

BABANAZAROVA Dilyara 28709428
CELIK Simay 28713301
KUDRYAVTSEVA Kristina 21137133

“ZOMBIES VS HUMANS : A CLASSIC WAR ”

Comme sujet de notre projet, on a choisi la guerre entre les humains et les zombies et on a choisi l'environnement du [Pygame](#) du langage Python pour facilement identifier les actions des agents.

Le projet représente un jeu effectué en deux équipes des agents. Le jeu se termine quand un des deux équipes a gagné, c'est-a-dire quand il ne reste que des agents de cette équipe dans l'environnement.

Le jeu se compose d'un environnement dynamique généré aléatoirement, des interactions entre les agents et les objets situés dans l'environnement, bien que les interactions parmi les agents. Dans notre projet on a deux types d'agents et deux types d'objets : dynamiques et statiques. Le projet est basé sur le modèle prédateur-proie, avec les comportements de fuite et de poursuite, du combat entre les agents de types différentes, l'épidémie du virus zombie et la reproduction entre les humains.

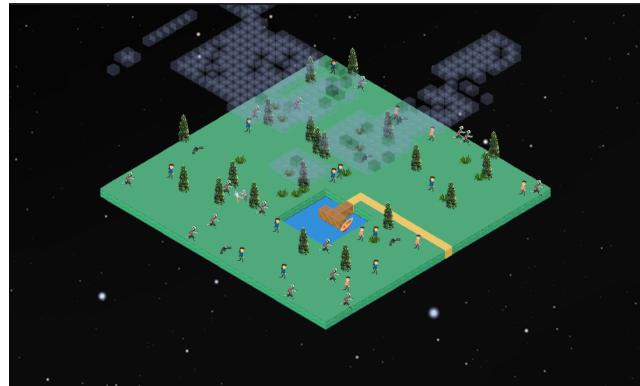
Les agents les plus importants du projet sont les agents dynamiques: les humains et les zombies. Le but des zombies est d'épandre leur population en transformant tous les humains en zombies. Le but des humains est d'éviter l'extinction des humains. Les humains peuvent se reproduire sexuellement, cependant, le seul moyen de reproduction pour les zombies est de transformer les humains en zombie comme le résultat des combats avec eux. Les deux types des agents sont en danger de la mort de faim ou bien de la mort d'âge.



L'ENVIRONNEMENT

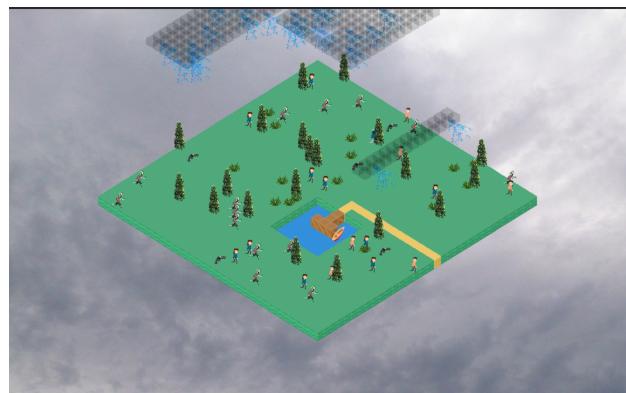
Les changements les plus importants dans l'environnement :

