

ShaderComp

Cahier de tests



En partenariat avec la société Ubisoft

Sommaire

<i>Introduction.....</i>	<i>3</i>
<i>Avant propos</i>	<i>3</i>
<i>Objet.....</i>	<i>3</i>
<i>Responsabilités.....</i>	<i>3</i>
<i>Pré requis</i>	<i>4</i>
<i>Eléments à tester.....</i>	<i>4</i>
<i>Environnement de test</i>	<i>4</i>
<i>Equipe de test.....</i>	<i>4</i>
<i>Automatisation des tests.....</i>	<i>5</i>
<i>Fiches de tests</i>	<i>6</i>
<i>Création d'un projet</i>	<i>6</i>
<i>Test d'initialisation du projet</i>	<i>6</i>
<i>Test des fonctions de récupération des box</i>	<i>7</i>
<i>Edition des nœuds</i>	<i>8</i>
<i>Test de réinitialisation des nœuds d'une box</i>	<i>8</i>
<i>Test de réinitialisation d'ajout/suppression d'un nœud</i>	<i>9</i>
<i>Edition des variables</i>	<i>10</i>
<i>Test d'ajout d'une variable à la vertex box</i>	<i>10</i>
<i>Test d'ajout d'une variable à la pixel box.....</i>	<i>12</i>
<i>Test d'accès aux variables du pipeline</i>	<i>14</i>
<i>Gestion des liens.....</i>	<i>15</i>
<i>Test de récupération de la liste de liens</i>	<i>15</i>
<i>Test d'ajout de liens</i>	<i>16</i>
<i>Test de suppression de liens</i>	<i>17</i>
<i>Gestion du projet.....</i>	<i>18</i>
<i>Test de génération d'un projet.....</i>	<i>18</i>
<i>Test de sauvegarde et de chargement d'un projet.....</i>	<i>19</i>
<i>Test de sauvegarde et de chargement d'une box</i>	<i>20</i>

Introduction

Avant propos

Ce document a été développé par les étudiants d'ING-3, majeure SCIA de l'Ecole pour l'Informatique et les Techniques Avancées (EPITA) dans le cadre d'un projet de fin d'étude en entreprise. Ce projet a été réalisé en collaboration avec la société Ubisoft France dans le but de fournir un outil de compilation et d'optimisation de shaders.

Objet

Ce document spécifie l'ensemble des fiches de test de l'API développée. Ces tests seront présentés à partir du chapitre 3.

Responsabilités

Les responsables de ce document sont les membres de l'équipe de test, à savoir Matthieu BOURNAT, Antoine CHESNEAU et Romain GUYOT de la HARDROUYERE.

Pré requis

L'exécution des cas de test spécifiés dans le chapitre suivant requiert :

- Les éléments de l'API spécifiés au paragraphe 2.1
- L'environnement de test spécifié au paragraphe 2.2
- Le personnel en charge de l'exécution des tests spécifié en paragraphe 2.3

Éléments à tester

Le tableau ci-dessous montre la liste des éléments à tester. La dernière colonne doit être complétée manuellement avec l'identification des éléments listés.

Titre	Description	Version
shaderComp/	Intégralité des sources de la lib	

Environnement de test

Le tableau ci-dessous décrit la configuration requise pour les tests. La dernière colonne doit être complétée manuellement avec une identification des éléments de configuration utilisés.

Titre	Description	Identification
Windows / Linux	Système d'exploitation	
Python v 2.7.6	Version du langage	
Ligne de commande	Le dossier courant doit être le dossier parent de la bibliothèque	

Equipe de test

Le tableau ci-dessous présente les ressources en charge d'exécuter les tests. Il doit être complété manuellement avec le nom et la signature des personnes concernées. Une personne au moins doit remplir la première ligne.

