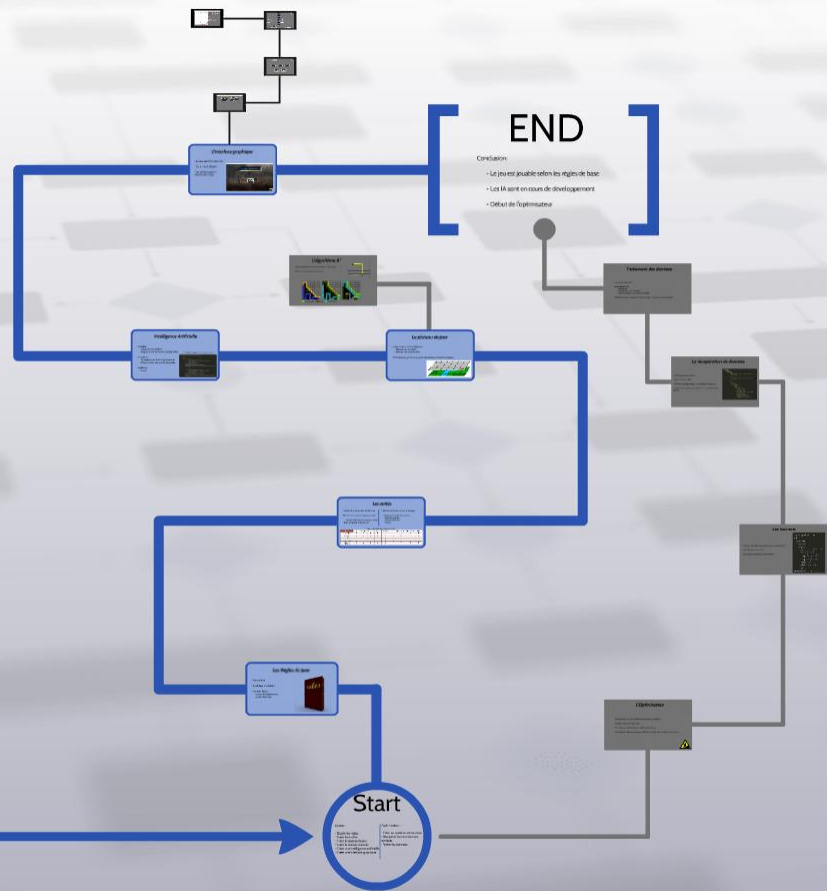




groupe 2 Faquin

Optimisateur de Wargame



Start

Le jeux :

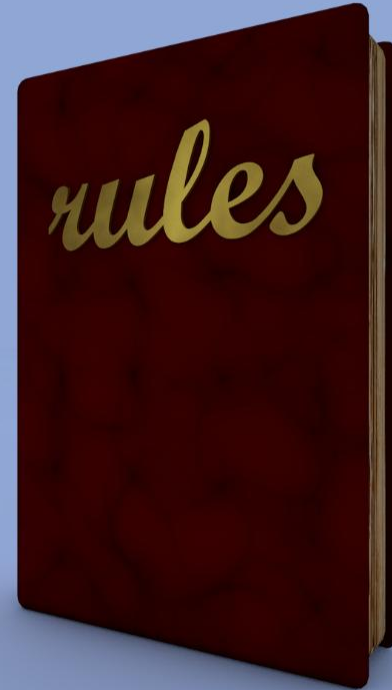
- Etablir les règles
- Créer les unités
- Créer le plateau de jeu
- Créer la version console
- Créer une intelligence artificielle
- Créer une interface graphique

L'optimisateur :

- Créer un système de tournois
- Récupérer les données des combats
- Traiter les données

Les Règles du jeux

- Tour par tour
- Jeu de type annihilation
- 2 phases de jeu:
 - phase de déplacement
 - phase d'attaque



