **Identification** | **Cas de test 2.3** |
| **Priorité** | 1 |
| **Date limite** | 11 décembre |
| **Description** | Le but du test est de vérifier si le bouton “btn\_base” change en effet le son selon les paramètres du mode de base testés dans le cas de test 3. Les contrôles du mode de base devraient également fonctionner. |
| **Contraintes** | Le testeur devra avoir sur son poste d’ordinateur les logiciels Max8 et Open Stage Control. Il devra travailler en local avec le patcher\_principal\_v5.maxpat sur Max 8 et test\_osc.json sur Open Stage Control. |
| **Dépendances** | Tester la liaison entre le patcher\_principal\_v5.maxpat et le fichier test\_osc.json avant de commencer le test. |
| **Procédure de test** | Données d’entrée : L’utilisateur appuie sur “btn\_base”  Résultats attendus : Le son se remet à jouer normalement.  Critères de validation :  Le son réagit-il aux contrôles du mode de base tel que mentionné dans le cas de test 3? Le son **ne réagit-il pas** aux touches du clavier du mode Hapsichord Overdrive tel que mentionné dans le cas de test 4.2? Le son **ne réagit-il pas** aux contrôles du mode Bric-à-brac tel que mentionné dans le cas de test 4.3? |
| **Résultat** | Effectué par: Vincent  Validation : OUI |

|  |  |
| --- | --- |
| **Scénario 3**  Marc-André | Une fois arrivé sur notre site, Ringo peut modifier ce qui se passe pendant l’expérience. Il peut faire cela par l’utilisation de l’interface présente sur ce site. L’utilisateur commence Concerto en faisant un clic sur le bouton Play et augmente le volume pour mieux entendre. Par la suite, il change ce qui est diffusé grâce à la l’utilisation de la playlist disponible dans le menu. Si l’utilisateur le veut, il peut aussi mettre le spectacle en pause. |

|  |  |
| --- | --- |
| Identification | Cas de test 3.1 |
| Priorité | 2 |
| Date limite | 11 décembre |
| Description | Le but de ce test est de déterminer si l’utilisateur peut démarrer et mettre en pause notre expérience. |
| Contraintes | L’utilisateur doit pouvoir mettre en marche ou en pause la musique et la projection de notre expérience par l'intermédiaire de notre site web. |
| Dépendances | S'assurer que le site fonctionne et que le bouton play et pause fonctionne même lorsqu'ils ne sont pas connectés au site. |
| Procédure de test | Données d’entrée : L’utilisateur appuis sur le bouton play/pause  Résultats attendus : Si la musique et la projection de l'expérience ne sont pas activées, elle commence. Si au contraire la musique et la projection jouent, elle s'arrête.  Critères de validation :la vidéo et la musique s’activent-elles et s'arrête-t-elle quand on appuie sur play/pause? |
| Résultat | Effectué par:Marc-André Bouchard  Validation: OUI  Sinon, ajouter un lien vers la fiche d’anomalie sur Github («bogue report») |

|  |  |
| --- | --- |
| Identification | Cas de test 3.2 |
| Priorité | 3 |
| Date limite | 11 décembre |
| Description | Le but de ce test est de déterminer si le slider permettant de modifier le volume de l'expérience |
| Contraintes | Il doit être possible de changer le volume de notre expérience par l'intermédiaire de notre site web. |
| Dépendances | S'assurer que le site fonctionne et que le slyder de volume fonctionne même lorsqu'ils ne sont pas connectés au site. |
| Procédure de test | Données d’entrée : Changer la valeur sur le slider du volume sur la page de notre site  Résultats attendus : Le volume de la projection change dépendamment de la valeur rentrée dans le slider.  Critères de validation : |
| Résultat | Effectué par : Marc-André Bouchard  Validation : OUI    Sinon, ajouter un lien vers la fiche d’anomalie sur Github («bogue report») |