Fonction	Nom	Signature
Chef Testeur		
Testeur assistant n°1		
Testeur assistant n°2		

Automatisation des tests

Un système d'automatisation des tests est proposé via la ligne de commande. Pour lancer la moulinette, taper la ligne suivante depuis l'interface en ligne de commande :

```
Python -m shaderComp.test_suite.core_test
```

Fiches de tests**Création d'un projet**

La procédure d'initialisation utilisée dans les tests ci-dessous nécessite le code suivant :

```
proj = Project.Project('project_tester')
```

Test d'initialisation du projet

Cas de test	CT-CP-01
Titre	Test d'initialisation du projet
Objectif	Vérifier que la création d'un nouveau projet fonctionne

Procédure de test	CT-CP-01		
Pré-requis	Importation de la classe « Project »		
ID	Actions à réaliser	Résultats attendu	OK
Initialisation			
1	myProject = Project.Project("project_tester")	myProject est différent de None	
2		myProject.linkManager est différent de None	
3		myProject.name est égal à "project_tester"	
4		myProject.pixelBox est différent de None	
5		myProject.vertexBox est différent de None	

Rapport de test	Testé par :		Le :	
Critère de succès / échec :			Approbation	
<input type="checkbox"/> Excellent : toutes les actions se sont déroulées sans le moindre problème				
<input type="checkbox"/> Bon : Une erreur ou moins est apparue				
<input type="checkbox"/> Moyen : plusieurs erreurs sont apparues				
<input type="checkbox"/> Faible : aucun test n'a réussi				
Commentaires :				

Test des fonctions de récupération des box

Cas de test	CT-CP-02
Titre	Test d'initialisation du projet
Objectif	Vérifier que les getters de box retournent bien ce qui est attendu

Procédure de test	CT-CP-02		
Pré-requis	Importation de la classe « Project »		
ID	Actions à réaliser	Résultats attendu	OK
Initialisation			
1	myProject = Project.Project("project_tester")	myProject.getVertexBox() retourne bien myProject.vertexBox	
2		myProject.getPixelBox() retourne bien myProject.pixelBox	
3		myProject.getBox("vertex") retourne bien myProject.vertexBox	
4		myProject.getBox("pixel") retourne bien myProject.pixelBox	

Rapport de test	Testé par :		Le :	
Critère de succès / échec :			Approbation	
<input type="checkbox"/> Excellent : toutes les actions se sont déroulées sans le moindre problème				
<input type="checkbox"/> Bon : Une erreur ou moins est apparue				
<input type="checkbox"/> Moyen : plusieurs erreurs sont apparues				
<input type="checkbox"/> Faible : aucun test n'a réussi				
Commentaires :				

Edition des nœuds

La procédure d'initialisation utilisée dans les tests ci-dessous nécessite le code suivant :

```
proj = Project.Project('project_tester')
myClamp = Clamp.Clamp(ShaderType.PIXEL_SHADER)
myDefaultVertexShader = DefaultVertexShader.DefaultVertexShader()
proj.appendNode(myClamp)
proj.appendNode(myDefaultVertexShader)
```

Test de réinitialisation des nœuds d'une box

Cas de test		CT-EN-01		
Titre		Test de réinitialisation des nœuds d'une box		
Objectif		Vérifier que la box réinitialisée est bien vide après la réinitialisation		
Procédure de test		CT-EN-01		
Pré-requis		Importation de la classe « Project »		
ID	Actions à réaliser		Résultats attendu	OK
Initialisation				
1	proj.clearBox("vertex")		La liste proj.vertexBox.nodeList est bien vide	
Initialisation				
2	proj.clearBox("pixel")		La liste proj.pixelBox.nodeList est bien vide	
Initialisation				
3	proj.clearBox("all")		La liste proj.pixelBox.nodeList est bien vide	
4			La liste proj.vertexBox.nodeList est bien vide	
Initialisation				
4	proj.clear()		La liste proj.pixelBox.nodeList est bien vide	
4			La liste proj.vertexBox.nodeList est bien vide	
Rapport de test		Testé par :	Le :	
Critère de succès / échec : <div><input type="checkbox"/> Excellent : toutes les actions se sont déroulées sans le moindre problème</div> <div><input type="checkbox"/> Bon : Une erreur ou moins est apparue</div> <div><input type="checkbox"/> Moyen : plusieurs erreurs sont apparues</div> <div><input type="checkbox"/> Faible : aucun test n'a réussi</div>				Approbation
Commentaires :				