1. Périodes de la Journée



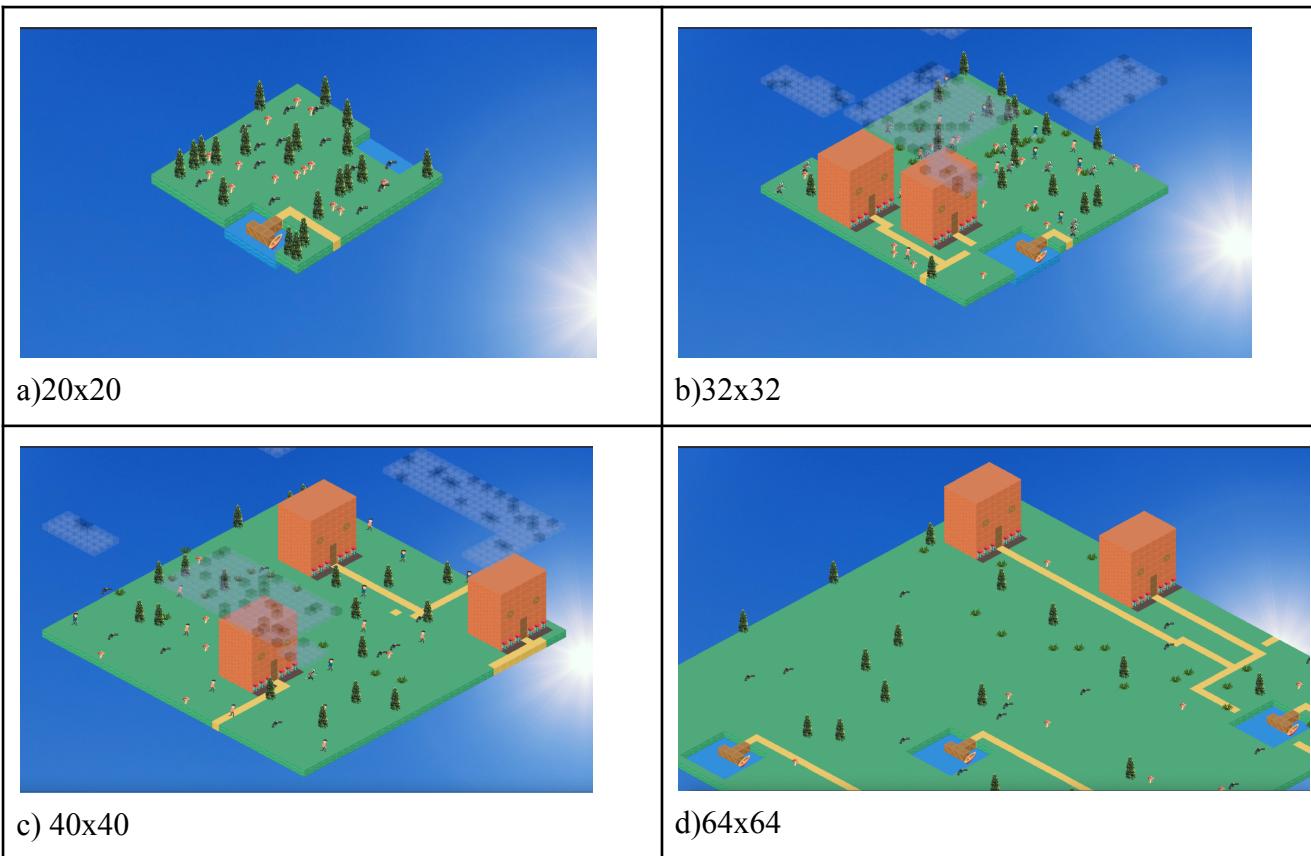
Dans notre monde, chaque 60 itérations on a le matin et le soir qui changent entre eux.

2. La Météo



On a la météo qui change aléatoirement chaque 120 itérations. Les changements de la météo dépendent d'une probabilité précisée dans le code du projet.

