

ML par Renforcement

Groupe 1- Nikmabe :
N.G. - K.P. - M.F. - B.B.

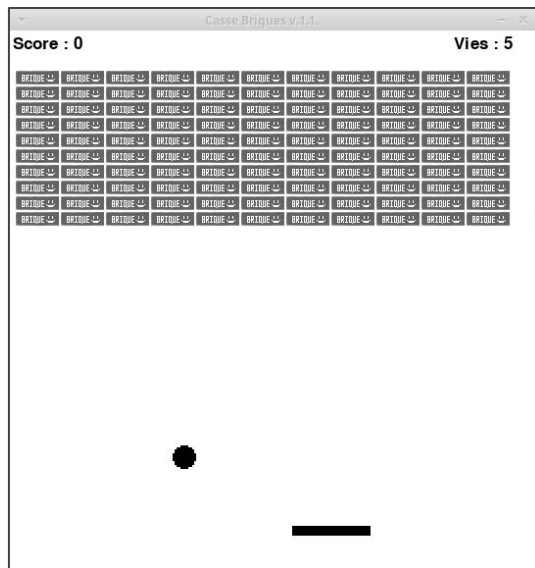




Face Recognition

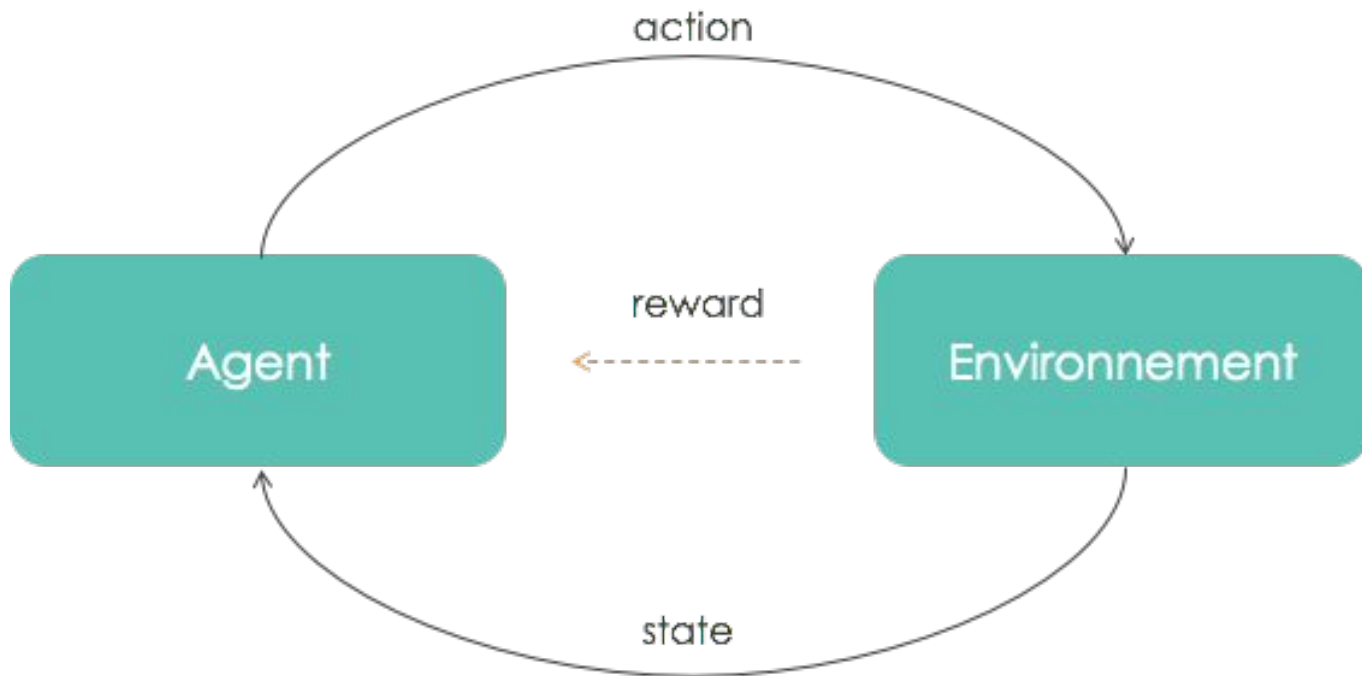