Test de réinitialisation d'ajout/suppression d'un nœud

Cas de test	CT-EN-02
Titre	Test de réinitialisation d'ajout/suppression d'un nœud
Objectif	Vérifier que les nœuds sont correctement ajoutés / supprimés

Procédure de test	CT-EN-02		
Pré-requis	Importation de la classe « Project »		
ID	Actions à réaliser	Résultats attendu	OK
Initialisation			
1	myLog = Log.Log (ShaderType.PIXEL_SHADER) self.project.appendNode (myLog)	myLog est dans la liste proj.pixelBox.nodeList	
2	self.project.removeNode (myLog)	myLog n'est plus dans la liste proj.pixelBox.nodeList	
Initialisation			
3	myLog = Log.Log (ShaderType.VERTEX_SHADER) self.project.addNode (myLog, 0)	myLog est dans la liste proj.vertexBox.nodeList à la position 0	
4	self.project.removeNodeAt (0, 'vertex')	myLog n'est plus dans la liste proj.vertexBox.nodeList	

Rapport de test	Testé par :		Le :	
Critère de succès / échec :			Approbation	
<input type="checkbox"/> Excellent : toutes les actions se sont déroulées sans le moindre problème				
<input type="checkbox"/> Bon : Une erreur ou moins est apparue				
<input type="checkbox"/> Moyen : plusieurs erreurs sont apparues				
<input type="checkbox"/> Faible : aucun test n'a réussi				
Commentaires :				

Edition des variables

La procédure d'initialisation utilisée dans les tests ci-dessous nécessite le code suivant :

```
proj = Project.Project('project_tester')
```

Test d'ajout d'une variable à la vertex box

Cas de test		CT-EV-01		
Titre		Test d'ajout d'une variable à la vertex box		
Objectif		Vérifier que les variables sont correctement ajoutées		
Procédure de test		CT-EV-01		
Pré-requis		Importation de la classe « Project », et « Var »		
ID	Actions à réaliser		Résultats attendu	OK
Initialisation				
1	test_input_v = self.project.addVertexInVar('test_input_v', 'float')		La référence test_input_v est différente de None	
2			Le champ test_input_v.name est égal à "test input v"	
3			Le champ test_input_v.type est égal à "float"	
4			Le champ test_input_v.varType est égal à Var.VarType.IN	
4			La référence test_input_v est égale à proj.vertexBox. inVars["test input v"]	
4			La référence test_input_v est égale à proj.getVertexInVars ("test input v")	
Initialisation				
1	test_output_v = self.project.addVertexOutVar('test_output_v', 'float')		La référence test_output_v est différente de None	
2			Le champ test_output_v.name est égal à "test output v"	
3			Le champ test_output_v.type est égal à "float"	
4			Le champ test_output_v.varType est égal à Var.VarType.OUT	
4			La référence test_output_v est égale à proj.vertexBox. inVars["test output v"]	
4			La référence test_output_v est égale à proj.getVertexInVars ("test output v")	

Initialisation			
1	test_uni_v = self.project.addVertexUniform('test_uni_v', test_uni_v', 'float')	La référence test_uni_v est différente de None	
2		Le champ test_uni_v.name est égal à "test_uni_v"	
3		Le champ test_uni_v.type est égal à "float"	
4		Le champ test_uni_v.varType est égal à Var.VarType.UNI	
4		Le champ test_uni_v.val est égal à "test_uni_v"	
4		La référence test_uni_v est égale à proj.vertexBox. uniforms["test_uni_v"]	
4		La référence test_uni_v est égale à proj.getVertexUniform ("test_uni_v")	
Rapport de test		Testé par :	Le :
Critère de succès / échec : <ul style="list-style-type: none"> <input type="checkbox"/> Excellent : toutes les actions se sont déroulées sans le moindre problème <input type="checkbox"/> Bon : Une erreur ou moins est apparue <input type="checkbox"/> Moyen : plusieurs erreurs sont apparues <input type="checkbox"/> Faible : aucun test n'a réussi 			Approbation
Commentaires :			