Les unités

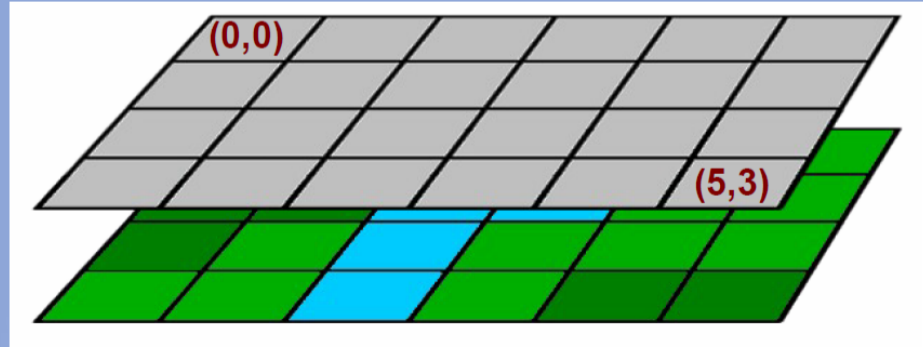
- Création d'un dictionnaire de référence
- Utilisation de fichier csv pour le stockage
- Objet Armee contenant toutes les unités
 - Calcul du prix selon trois critères
 - Potentiel offensif
 - Potentiel défensif
 - Portée
 - Module dédié dans le code pour l'objet Unite héritant de l'objet Armee

tableau des unités de l'armée des humains

HUMAINS	PV	ARMURE	PRECISION	ATQ	DÉPLACEMENT	PORTÉE	TAUX D'ESQUIVE	TAUX DE CRIT	PRIX
ENROLÉ	10	2	50	10	6	1	1	1	24
ÉPÉISTE	25	10	60	20	5	1	5	5	56
LANCIER	25	10	60	20	5	2	2	5	56
CHEVALIER	40	30	70	40	3	1	1	10	107
ARCHER	15	5	90	20	6	5	7	7	52
ARBALÉTRIER	20	7	90	25	6	6	4	8	65
CAVALIER	25	10	60	20	7	1	2	8	58
GÉNÉRAL	80	20	80	50	7	1	2	10	156

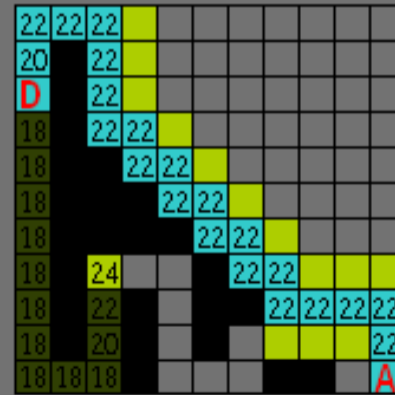
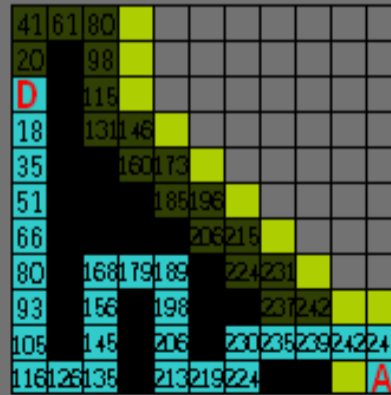
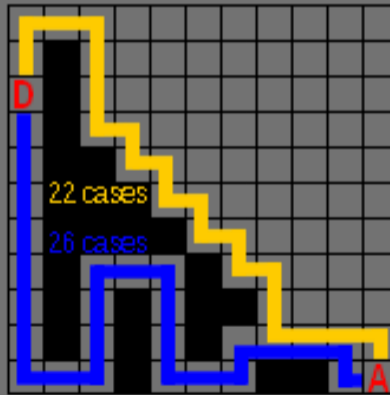
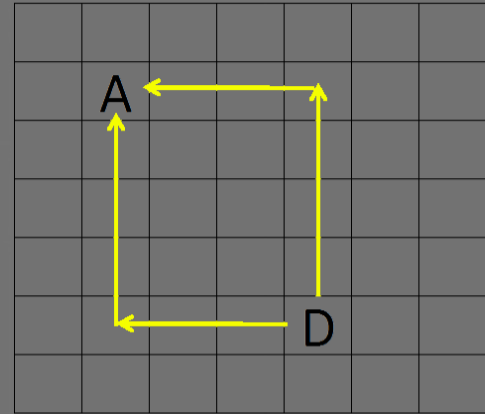
Le plateau de jeux

- Divisé en deux parties distinctes:
 - Plateau avec les unités
 - Plateau avec les obstacles
- Méthode pour générer une grille d'obstacles avec les deux plateaux



L'algorithme A*

- Moyen simple de trouver un des chemins les plus courts
- Mais très consommateur de ressources