Optical Character Recognition

GAME-AI-TRAINER

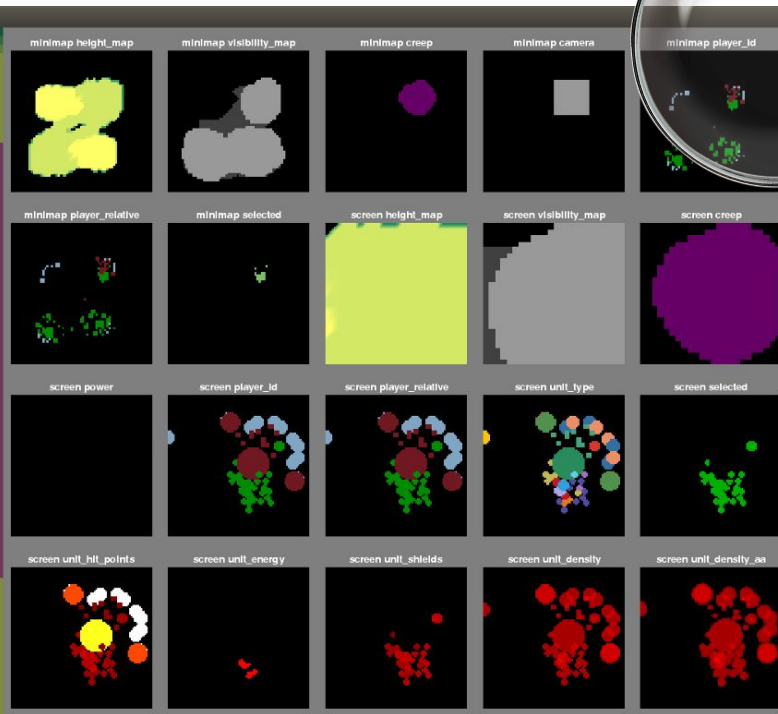
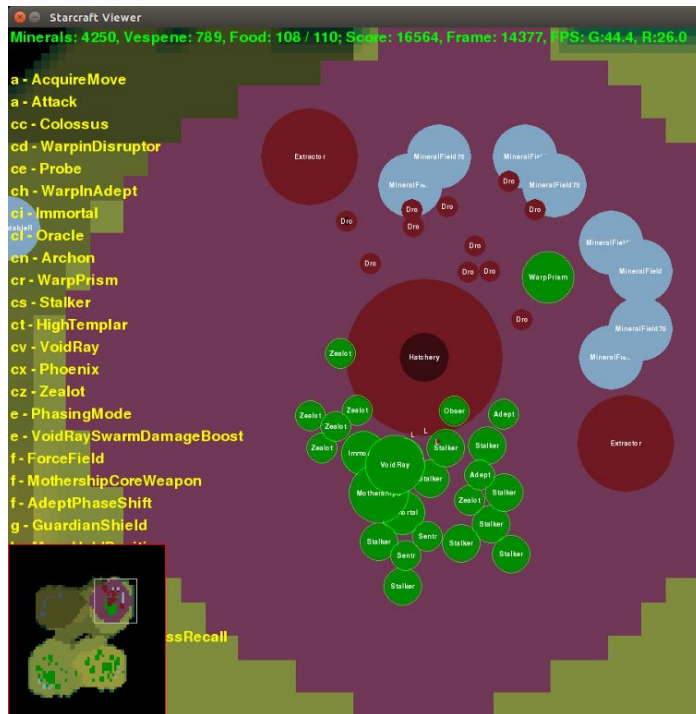