Test d'ajout d'une variable à la pixel box

Cas de test		CT-EV-02		
Titre		Test d'ajout d'une variable à la pixel box		
Objectif		Vérifier que les variables sont correctement ajoutées		
Procédure de test		CT-EV-02		
Pré-requis		Importation de la classe « Project », et « Var »		
ID	Actions à réaliser		Résultats attendu	OK
Initialisation				
1	test_input_v = self.project.addPixelInVar('test_input_v', 'float')		La référence test_input_p est différente de None	
2			Le champ test_input_p.name est égal à "test_input_p"	
3			Le champ test_input_p.type est égal à "float"	
4			Le champ test_input_p.varType est égal à Var.VarType.IN	
4			La référence test_input_p est égale à proj.pixelBox. inVars["test_input_p"]	
4			La référence test_input_p est égale à proj.getPixelInVars ("test_input_p")	
Initialisation				
1	test_output_p = self.project.addPixelOutVar('test_output_p', 'float')		La référence test_output_p est différente de None	
2			Le champ test_output_v.name est égal à "test_output_p"	
3			Le champ test_output_p.type est égal à "float"	
4			Le champ test_output_p.varType est égal à Var.VarType.OUT	
4			La référence test_output_p est égale à proj.pixelBox. inVars["test_output_p"]	
4			La référence test_output_p est égale à proj.getPixelInVars ("test_output_p")	
Initialisation				
1	test_uni_p = self.project.addVertexUniform('test_uni_p', test_uni_p', 'float')		La référence test_uni_p est différente de None	
2			Le champ test_uni_p.name est égal à "test_uni_p"	
3			Le champ test_uni_p.type	

		est égal à "float"	
4		Le champ test_uni_p.varType est égal à Var.VarType.UNI	
4		Le champ test_uni_p.val est égal à "test_uni_p"	
4		La référence test_uni_v est égale à proj.pixelBox. uniforms["test_uni_p"]	
4		La référence test_unit_p est égale à proj.getPixelUniform ("test_uni_p")	
Rapport de test		Testé par :	Le :
Critère de succès / échec : <ul style="list-style-type: none"> <input type="checkbox"/> Excellent : toutes les actions se sont déroulées sans le moindre problème <input type="checkbox"/> Bon : Une erreur ou moins est apparue <input type="checkbox"/> Moyen : plusieurs erreurs sont apparues <input type="checkbox"/> Faible : aucun test n'a réussi 			Approbation
Commentaires :			

Test d'accès aux variables du pipeline

Cas de test		CT-EV-03		
Titre		Test d'accès aux variables du pipeline		
Objectif		Vérifier que les variables sont correctement ajoutées		
Procédure de test		CT-EV-03		
Pré-requis		Importation de la classe « Project »		
ID	Actions à réaliser		Résultats attendu	OK
Initialisation				
1	test_pipeline = getVertexPipelineVar('Vertex')		La référence test_pipeline est différente de None	
2	test_pipeline = getVertexPipelineVar('Normal')		La référence test_pipeline est différente de None	
3	test_pipeline = getVertexPipelineVar('Position')		La référence test_pipeline est différente de None	
4	test_pipeline = getVertexPipelineVar('Color')		La référence test_pipeline est différente de None	
5	test_pipeline = getVertexPipelineVar('FragColor')		La référence test_pipeline est différente de None	
6	test_pipeline = getVertexPipelineVar('FragCoord')		La référence test_pipeline est différente de None	
7	test_pipeline = getVertexPipelineVar('FogColor')		La référence test_pipeline est différente de None	
8	Réitérer les tests précédents pour les variables du pipeline du pixel shader (en remplaçant « Vertex » par « Pixel »)		Mêmes résultats attendus	
Rapport de test		Testé par :	Le :	
Critère de succès / échec : <div><input type="checkbox"/> Excellent : toutes les actions se sont déroulées sans le moindre problème</div> <div><input type="checkbox"/> Bon : Une erreur ou moins est apparue</div> <div><input type="checkbox"/> Moyen : plusieurs erreurs sont apparues</div> <div><input type="checkbox"/> Faible : aucun test n'a réussi</div>			Approbation	
Commentaires :				