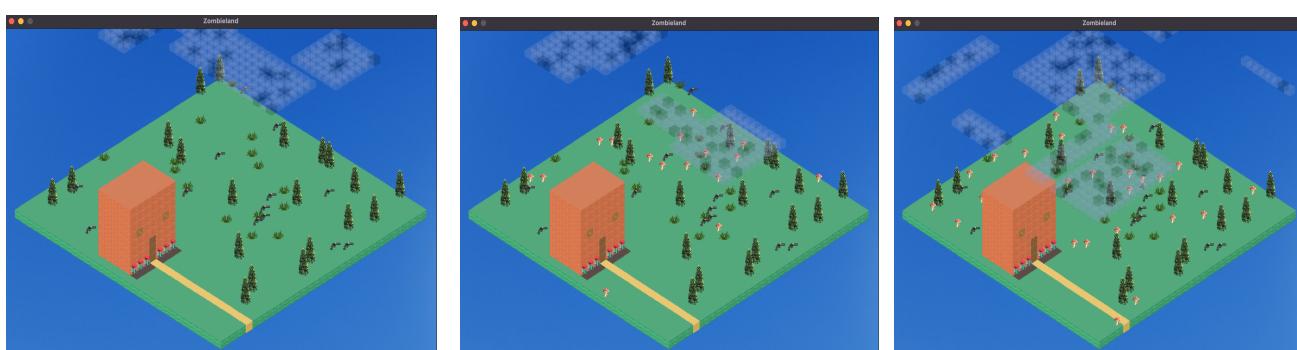
3. Génération Aléatoire de l'Environnement



Dans ces 4 exécutions différentes avec les différentes largeurs et hauteurs du monde, on peut voir que chaque fois notre monde change aléatoirement.

Chaque exécution, on initialise notre monde en vérifiant 2 variables : MAXENVOBJ et MAXSURFACE qui respectivement calculent en fonction de largeur et l'hauteur du monde, le nombre et la place maximum que les objets peuvent occuper. Après on met les structures (les maisons, le lac et les chemins) construits avec des objets en choisissant leurs (x,y) aléatoirement. On ne peut pas choisir (x,y) qui est déjà rempli afin d'éviter le problème de collision.

