



État de développement

Fonctionnalités :

L'application est complètement fonctionnelle cependant de nombreuses fonctionnalités n'ont pas été implémentées.

Le système de map avec tile est parfaitement fonctionnel. Son optimisation par chunk est également fonctionnelle. Il est possible de modifier la map pendant le jeu. Certaines tiles peuvent interagir entre elles. Le système de sauvegarde/chargement de la map à partir d'un fichier .eagl est fonctionnel. Le système de lumière est fonctionnel et commun aux tiles et entités.

Les événements sont également fonctionnels ils peuvent se faire par "broadcast" ou par un lien plus précis (entre deux instances de classe).

Les entités ainsi que toutes ses dérivées (joueur, monstres, items et entités générales) sont elles aussi fonctionnelles ainsi que les collisions entre entités/entités et entités/tiles.

Le système d'inventaire est fonctionnel malgré le fait que les armures et la partie équipement n'a pas été réalisée ainsi que quelques bugs concernant le déplacement d'items entre eux dans l'inventaire. Le système d'expérience n'a finalement pas été fait pour s'orienter vers un système de progression par armes/armures (plus le joueur avancera dans le jeu plus il se verra doté d'armes et d'armures puissantes pour se débarrasser des ennemis).

Problèmes rencontrés :

Nous n'avons pas rencontré de problèmes particuliers. Les seuls problèmes rencontrés furent des problèmes facilement résolubles (problème d'inattention ou code mal écrit).



Lancement et utilisation :

Le jeu se lance par la classe LauncherTester. Il n'est pas nécessaire de lui fournir de paramètres.

Les contrôles en jeu sont :

- Z : se déplacer en haut (mode édition)
- Q : se déplacer a gauche
- S : se déplacer en bas (mode édition)
- D : se déplacer a droite
- Space: sauter
- V : vsync on/off
- 1-8 / Molette : sélectionner un item de la quick-bar
- U : Ouvrir le menu de sélection de tile
- I : Ouvrir l'inventaire
- Escape : Fermer le menu de sélection de tile/Menu de pause
- Left Click : Utiliser l'item en main / placer une tile (mode édition)
- Right Click : Sélectionner la variante de tile dans le menu de sélection de tile / supprimer une tile (mode édition)
- B : passer de l'édition du premier plan a celle du second plan et inversement (mode édition)
- Tab : mode édition on/off
- Shift+Left Click : sélectionner une zone a remplir (mode édition)
- Shift+Right Click : sélectionner une zone a supprimer (mode édition)
- C : changer la variante de la tile sélectionnée
- R : tourner la tile sélectionnée



Test de JUnit :

Les tests unitaires ont été fait sur la classe Math et Inventory. Le test de Math peut être lancé directement avec JUnit cependant Inventory étant un objet possédant des GUI, il a besoin d'initialiser ceux-ci. C'est pourquoi le test est fait au lancement du jeu et s'affiche dans la console au lieu de passer par l'affichage de JUnit.

Math :

Truncate (number, decimal)		
Expected	Actual	Result
1f	1.12345f, 0	1f
1.1f	1.12345f, 1	1.1f
1.12f	1.12345f, 2	1.12f
1.123f	1.12345f, 3	1.123f
1.1234f	1.12345f, 4	1.1234f

Pow (number, decimal)		
Expected	Actual	Result
4f	2f, 2	4f
256f	2f, 8	256f
16f	4f, 2	16f
216f	6f, 3	216f
1f	6f, 0	1f

fastAbs(number)



Expected	Value	Actual
1f	1f	1f
1f	-1f	1f
145f	145f	145f
.42f	-.42f	.42f
1516451651f	1516451651f	1516451651f

fastFloor(number)		
Expected	Value	Actual
1	1.42f	1
512	512.9999f	512
-6	-5.1f	-6
0	.42f	0
-1912	-1911.9999f	-1912

round(number, decimal)		
Expected	Value	Actual
1.11f	1.106f, 2	1.11f
1.12f	1.116f, 2	1.12f
-101.666f	-101.6664f, 3	-101.666f
0f	.000001f, 4	0f
1.5f	-1.5f, 1	1.5f

Inventory :



Add (item, stack)		
Expected	Actual	Result
true	get(0), 1	true
true	get(0), 49	true
false	get(0), 1	false
true	get(2), 1	true
false	get(2), 1	false

Remove (item, stack)		
Expected	Actual	Result
true	get(0), 1	true
true	get(0), 49	true
false	get(0), 1	false
true	get(2), 1	true
false	get(2), 1	false

getArrow (item)		
Expected	Actual	Result
null	null	null
null	get(2)	null
Arrow	get(0)	Arrow