

WORM U X



Plan

- Historique
- Qu'est-ce que Wormux
- Règles du jeu
- La licence
- (démonstration)
- Développement
- Travail collaboratif
- Graphisme
- Conception
- Développements futurs



Historique...

- 2001, Date de lancement.
- 2003, arrivé de Victor Stinner puis d'un grand nombre de contributeurs.
- Coup publicitaire en 2004-2005.



Qu'est ce que Wormux ?

- Rappel des règles du jeu.
- La licence choisie.
- Démonstration.



Règles du jeu

1. Sélection d'un avatar.



2. Choix d'une arme.

3. Tirer sur les avatars adverses.





La Licence choisie

The GPL

General Public License

- Liberté d'utilisation.
- Un libre accès aux sources du projet pour les modifier.
- Une liberté de redistribution.



Démonstration



Développement

- Quatre développeurs actifs : lami, lodesi, matthieu, et haypo.
- Soucis d'hébergement avec TuxFamily.org
- Utilisation de CVS et d'IRC.



Développement (suite)

- Passage de ClanLib 0.6 à la version 0.7
- Coding Party
- Critique du jeu sur happypenguin.org (langue française)



Travail collaboratif

- $\frac{3}{4}$ des contributions viennent par email.
- Peu de patch, beaucoup de rapport de bugs.
- Effets sonores de Toddd.



Travail collaboratif (suite)

- Jeu internationalisé.
- Site web internationalisé.
- Equipe internationale.



Graphisme

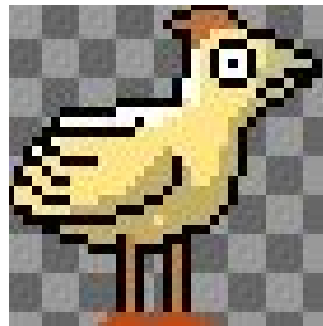
Partie intégrante du projet, quantité importante du travail avec le codage.

- Besoins en terrains, avatars, animations.
- Essayer de garder le style « cartoon » des Worms originaux.

Graphisme (suite)

Utilisation du format PNG.

- Avantage d'un format libre, une couche "Alpha", compression sans destruction...

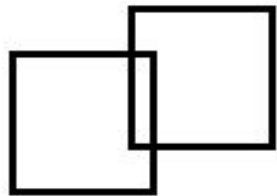




Conception

Moteur de gestion des phénomènes physiques

La plupart des objets de Wormux (avatars, projectiles, vent...) sont gérés par un moteur physique. Son rôle est de gérer les :



- Trajectoires
- Collisions
- Rebonds





Conception (suite)

Gestionnaire de terrains

Son rôle est de gérer les dégâts dû aux explosions, tirs...

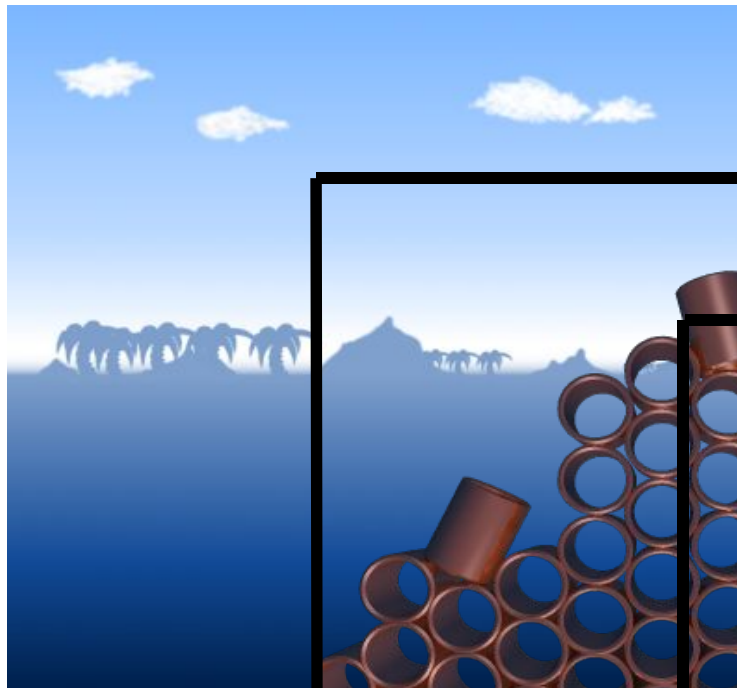
1. Chargement d'un terrain en mémoire.
2. Affichage.
3. Explosion d'un morceau de terrain.
4. Affichage (avec mise en transparent des pixels détruits).