4. Les Objets Dynamiques

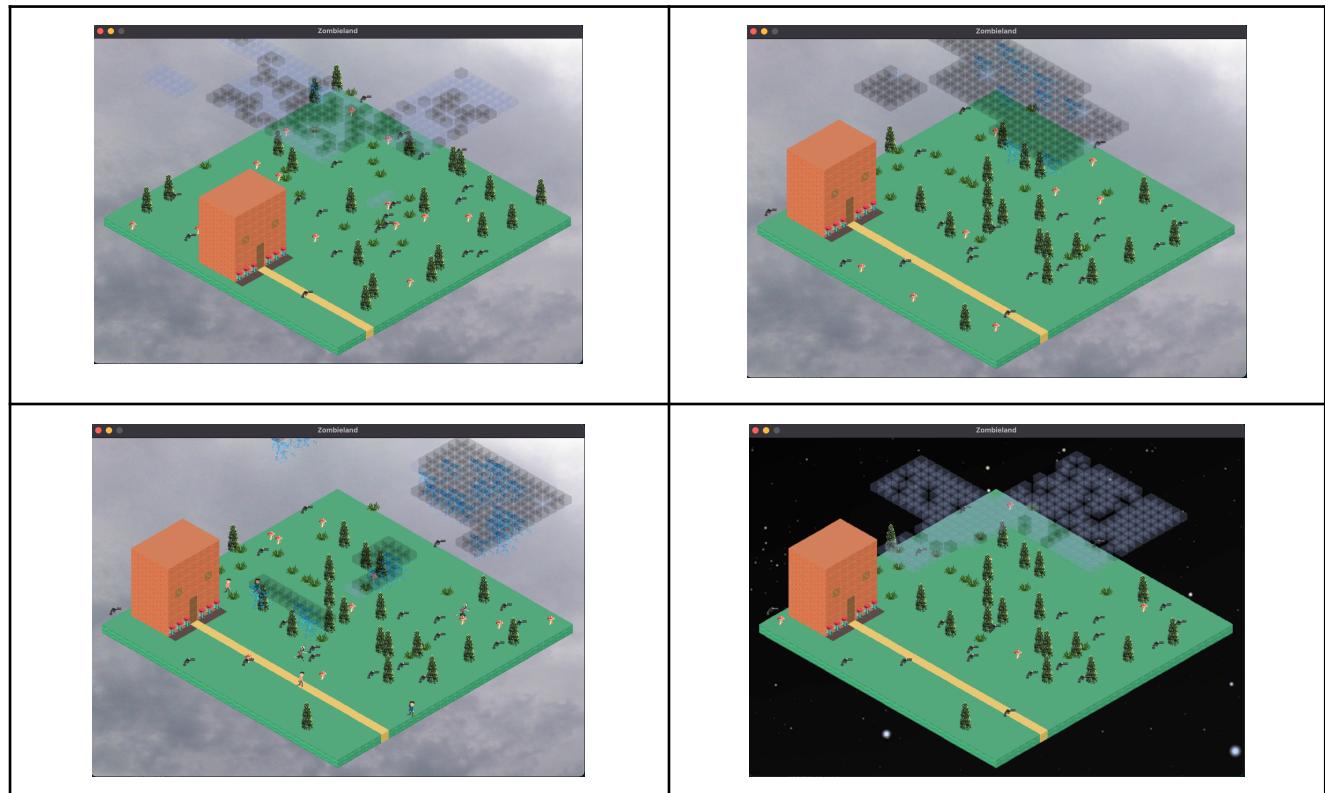


Chaque fois quand on initialise l'environnement, les nuages sont placés aléatoirement dans le niveau supérieur de l'environnement. Les nuages peuvent être chargés ou non chargés. On peut les distinguer avec leurs couleurs.

La dynamique entre les nuages correspond au modèle du feu de forêt vu dans le cours.

Les places des nuages changent à chaque itération du monde si la météo est ensoleillée. Si il est nuageuse, on regarde autour de nuages chargés et si il a un voisin chargé aussi, alors il y aura un orage. De plus, quand il est nuageuse, si un nuage a un voisin chargé il devient chargé aussi. De même manière et avec une probabilité tous les nuages deviennent chargés, mais quand la météo redevient ensoleillée, les nuages ne sont plus effectués par leurs voisins.

Cette interaction est représentée dans les images suivantes :



LES CRÉATURES

1. Les Agents et Les Agents

1. Les Humains VS Les Zombies

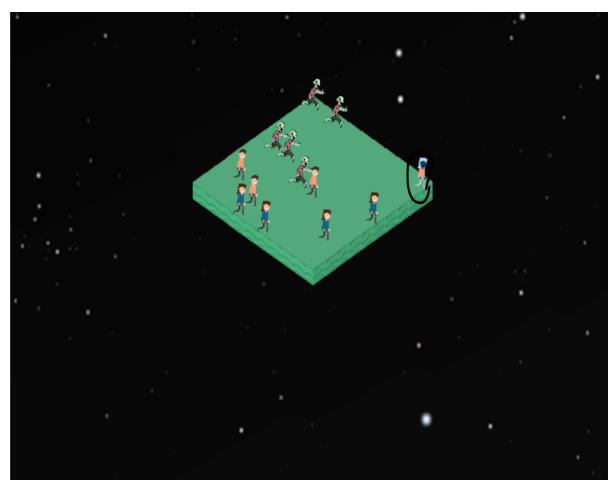
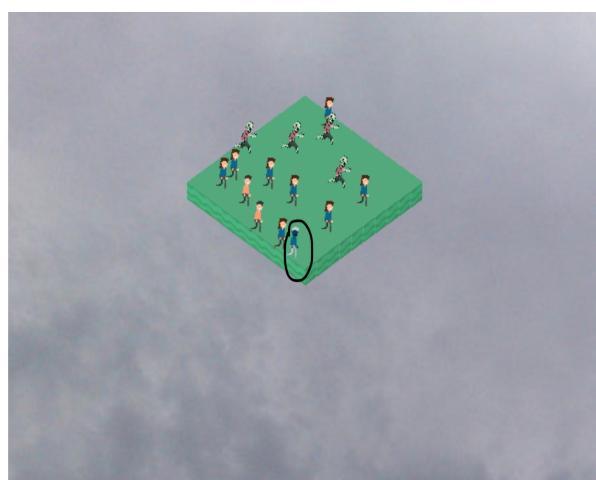


Les zombies infectent les humains en les mordant, cependant les humains se protègent en tirant les zombies.