Gestion des liens

La procédure d'initialisation utilisée dans les tests ci-dessous nécessite le code suivant :

```
project = Project.Project('project_tester')
myClamp = Clamp.Clamp(ShaderType.PIXEL_SHADER)
myDefaultVertexShader = DefaultVertexShader.DefaultVertexShader()
project.appendNode(myClamp)
project.appendNode(myDefaultVertexShader)
```

Test de récupération de la liste de liens

Cas de test		CT-GL-01		
Titre		Test de récupération de la liste de liens		
Objectif		Vérifier que les liens sont accessibles		
Procédure de test		CT-GL-01		
Pré-requis		Importation de tous les éléments des modules shaderComp.core et shaderComp.shaders		
ID	Actions à réaliser		Résultats attendu	OK
Initialisation				
1	project.addValueLinkByName (myDefaultVertexShader, 'vertex', 1.0)		La longueur de la liste retournée par l'appel project.getLinkList ("vertex") est bien égale à 1	
2	Project.addValueLink (myClamp.getInVar ('min'), 0.0)		La longueur de la liste retournée par l'appel project.getLinkList ("pixel") est bien égale à 1	
3	project.addValueLinkNode (myClamp, myClamp.getInVar ('max'), 1.0)		La longueur de la liste retournée par l'appel project.getLinkList ("pixel") est bien égale à 2	
Rapport de test		Testé par :		Le :
Critère de succès / échec : <div><input type="checkbox"/> Excellent : toutes les actions se sont déroulées sans le moindre problème</div> <div><input type="checkbox"/> Bon : Une erreur ou moins est apparue</div> <div><input type="checkbox"/> Moyen : plusieurs erreurs sont apparues</div> <div><input type="checkbox"/> Faible : aucun test n'a réussi</div>				Approbation
Commentaires :				

Test d'ajout de liens

Cas de test		CT-GL-02		
Titre		Test d'ajouts de liens		
Objectif		Vérifier que les fonctions d'ajout de liens n'ont pas d'erreur		
Procédure de test		CT-GL-02		
Pré-requis		Importation de tous les éléments des modules shaderComp.core et shaderComp.shaders		
ID	Actions à réaliser		Résultats attendu	OK
Initialisation				
1	Project.addValueLink(myClamp.getInVar('min'), 0.0)		project.linkManager.containLinkTo(myClamp, myClamp.getInVar('min')) renvoie True	
2	project.addValueLinkNode(myClamp, myClamp.getInVar('max'), 1.0)		project.linkManager.containLinkTo(myClamp, myClamp.getInVar('max')) renvoie True	
3	project.addValueLinkByName(myDefaultVertexShader, 'vertex', 1.0)		project.linkManager.containLinkTo(myDefaultVertexShader, myDefaultVertexShader.getInVar('vertex')) renvoie True	
Initialisation				
4	myLog = Log.Log(ShaderType.PIXEL_SHADER) project.appendNode(myLog)		project.linkManager.containLinkTo(myLog, myLog.getInVar('input')) renvoie False	
5	project.addLink(myClamp.getOutVar('result'), myLog.getInVar('input'))		project.linkManager.containLinkTo(myLog, myLog.getInVar('input')) renvoie True	
Initialisation				
4	myLog = Log.Log(ShaderType.PIXEL_SHADER) project.appendNode(myLog)			
5	project.addLinkNode(myClamp, myClamp.getOutVar('result'), myLog, myLog.getInVar('input'))		project.linkManager.containLinkTo(myLog, myLog.getInVar('input')) renvoie True	
Rapport de test		Testé par :	Le :	
Critère de succès / échec : <div><input type="checkbox"/> Excellent : toutes les actions se sont déroulées sans le moindre problème</div> <div><input type="checkbox"/> Bon : Une erreur ou moins est apparue</div> <div><input type="checkbox"/> Moyen : plusieurs erreurs sont apparues</div> <div><input type="checkbox"/> Faible : aucun test n'a réussi</div>			Approbation	
Commentaires :				