■ En cours d'analyse ■ Déjà visité ■ Chemin renvoyé

Intelligence Artificielle

- IA simple:
 - Déplacement aléatoire
 - Attaque le premier ennemi qu'elle croise
- IA médium:
 - Se déplace vers l'unité la plus proche
 - Attaque cette même unité si possible
- IA difficile:
 - à venir

code de l'IA simple pour placer ses unités

```
def placerUnites(plateau, joueur, listeDesArmees):  
    cases=[]  
    if joueur == 0 :  
        for j in range(0,4):  
            for i in range(len(plateau.troupes[j])):  
                if plateau.troupes[j][i]==1 and plateau.terrain[j][i]==1:  
                    cases+=[(i,j)]  
        for j in listeDesArmees[joueur].compo:  
            k=randint(0,len(cases)-1)  
            listeDesArmees[joueur].compo[j].coord=cases[k]  
            plateau.troupes[cases[k][1]][cases[k][0]]=listeDesArmees[joueur].compo[j]  
            del cases[k]  
    else :  
        for j in range(16,20):  
            for i in range(len(plateau.troupes[j])):  
                if plateau.troupes[j][i]==1 and plateau.terrain[j][i]==1:  
                    cases+=[(i,j)]  
        for j in range(len(listeDesArmees[joueur].compo)):  
            k=randint(0,len(cases)-1)  
            listeDesArmees[joueur].compo[j].coord=cases[k]  
            plateau.troupes[cases[k][1]][cases[k][0]]=listeDesArmees[joueur].compo[j]  
            del cases[k]
```

L'interface graphique

- Réalisée pour l'utilisateur final
- Plus simple d'utilisation
- Aide extérieure pour la réalisation des images



QUITTER

JOUEUR 1:

HUMAIN

IA
SIMPLE

IA
MEDIUM

IA
DIFFICILE

CHOISISSEZ UNE RACE :