Si le zombie et l'humain sont dans la même case, d'abord humain essaie de tuer le zombie et cela dépend d'un probabilité et du nombre de balles dans son pistolet. S'il réussit, on enlève le zombie et sinon cet humain devient infecté par le mordant du zombie. Il devient un zombie lors de la terminaison de la période d'infection s'il n'est toujours pas guéri. Cette interaction entre les zombies et les humains est représentée par l'état de combat entre les deux.

Dans l'image au-dessus, on peut voir un exemple du combat entre un zombie et un humain où l'humain a gagné.

Dans les images suivantes, on voit respectivement un homme infecté et une femme infectée.

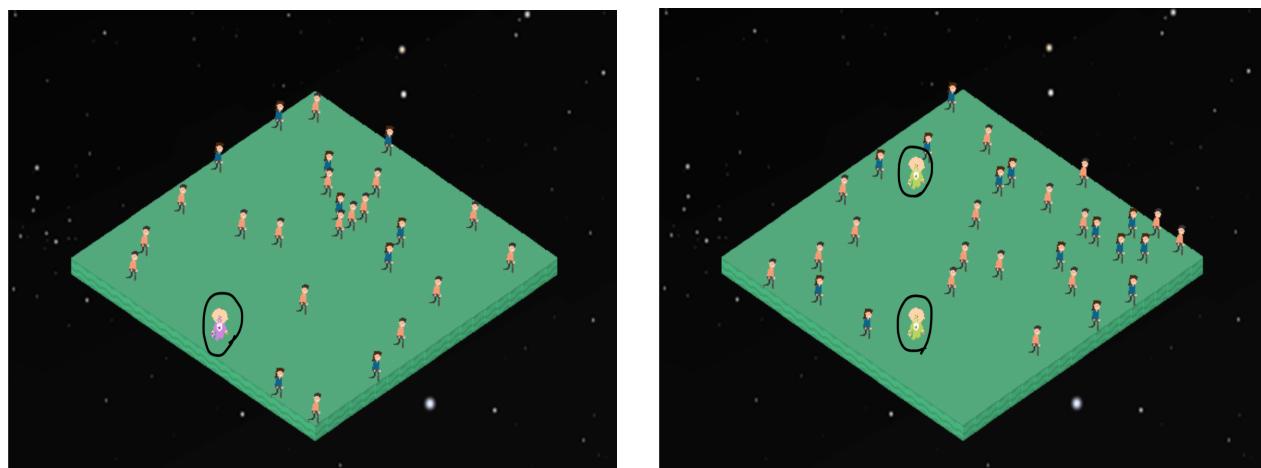


2. Le Comportement Entre Les Sexes Opposés

Pour les humains, la deuxième priorité après le comportement de fuite est la reproduction sexuelle. Pour cela, le comportement de poursuite des partenaires sexuels est intégré. Si un homme et une femme sont dans la même case comme le résultat de cet comportement, un bébé, aléatoirement le garçon ou la fille, se reproduira.

S'il n'y a pas de zombie dans la proximité, avec une probabilité, un homme va bouger vers une femme si elle est dans ses voisins. Sinon il va se déplacer aléatoirement.

Si c'est une femme, avec une probabilité, elle reste dans la même case si son voisin est un homme et sinon elle va se déplacer aléatoirement.