|  |  |
| --- | --- |
| Identification | Cas de test 3.3 |
| Priorité | 2 |
| Date limite | 11 décembre |
| Description | Le but est de déterminer s' il est possible de changer la piste musicale et vidéo pas l’utilisation de la fonction de playlist présente dans l'expérience. |
| Contraintes | Il doit être possible de changer la vidéo et la piste musicales par l'intermédiaire de notre site web. |
| Dépendances | S'assurer que le site fonctionne et la fonction des playlists fonctionne même lorsqu'ils ne sont pas connectés au site. |
| Procédure de test | Données d’entrée: Changer la piste diffusée via l'intermédiaire de la playlist.  Résultats attendus : le changement de vidéo et de piste musical associé à celle choisie par l’utilisateur  Critères de validation : La piste musicale change-t-elle quand l’utilisateur change la donnée dans la playlist? La vidéo correspond-elle à la piste musicale sélectionnée? |
| Résultat | Effectué par : Marc-André Bouchard  Validation : OUI, OUI  Sinon, ajouter un lien vers la fiche d’anomalie sur Github («bogue report») |

|  |  |
| --- | --- |
| **Scénario 4**  Émile | Ringo arrive sur notre page web. Devant lui s’affiche le contrôleur midi Open Stage Control, avec trois sections distinctes (Base, Bric-à-brac et Harpsichord Overdrive), chacune contenant plusieurs éléments interactifs à essayer.  L’utilisateur veut modifier le son. S’offre à lui une panoplie de modificateur sonore, dont le slider de vitesse, l’«Harpsichord Overdrive» et le «Bric-à-Brac». |

|  |  |
| --- | --- |
| Identification | Cas de test 4.1.1 |
| Priorité | 2 |
| Date limite | 11 décembre |
| Description | Le but du test est de s’assurer que le slider vitesse modifie adéquatement la vitesse du média sonore sortant. Ces tests se feront en ligne, sur le logiciel Open Stage Control. |
| Contraintes | Le testeur devra avoir sur son poste d’ordinateur les logiciels Max8 et Open Stage Control. Il devra travailler en local avec le patcher\_principal\_v5.maxpat sur Max8 et test\_osc.json sur Open Stage Control. |
| Dépendances | Tester la liaison entre le patcher patcher\_principal\_v5.maxpat et le fichier test\_osc.json avant de commencer le test.  Tester l’efficacité du bouton «btn\_base» |
| Procédure de test | Données d’entrée : Glisser le slider «Vitesse» sur Open Stage Control  Résultats attendus : Observer une accélération ou une diminution de la vitesse de la vidéo émise dans le patcher patcher\_principal\_v5.maxpat.  Critères de validation : Est-ce que la vidéo dans le patcher est affectée par les changements de vitesse sur Open Stage Control? |
| Résultat | Effectué par : Émile Désilets  Validation : OUI |

|  |  |
| --- | --- |
| Identification | Cas de test 4.1.2 |
| Priorité | 3 |
| Date limite | 11 décembre |
| Description | Le but du test est de s’assurer que le bouton «default» du slider vitesse dans Open Stage Control affecte bel et bien la boîte message «0.95» lié au slider de vitesse dans patcher\_principal\_v5.maxpat. Ces tests se feront en ligne, sur le logiciel Open Stage Control. |
| Contraintes | Le testeur devra avoir sur son poste d’ordinateur les logiciels Max8 et Open Stage Control. Il devra travailler en local avec le patcher\_principal\_v5.maxpat sur Max8 et test\_osc.json sur Open Stage Control. |
| Dépendances | Tester la liaison entre le patcher patcher\_principal\_v5.maxpat et le fichier test\_osc.json avant de commencer le test.  Modifier la vitesse de la vidéo avec le slider vitesse  Tester l’efficacité du bouton «btn\_base» |
| Procédure de test | Données d’entrée : Appuyer sur le bouton «Default» sur Open Stage Control après avoir modifié la vitesse de la vidéo.  Résultats attendus : La vidéo devrait avoir une vitesse normale  Critères de validation : Est-ce que le bouton «Default» dans Open Stage Control affecte bel et bien la boîte message «0.95» lié au slider de vitesse dans patcher\_principal\_v5.maxpat? |
| Résultat | Effectué par : Émile Désilets  Validation : OUI |