Test de suppression de liens

Cas de test		CT-GL-03		
Titre		Test de suppression de liens		
Objectif		Vérifier que les fonctions de suppression de liens n'ont pas d'erreurs		
Procédure de test		CT-GL-03		
Pré-requis		Importation de tous les éléments des modules shaderComp.core et shaderComp.shaders		
ID	Actions à réaliser		Résultats attendu	OK
Initialisation				
1	myLog = Log.Log(ShaderType.PIXEL_SHADER) self.project.appendNode(myLog)			
2	project.addLink(myClamp.getOutVar('result'), myLog.getInVar('input'))		project.linkManager.containLinkTo(myLog, myLog.getInVar('input')) renvoie True	
3	project.deleteLink(myClamp, myLog, myClamp.getOutVar('result'), myLog.getInVar('input'))		project.linkManager.containLinkTo(myLog, myLog.getInVar('input')) renvoie False	
4	project.addValueLink(myLog.getInVar('input'), 0.0)		project.linkManager.containLinkTo(myLog, myLog.getInVar('input')) renvoie True	
5	project.deleteValueLink(myLog, myLog.getInVar('input'))		project.linkManager.containLinkTo(myLog, myLog.getInVar('input')) renvoie False	
Rapport de test		Testé par :	Le :	
Critère de succès / échec : <div><input type="checkbox"/> Excellent : toutes les actions se sont déroulées sans le moindre problème</div> <div><input type="checkbox"/> Bon : Une erreur ou moins est apparue</div> <div><input type="checkbox"/> Moyen : plusieurs erreurs sont apparues</div> <div><input type="checkbox"/> Faible : aucun test n'a réussi</div>			Approbation	
Commentaires :				

Gestion du projet

La procédure d'initialisation utilisée dans les tests ci-dessous nécessite le code suivant :

```
project = Project.Project('test_project')
myColor = Color.Color()
myColor = myColor
myDefaultVertexShader = DefaultVertexShader.DefaultVertexShader()
myDefaultVertexShader = myDefaultVertexShader

project.appendNode(myColor)
project.appendNode(myDefaultVertexShader)
myColor.setParams(0, 0.3, 0.6, 1.0);
vertexInVertVar = self.project.addVertexInVar('vertex', 'vec4')
vertexOutFinalPositionVar = self.project.addVertexOutVar('final_position',
'vec4')
pixelOutFinalColor = self.project.addPixelOutVar('final_color', 'vec4')

project.addLink(vertexInVertVar, myDefaultVertexShader.getInVar('vertex'))
project.addLink(myDefaultVertexShader.getOutVar('position'),
vertexOutFinalPositionVar)
project.addLink(myColor.getOutVar('color'), pixelOutFinalColor)
project.addLink(self.project.getVertexPipelineVar('Vertex'),
vertexInVertVar)
project.addLink(vertexOutFinalPositionVar,
project.getVertexPipelineVar('Position'))
project.addLink(pixelOutFinalColor,
project.getPixelPipelineVar('FragColor'))
```

Test de génération d'un projet

Cas de test		CT-GP-01		
Titre		Test de génération d'un projet		
Objectif		Vérifier que le code source du shader est correctement généré		
Procédure de test		CT-GP-01		
Pré-requis		Importation de tous les éléments des modules shaderComp.core et shaderComp.shaders		
ID	Actions à réaliser		Résultats attendu	OK
Initialisation				
1	project.compute('GLSLPrinter')		Le dossier « test_project » existe	
2			Le fichier « vertexShader.glsl » existe dans le dossier « test_project »	
3			Le fichier « pixelShader.glsl » existe dans le dossier « test_project »	
Restauration		Suppression du répertoire « test_project » et des fichiers qu'il contient		
Rapport de test		Testé par :	Le :	
Critère de succès / échec : <div><input type="checkbox"/> Excellent : toutes les actions se sont déroulées sans le moindre problème <input type="checkbox"/> Bon : Une erreur ou moins est apparue <input type="checkbox"/> Moyen : plusieurs erreurs sont apparues <input type="checkbox"/> Faible : aucun test n'a réussi</div>				Approbation
Commentaires :				