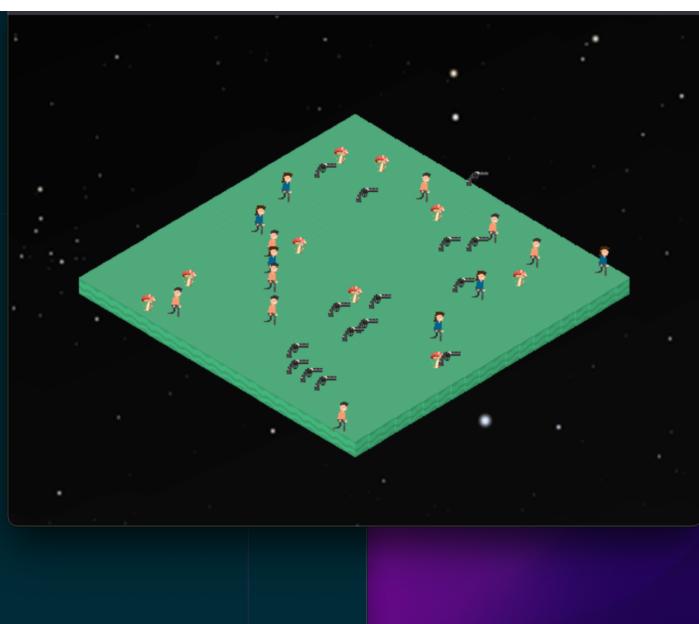
2. Les Agents et Les Objets

1. Le Ramassage de Nourriture et des Armes

```
1462 |     shuffle(zombies)
1463 |     shuffle(humans)
1464 |     Food.randomDrop(it, foods)
1465 |     Food.decomposition(foods)
1466 |     Gun.randomDrop(it, guns)
1467 |
1468 |     for objList in [foods, guns, cure]:
1469 |         for i in objList:
1470 |             setAgentAt(i.x, i.y, i.type)
1471 |     for z in zombies:
1472 |         if z.type!=zombieId:
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

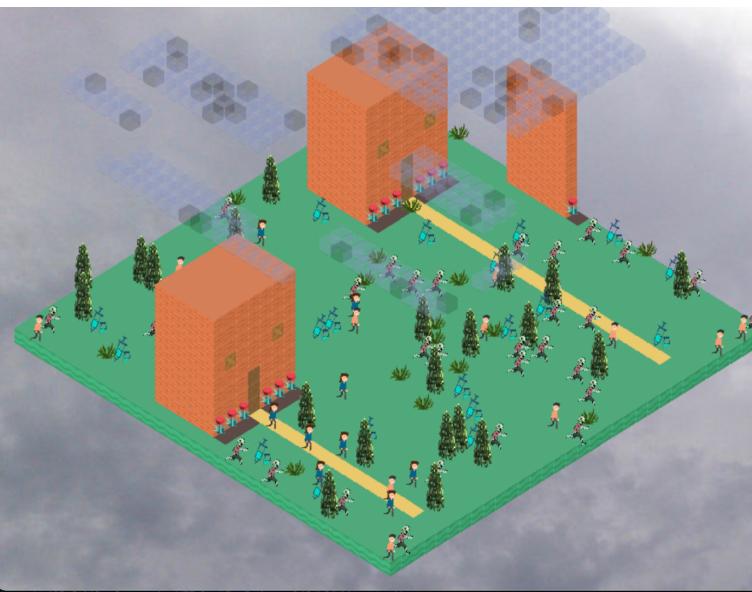
```
[fps] 29.714601805020596
human ate
a baby is born
[fps] 27.440458553484962
human ate
a baby is born
a baby is born
armed
armed
armed
[fps] 29.60690958525872
human ate
armed
a baby is born
human ate
[fps] 29.689384761473985
a baby is born
human ate
[fps] 29.473920913864248
armed
human ate
[fps] 29.6128711904521
a baby is born
armed
human ate
[]
```



Les conditions pour la mort des agents se composent de la mort de l'âge et la mort de faim et de la transformation en zombie. Dans le but d'éviter leurs morts, des agents humains interagissent avec l'environnement en ramassant des objets statiques de deux sortes - d'objets alimentaires (foods) et des armes (guns). Pour que les humain évitent la mort de faim, les humain peuvent recueillir la nourriture placée au hasard dans l'environnement. La nourriture donne des points d'énergie aux humains. Les points d'énergie sauvent les humains de la faim et ralentissent le passage de l'âge maximum. Les armes donnent aux agents la possibilité de repousser les attaques des zombies en les tuant, pour que les humain ne soient pas converties en zombie eux-mêmes.

On peut vérifier ces changements avec l'affichage dans le terminal et on peut les voir pendant l'exécution, comme montré dans l'image au-dessus.

2. La guérison