|  |  |
| --- | --- |
| Identification | Cas de test 4.2 |
| Priorité | 1 |
| Date limite | 11 décembre |
| Description | Le but du test est de s’assurer que le clavier midi «Harpsichord Overdrive» sur Open Stage Control soit lié au clavier «Harpsichord Overdrive» dans le patcher\_principal\_v5.maxpat. |
| Contraintes | Le testeur devra avoir sur son poste d’ordinateur les logiciels Max8 et OpenStage Control. Il devra travailler en local avec le patcher\_principal\_v5.maxpat sur Max8 et test\_osc.json sur Open Stage Control. |
| Dépendances | Tester la liaison entre le patcher patcher\_principal\_v5.maxpat et le fichier test\_osc.json avant de commencer le test.  Tester l’efficacité du bouton «btn\_overdrive» |
| Procédure de test | Données d’entrée : Appuyer sur les touches du clavier dans test\_osc.json sur Open Stage Control.  Résultats attendus : Les touches devraient s’appuyer en même temps dans le patcher patcher\_principal\_v5.maxpat et dans test\_osc.json sur Open Stage Control.  Critères de validation : Est-ce que le clavier Open Stage Control affecte celui sur Max8? Si oui, sont-ils synchronisés à la seconde près? |
| Résultat | Effectué par : Émile Désilets  Validation : OUI, OUI |

|  |  |
| --- | --- |
| Identification | Cas de test 4.3.1 |
| Priorité | 1 |
| Date limite | 11 décembre |
| Description | Le but du test est de s’assurer que les deux dropdown «effets\_nuisance1» et «effets\_nuisance2» dans test\_osc.json sur Open Stage Control soient liés aux deux dropdown dans le sou patcher «nuisance» sur patcher\_principal\_v5.maxpat dans Max. |
| Contraintes | Le testeur devra avoir sur son poste d’ordinateur les logiciels Max8 et Open Stage Control. Il devra travailler en local avec le patcher\_principal\_v5.maxpat sur Max8 et test\_osc.json sur Open Stage Control. |
| Dépendances | Tester la liaison entre le patcher patcher\_principal\_v5.maxpat et le fichier test\_osc.json avant de commencer le test.  Tester l’efficacité du bouton «btn\_nuisance» |
| Procédure de test | Données d’entrée : Choisir une valeur dans un des dropdown dans test\_osc.json sur Open Stage Control. Refaire par la suite avec l’autre dropdown.  Résultats attendus : La valeur choisie dans le dropdown dans test\_osc.json sur Open Stage Control devrait changer la valeur dans le dropdown dans patcher\_principal\_v5.maxpat sur Max.  Critères de validation : Est-ce que les dropdown sur OpenStage Control affectent ceux dans Max? Si oui, sont-ils synchronisés à la seconde près? |
| Résultat | Effectué par : Émile Désilets  Validation : OUI, OUI |

|  |  |
| --- | --- |
| Identification | Cas de test 4.3.2 |
| Priorité | 2 |
| Date limite | 11 décembre |
| Description | Le but du test est de s’assurer que les deux knobs «knob\_nuisance1» et «knob\_nuisance2» sur test\_osc.json dans OpenStage Control soient liés aux deux boîtes «number» dans le sou patcher «nuisance» sur patcher\_principal\_v5.maxpat dans Max. |
| Contraintes | Le testeur devra avoir sur son poste d’ordinateur les logiciels Max8 et OpenStage Control. Il devra travailler en local avec le patcher\_principal\_v5.maxpat sur Max8 et test\_osc.json sur OpenStage Control. |
| Dépendances | Tester la liaison entre le patcher patcher\_principal\_v5.maxpat et le fichier test\_osc.json avant de commencer le test.  Tester l’efficacité du bouton «btn\_nuisance»  Tester l’efficacité des dropdowns «effets\_nuisance1» et «effets\_nuisance2» |
| Procédure de test | Données d’entrée : Tourner un des knobs dans test\_osc.json sur OpenStage Control. Refaire par la suite avec l’autre knob.  Résultats attendus : Tourner les knobs dans test\_osc.json sur OpenStage Control devrait changer la valeur des boîtes de type «number» nommées «fréquence» dans le sou patcher «nuisance» sur patcher\_principal\_v5.maxpat dans Max.  Critères de validation : Est-ce que les knobs sur OpenStage Control affectent les boîtes de type «number» nommées «fréquence» dans Max? Si oui, sont-ils synchronisés à la seconde près? |
| Résultat | Effectué par : Émile Désilets  Validation : OUI, OUI |