Test de sauvegarde et de chargement d'un projet

Cas de test	CT-GP-02		
Titre	Test de sauvegarde et de chargement d'un projet		
Objectif	Vérifier que les procédures de sauvegarde / chargement fonctionnent		
Procédure de test	CT-GP-02		
Pré-requis	Importation de tous les éléments des modules shaderComp.core et shaderComp.shaders		
ID	Actions à réaliser	Résultats attendu	OK
Initialisation			
1	project.save('test_project_sav.bin')	Le fichier « test_project_sav.bin » existe	
1	newproj = Project.Project.load('test_project_sav.bin')	La référence newproj est différente de None	
1	newproj.compute('GLSLPrinter')	La référence newproj est différente de None	
2		Le fichier « vertexShader.glsl » existe dans le dossier « test_project »	
3		Le fichier « pixelShader.glsl » existe dans le dossier « test_project »	
Restauration	Suppression du fichier « test_project_sav.bin », du répertoire « test_project » et des fichiers qu'il contient		
Rapport de test	Testé par :		Le : Approbation
Critère de succès / échec : <input type="checkbox"/> Excellent : toutes les actions se sont déroulées sans le moindre problème <input type="checkbox"/> Bon : Une erreur ou moins est apparue <input type="checkbox"/> Moyen : plusieurs erreurs sont apparues <input type="checkbox"/> Faible : aucun test n'a réussi			
Commentaires :			

Test de sauvegarde et de chargement d'une box

Cas de test		CT-GP-03		
Titre		Test de sauvegarde et de chargement d'une box		
Objectif		Vérifier que les procédures de sauvegarde / chargement des box fonctionnent		
Procédure de test		CT-GP-03		
Pré-requis		Importation de tous les éléments des modules shaderComp.core et shaderComp.shaders		
ID	Actions à réaliser		Résultats attendu	OK
Initialisation				
1	project.saveBox('test_project_pixel_box.bin', 'pixel') project.saveBox('test_project_vertex_box.bin', 'vertex')		Les fichiers « test_project_pixel_box.bin » et « test_project_vertex_box.bin » existent	
1	newProj = Project.Project('load_box_test') newProj.loadBox('test_project_pixel_box.bin', 'pixel') newProj.loadBox('test_project_vertex_box.bin', 'vertex')		La référence newproj est différente de None Aucune erreur ne survient lors du chargement des box	
1			Le champs newproj.pixelBox.name vaut "Pixeltest project"	
2			Le champs newproj.vertexBox.name vaut "Vertextest project"	
3			Le shader Color est bien dans la liste de nœuds de la pixel box du projet chargé	
3			Le shader DefaultVertexShader est bien dans la liste de nœuds de la vertex box du projet chargé	
3	newProj.appendNode(newProj.loadBoxAsNode('test_project_pixel_box.bin'))		Le champs newProj.pixelBox.nodeList [1].name vaut bien "Pixeltest project"	
Restauration		Suppression du fichier « test_project_sav.bin », du répertoire « test_project » et des fichiers qu'il contient		
Rapport de test		Testé par :		Le :
Critère de succès / échec : <div><input type="checkbox"/> Excellent : toutes les actions se sont déroulées sans le moindre problème</div> <div><input type="checkbox"/> Bon : Une erreur ou moins est apparue</div> <div><input type="checkbox"/> Moyen : plusieurs erreurs sont apparues</div> <div><input type="checkbox"/> Faible : aucun test n'a réussi</div>				Approbation
Commentaires :				