```

136 # set surface of displayed tiles (ie. nb of
137 worldWidth = 32 #32 #after 64 it lags
138 viewWidth = 32#32
139 viewHeight = 32#32
140
141 scaleMultiplier = 0.25 # re-scaling of loaded
142
143 objectMapLevels = 15 # number of levels for
can pile upon one another.
144
145 # set scope of displayed tiles
146 xViewOffset = 0
147 yViewOffset = 0
148
149
150 addNoise = False
151
152 maxFps = 30 # set up maximum number of frame

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

Human 4383704064 was infected and had gun : 0
infected F
[fps] 23.029482959143255
Human 4383276996 was infected and had gun : 0
infected F
Human 4383276864 was infected and had gun : 0
infected M
Human 4383276288 was infected and had gun : 0
infected F
Human turned into a zombie
Human turned into a zombie
Human turned into a zombie
Human 4383276672 was infected and had gun : 0
infected F
Human 4383276000 was infected and had gun : 0
infected M
[fps] 23.15370072439223
[ ]

```

Ln 135, Col 16 Spaces: 4 UTF-8 LF Python 3.10.4 64-bit

Les humains infectés par les zombies peuvent guérir en prenant une cure anti-zombie. Si la cure se trouve dans les voisins d'un humain infecté, il va marcher vers elle et quand ils sont dans la même case, il peut la prendre. Si l'humain infecté ne trouve pas de cure pendant sa période d'infection, il devient un zombie.

Une quantité limitée des objets de guérison est allouée aléatoirement dans l'environnement.

3. La nourriture

The screenshot displays a 3D isometric simulation of a zombie-infested town. The environment is set on a green hillside with several orange brick houses featuring red roofs and flower boxes. The terrain is dotted with green trees and bushes. A large number of zombies are scattered throughout the scene, some walking and others standing. A yellow path or road leads from the foreground towards one of the houses. In the bottom left corner, there is a terminal window showing the game's source code and log output.

```
#zombies.append(Zombie(winnerzombiefield.Ty, Tx))
# print("Z remove")
return

# def reproduce(self, list_humans
def reproduce(self, list_humans
    options=[ImageIdF, ImageIdM]
    sex_choice = choice(options)
    for h in list_humans:
        if met(self, h):
            if self.getType() != h.getType():
                if random() < PROB:
                    coords = self.getCoord()
                    #print("a b")
                    if sex_choice == ImageIdF:
                        list_humans.append(Human(coords[0], coords[1], sex_choice))
                    else:
                        list_humans.append(Woman(coords[0], coords[1], sex_choice))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
```

```
food decomposition
food decomposition
food drops
food decomposition
food decomposition
food decomposition
food drops
Human 443660560 was infected and had gun : 0
infected F
food drops
food drops
food drops
[fps] 20.396012844180262
food drops
Human 4436657632 was infected and had gun : 0
infected F
human ate
```

Chaque fois quand il y aura DROPDAYFOOD itérations et avec une probabilité, la nourriture va tomber aléatoirement dans le monde. Et il y a une période de décomposition DECOMPDAYFOOD pour les nourritures aussi, si pendant cette période les humains ne vont pas manger, la nourriture disparaîtra.