|  |  |
| --- | --- |
| **Scénario 5**  Jeremy | Ringo souhaite dynamiser son expérience en explorant les différents angles de caméras proposés. Devant lui, dans Open Stage Control, se trouvent quatre boutons correspondant aux quatre caméras disponibles: btn\_cam1, btn\_cam2, btn\_cam3 et btn\_cam4. Il clique sur le bouton btn\_cam2, ce qui affiche la vue de la caméra 2, et ainsi de suite pour le reste des boutons (le nombre après “cam” égale à la caméra sélectionnée). |

|  |  |
| --- | --- |
| Identification | Cas de test 5.1.1 |
| Priorité | 2 |
| Date limite | 11 décembre |
| Description | Le but du test est de vérifier que quand l’utilisateur appuie sur le bouton btn\_cam1 d’Open Stage Control affiche bien le rendu de la première caméra. |
| Contraintes | Le testeur devra avoir sur son poste d’ordinateur les logiciels Max8 et OpenStage Control. Il devra travailler en local avec le patcher\_principal\_v5.maxpat sur Max8 et test\_osc.json sur OpenStage Control. Le testeur doit aussi s’assurer que les quatre caméras soient connectées à obs. |
| Dépendances | Tester la liaison entre le patcher patcher\_principal\_v5.maxpat et le fichier test\_osc.json avant de commencer le test.  Tester les retours des quatre caméras  Tester l’efficacité du bouton «btn\_cam1» |
| Procédure de test | Données d’entrée : Appuyez sur le bouton btn\_cam1 Open Stage Control  Résultats attendus : La vue de la caméra 1 devrait être présentée dans l'interface web de l’utilisateur.  Critères de validation : Est-ce que la vue de la caméra 1 devrait être présentée dans l'interface web de l’utilisateur? |
| Résultat | Effectué par : Jeremy Levesque  Validation : OUI ou NON  Sinon, ajouter un lien vers la fiche d’anomalie sur Github («bogue report») |

|  |  |
| --- | --- |
| Identification | Cas de test 5.1.2 |
| Priorité | 2 |
| Date limite | 11 décembre |
| Description | Le but du test est de vérifier que quand l’utilisateur appuie sur le bouton btn\_cam2 d’Open Stage Control affiche bien le rendu de la deuxième caméra. |
| Contraintes | Le testeur devra avoir sur son poste d’ordinateur les logiciels Max8 et OpenStage Control. Il devra travailler en local avec le patcher\_principal\_v5.maxpat sur Max8 et test\_osc.json sur OpenStage Control. Le testeur doit aussi s’assurer que les quatre caméras soient connectées à obs. |
| Dépendances | Tester la liaison entre le patcher patcher\_principal\_v5.maxpat et le fichier test\_osc.json avant de commencer le test.  Tester les retours des quatre caméras  Tester l’efficacité du bouton «btn\_cam2» |
| Procédure de test | Données d’entrée : Appuyer sur le bouton btn\_cam2 Open Stage Control  Résultats attendus : La vue de la caméra 2 devrait être présentée dans l'interface web de l’utilisateur. |
| Résultat | Effectué par : Jeremy Levesque  Validation : OUI ou NON  Sinon, ajouter un lien vers la fiche d’anomalie sur Github («bogue report») |