QUITTER

HUMAINS

NAINS

DEMONS

ORKS

ELFES

CHEVALIER --PRIX: 107



TOTAL: 0

QUITTER

ENROLE --PRIX: 24



TOTAL: 0

ARCHER --PRIX: 52



TOTAL: 0

SOLDE RESTANT: 1000

CAVALIER --PRIX: 58



TOTAL: 0

EPEISTE --PRIX: 56



TOTAL: 0

ARBALETRIER --PRIX: 65



TOTAL: 0

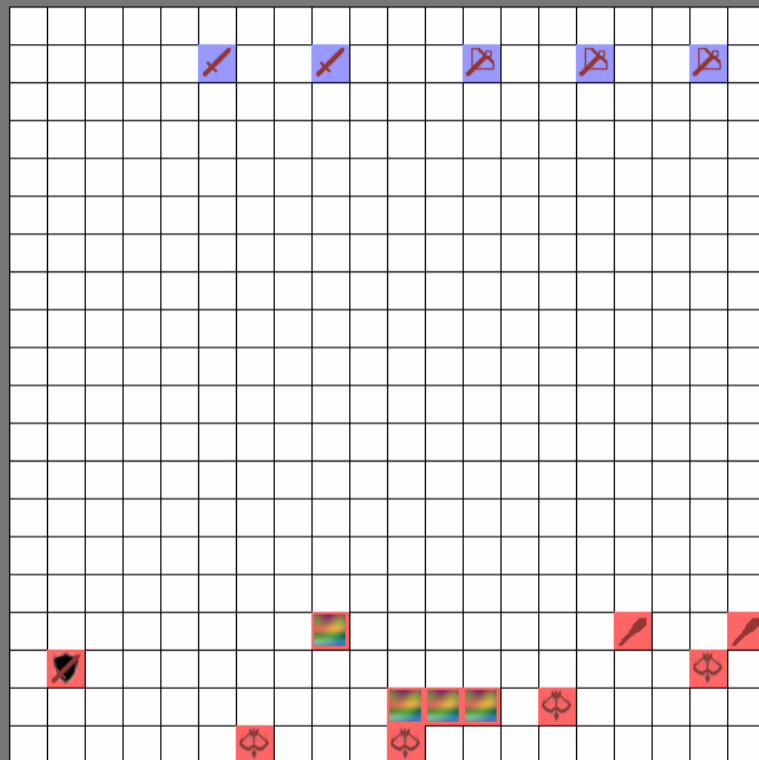
LANCIER --PRIX: 56



TOTAL: 0

STOP

APPUYEZ SUR STOP QUAND VOUS AUREZ TERMINER DE CHOISIRE VOS UNITÉS



NOM: ARBALETRIER

PV: 20

ARMURE: 7

ATTAQUE: 25

DEPLACEMENT RESTANT: 6

PORTÉE DE L'ATTAQUE: 6

L'Optimisateur

- Permet de faire combattre différentes armées dans un tournoi
- Récupère les données du combat
- Permet de connaître les forces et faiblesses d'une armée
- Après étude de plusieurs combats, modifie la composition d'une armée et recommence



Les tournois

- Fait combattre différentes armées selon un ordre prédéfini
- Fixe l'aléatoire via une *seed*
- Enregistre le combat dans un fichier texte

```
def tournoi(fichierArmees):  
  
    with open("armeeia/"+fichierArmees+".save", 'rb') as fichier :  
        mon_depickler=pickle.Unpickler(fichier)  
        listArmee=mon_depickler.load()  
  
    victoires=[]  
    graine=0  
    sauvegarde=fichierArmees+"\n"  
  
    for i in range(len(listArmee)):  
        victoires[i]=0  
  
    for i in range(len(listArmee)):  
        for k in range(len(listArmee)):  
            if i != k:  
                victoires[i]=0  
                victoires[k]=0  
                for j in range(5):  
                    seed(graine)  
                    contenuArmees=[deepcopy(listArmee[i]),deepcopy(listArmee[k])]  
                    p=Plateau(20,20)  
                    Tours=0  
                    bebeBeel.placerUnites(p,0,contenuArmees)  
                    bebeBeel.placerUnites(p,1,contenuArmees)  
  
                    while contenuArmees[0].compo != {} and contenuArmees[1].compo != {} :  
  
                        if Tours%2==0:  
                            bebeBeel.deplacerUnites(p,1,0,contenuArmees)  
                            bebeBeel.attaqueAuto(p,0,1,contenuArmees)  
                            Tours+=1  
                        else :  
                            bebeBeel.deplacerUnites(p,0,1,contenuArmees)  
                            bebeBeel.attaqueAuto(p,1,0,contenuArmees)  
                            Tours+=1  
                        if contenuArmees[0].compo == {}:  
                            victoires[k]+=1  
                        else :  
                            victoires[i]+=1  
                    sauvegarde+=str(i)+" "+str(k)+" "+str(graine)+"\n"  
                    graine+=1  
                if victoires[k]>victoires[i]:  
                    victoires[k]+=1  
                else :  
                    victoires[i]+=1  
  
    fichiersauvegarde=open("Donnees_tournois/"+fichierArmees+".txt", 'w')  
    fichiersauvegarde.write(sauvegarde)  
    fichiersauvegarde.close()
```

La récupération de données

- Se fait pendant les combats
- Stocke au format .JSON
- Affiche sur une page web pour une meilleure visualisation
- Création d'un programme permettant la reconstitution d'une bataille

```
var donnee = {
  "tournoi1":{
    "combat1":{
      "armee1":{
        "Composition":["archer1","archer2","cavalier1"],
        "race":"humain"
      },
      "armee2":{
        "composition":["gobelin1","gobelin2","archer1"],
        "race":"ork"
      },
      "1":{
        "seed":1,
        "armee1":{
          "vainqueur":1,
          "tour1":{
            "bonus":0,
            "archer1":{
              "cible":"gobelin1",
              "attaque":1,
              "dégâts_Infligés":20,
              "attaquant":["gobelin2","archer1"],
              "esquive":[0,1],
              "dégâts_Reçu":[5,0],
              "contre-attaque":[0,1],
              "dégâts_ContreAttaque":[0,10],
              "PVrestant":10
            }
          }
        }
      }
    }
  }
}
```

Traitement des données

- Lecture du fichier .JSON
- étude des résultats:
 - unités en vie
 - unités les plus/moins efficaces
 - taux de victoire avec la composition actuelle
- Modification de la composition de l'armée jusqu'à un taux de victoire acceptable

END

Conclusion:

- Le jeu est jouable selon les règles de base
- Les IA sont en cours de développement
- Début de l'optimisateur



groupe 2 Faquin

Optimisateur de Wargame