Puis bouclage des opérations 2. 3. 4.

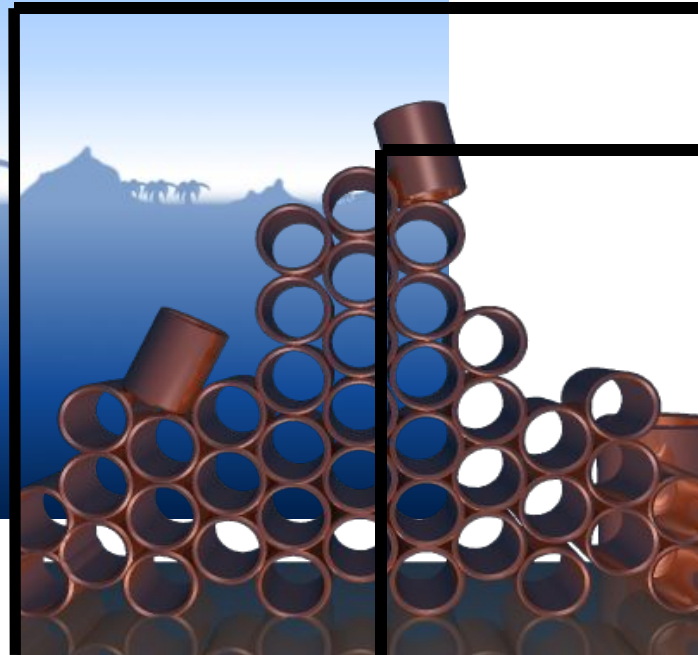


Conception (re-suite)

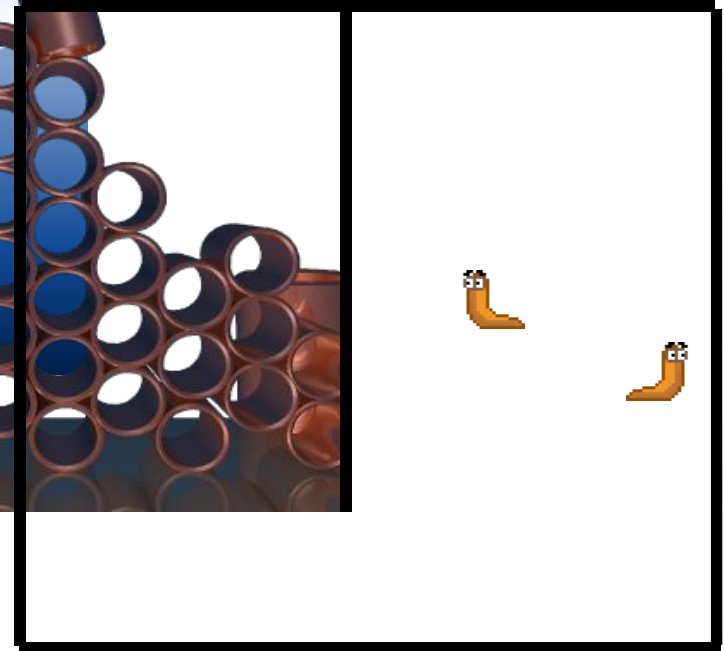
1



2



3





Développements futurs

- Corde ninja.
- Jeu en réseau.
- Générateur de terrain aléatoire.



Développements futurs (suite et fin)

- Version Windows.
- Intelligence Artificielle.
- Réécriture du jeu avec SDL ? :-)



Le mot de la fin

www.wormux.org

Paquets Mandrake, Gentoo, Sorcerer,
SourceMage, FreeBSD, Debian, ...