|  |  |
| --- | --- |
| Identification | Cas de test 5.1.3 |
| Priorité | 2 |
| Date limite | 11 décembre |
| Description | Le but du test est de vérifier que quand l’utilisateur appuie sur le bouton btn\_cam3 d’Open Stage Control affiche bien le rendu de la troisième caméra. |
| Contraintes | Le testeur devra avoir sur son poste d’ordinateur les logiciels Max8 et OpenStage Control. Il devra travailler en local avec le patcher\_principal\_v5.maxpat sur Max8 et test\_osc.json sur OpenStage Control. Le testeur doit aussi s’assurer que les quatre caméras soient connectées à obs. |
| Dépendances | Tester la liaison entre le patcher patcher\_principal\_v5.maxpat et le fichier test\_osc.json avant de commencer le test.  Tester les retours des quatre caméras  Tester l’efficacité du bouton «btn\_cam3» |
| Procédure de test | Données d’entrée : Appuyez sur le bouton btn\_cam3 Open Stage Control  Résultats attendus : La vue de la caméra 3 devrait être présentée dans l'interface web de l’utilisateur.  Critères de validation : Est-ce que la vue de la caméra 3 devrait être présentée dans l'interface web de l’utilisateur. |
| Résultat | Effectué par : Jeremy Levesque  Validation : OUI ou NON  Sinon, ajouter un lien vers la fiche d’anomalie sur Github («bogue report») |

|  |  |
| --- | --- |
| Identification | Cas de test 5.1.4 |
| Priorité | 2 |
| Date limite | 11 décembre |
| Description | Le but du test est de vérifier que quand l’utilisateur appuie sur le bouton btn\_cam4 d’Open Stage Control affiche bien le rendu de la quatrième caméra. |
| Contraintes | Le testeur devra avoir sur son poste d’ordinateur les logiciels Max8 et OpenStage Control. Il devra travailler en local avec le patcher\_principal\_v5.maxpat sur Max8 et test\_osc.json sur OpenStage Control. Le testeur doit aussi s’assurer que les quatre caméras soient connectées à obs. |
| Dépendances | Tester la liaison entre le patcher patcher\_principal\_v5.maxpat et le fichier test\_osc.json avant de commencer le test.  Tester les retours des quatre caméras  Tester l’efficacité du bouton «btn\_cam4» |
| Procédure de test | Données d’entrée : Appuyer sur le bouton btn\_cam4 Open Stage Control  Résultats attendus : La vue de la caméra 4 devrait être présentée dans l'interface web de l’utilisateur. |
| Résultat | Effectué par : Jeremy Levesque  Validation : OUI ou NON  Sinon, ajouter un lien vers la fiche d’anomalie sur Github («bogue report») |

## **QUOI TESTER?**

**Fonctionnalité**

· Recevoir quatre feeds vidéos dans le logiciel OBS

· Intégrer des commandes dans OBS permettant de changer la vue de caméra

· Projeter les clips vidéo avec le projecteur

· Projeter la lumière des spots lumineux

· Les spots lumineux s’adaptent aux hautes et basses fréquences de la musique

**Intégration**

· Permettre à OpenStage Control de contrôler les commandes de changement de caméras sur OBS

· Brancher le projecteur sur l’ordinateur

· Mapper la zone de projection

· S’assurer que les spots lumineux ne nuisent pas à la projection des vidéos

· Lier les spots, le patcher Max et QLC+ ensemble

**Comptabilité/Performance**

· Vérifier la qualité des vidéos entrantes des caméras dans OBS

· S’assurer de ne pas avoir de décalage entre la vidéo provenant des caméras et du son provenant du patcher Max

· S’assurer de pouvoir maintenir un stream constant avec les caméras s’étirant sur plusieurs heures

· S’assurer de pouvoir brancher quatre caméras sur un seul ordinateur

· S’assurer que le projecteur et l’ordinateur soient compatibles l’un avec l’autre

· S’assurer que le lien entre Max, QLC+ et les spots soit solide et fiable