

Necrew Arena

<https://github.com/mariusvitta/tacticsarena>

Vivien-Junior Obanda, Dylan Renaudin, Alexandre Danjou et
Marius Vitta

Le Mans Université

23 avril 2019

Présentation du jeu : Tactics arena

- Jeu de stratégie
- Règles du jeu
 - Affrontement de deux équipes de personnages sur une carte de type échiquier
 - Combat au tour par tour
 - Soigneurs chargés de rendre des points de vie aux personnages
 - Attaquants chargés de tuer les personnages de l'équipe adverse
 - Choix de l'ordre de jeu des personnages par les joueurs

Conception

- Reprise des règles classiques de jeu à quelques exceptions
- Règles de jeu remodelées
 - Tours de jeu alternés suivant un ordre défini en début de partie
 - Équipes de 2 personnages dotés de quatre sorts différents
- 4 classes de personnages, 16 sorts en tout
- Structures de données pour les sorts, les personnages et les équipes
- Matrice de caractères pour la carte de jeu

Conception

■ Organisation du travail

- Dylan Renaudin : structures de données, gestion des déplacements, sorts et modification du code pour le rendre plus générique
- Marius Vitta : initialisation de la partie, gestion des tours de jeu et intelligence artificielle
- Vivien-Junior Obanda : affichages, mise à jour, sorts et interface graphique
- Alexandre Danjou : sorts et réseau

Jeu version console

- Matrice de caractères de taille 11x11
- Structure de données pour les personnages et les équipes
- 4 catégories de sorts
 - De modification de positions : *Saut, Attire*
 - D'application de dégâts : *GrosCoup, Diago* ...
 - De modification de statistiques : *Armure, Transformations* ...
 - De soins : *Revitalisation, Soin*
- Fonctions majeures
 - D'initialisation du jeu
 - De déplacement des personnages
 - De mise à jour et d'affichage du plan de jeu
 - De tour de jeu
 - De création et de suppression des personnages et des sorts
 - Une fonction par sort

Intelligence Artificielle

- Algorithme Min-Max
 - Choix du meilleur coup possible
 - Deux fonctions d'évaluation mutuellement récursives : joueur et adversaire
- Utilisation de la portée
- Choix du sort infligeant le plus de dégâts
- Si l'adversaire est hors portée : procéder au déplacement
 - Vers l'ennemi le plus proche ayant le moins de points de vie
 - Case par case afin de trouver la direction vers le point d'arrivée
- Création d'un type énuméré comprenant les quatre points cardinaux

Réseau

- Modèle Client-Serveur
- Mise à jour du plan de jeu uniquement du côté du serveur
- Adaptation de fonctions existantes
 - Ajouts de paramètres et création de variables globales
- Réimplémentation de fonction : rajout de vérifications utiles au réseau
- Création des fonctions serveur et client
 - *send_all* : transmet des informations à tous, sauf à un client
 - Fonction Serveur : met à jour la map, exécute la partie, transmet les informations aux clients et vérifie les réponses
 - Fonction Client : reçoit les informations et envoie les actions au serveur

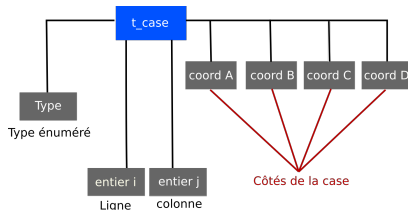
Interface graphique

■ Bibliothèques SDL

- *SDL_images* et *SDL_ttf*

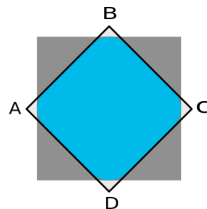
■ Affichage du plan de jeu :

- Ligne par ligne en ordre décroissant :
superposition des textures respectée
- Plan de jeu isométrique :
cases en losanges



Interface graphique

- Menu de jeu et de sort
- Adaptation et réimplémentation de fonctions
 - Cases bleues pour les portées
 - Récupération des clics de l'utilisateur
 - Effet de clignotement sur le personnage touché
- Fonction de recherche de case
- Vérification de position des clics dans la case
 - Calcul des équations des quatre droites de la case
 - Clic compris entre (AB), (BC), (CD) et (AD)

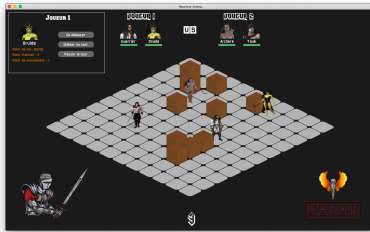


Résultats

- IA fonctionnel pour les sorts de portées et de dégâts positifs
- Sorts de portées ou de dégâts nuls non utilisés
- Dégâts infligés à l'adversaire

Résultats

■ Interface Graphique : Necrew Arena - Tactics Arena



[illegible]

Améliorations possibles

- Création de tableaux de personnages
- Ajout d'autres modes de jeu
- Sauvegarde de la partie
- Intelligence Artificielle
 - Évaluation de l'état du personnage
 - Prise en compte de la portée
- Réseau
 - Gestion d'un plus grand nombre de joueurs
 - Implémentation d'une version SDL en réseau
- Interface graphique
 - Intégration d'animations et de bruitages
 - Gestion de l'élargissement de la fenêtre
 - Rajout d'une aide reposant sur l'IA

Conclusion

- Contraintes principales respectées
 - Version terminal, réseau et SDL fonctionnels et intégrés au jeu
 - Intelligence Artificielle fonctionnelle, non intégrée
- Réalisation dans les temps
- Apports du module
 - Travail en groupe
 - Utilisation d'outils : *git*, *gdb*, *doxygen*
 - Approfondissement des connaissances sur la compilation séparée
 - Découverte de nouvelles bibliothèques
 - Développement de capacités de recherches et d'auto-apprentissage