DÉFINITION : APPRENTISSAGE PAR RENFORCEMENT

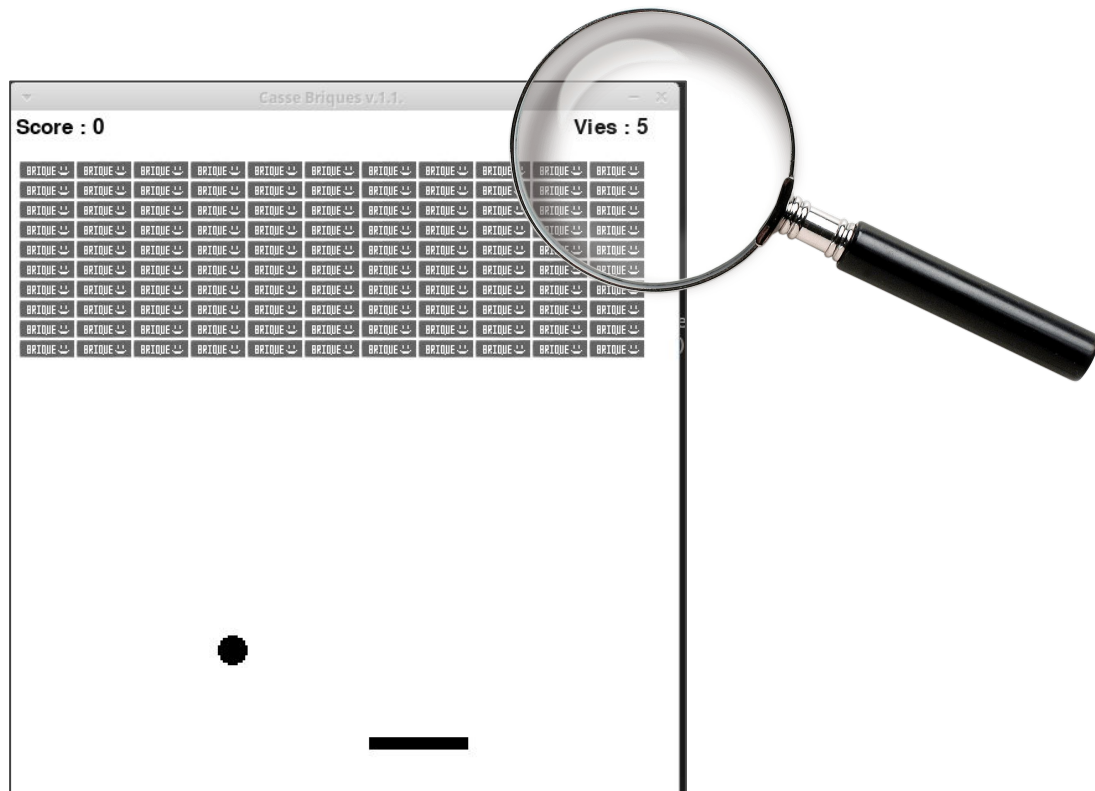


COMMENT CRÉER UN CADRE PROPICE À L'ApR



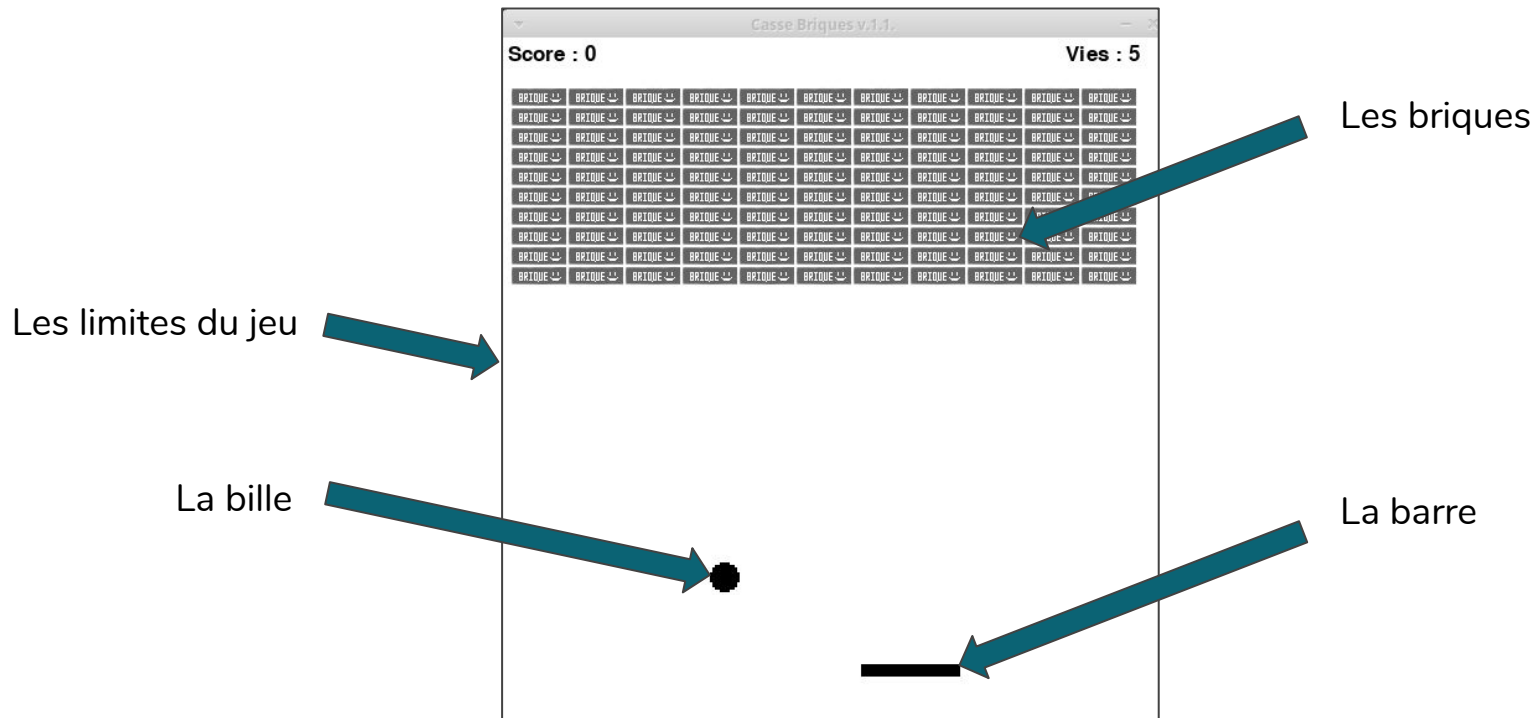


CADRE IA_GAME_TRAINER



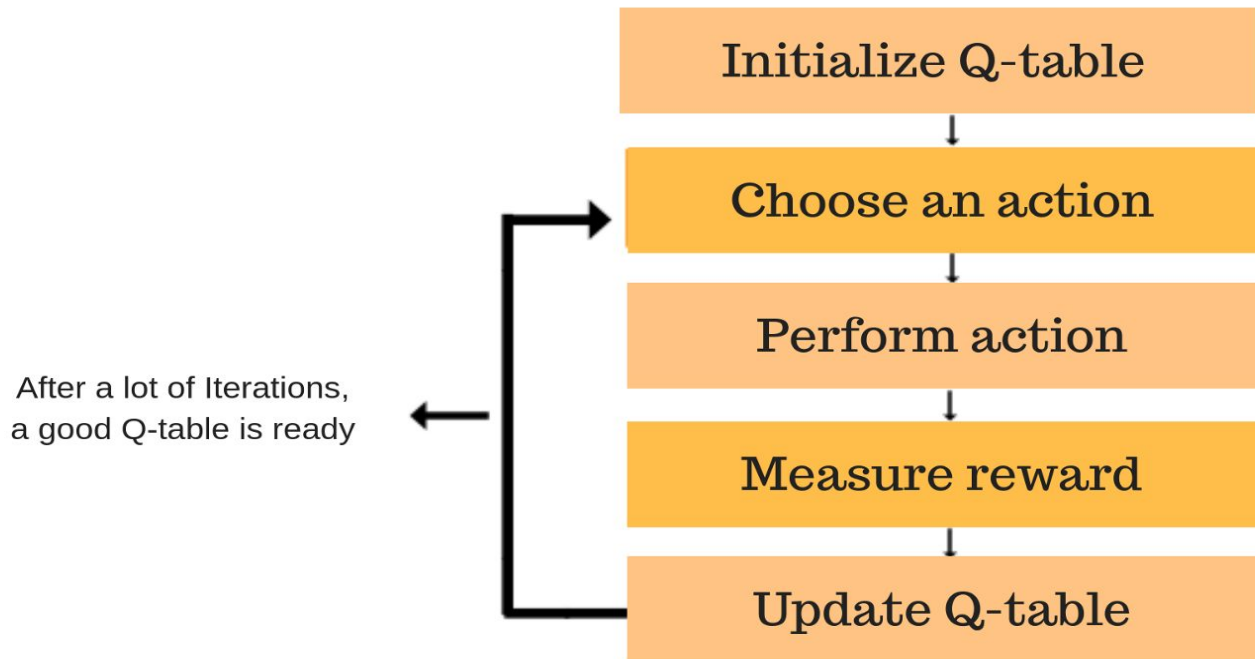


CADRE IA_GAME_TRAINER










Q-LEARNING





GÉNÉRER UNE PRÉDICTION (ApR) :

1 - Générer des données

							
							
							
					END		

Actions:








Start	0	0	0	0
Nothing/Blank	0	0	0	0
Bomb	0	0	0	0
Power-up	0	0	0	0
End	0	0	0	0



GÉNÉRER UNE PRÉDICTION (ApR) :

1 - Générer des données

							
							
							
					END		

Actions:



Start	0	0	0	0
Nothing/Blank	0	0	0	0
Bomb	0	0	0	0
Power-up	0	0	0	0
End	0	0	0	0



GÉNÉRER UNE PRÉDICTION (ApR) :

2 - Jouer des parties

Actions:    

Start	0,7	0,5	1,2	0,6
Nothing/Blank	0,8	1,5	1,4	0,4
Bomb	0,3	0,7	1,8	0,4
Power-up	0,8	0,1	1,3	1,1
End	0,7	0,5	0,9	1,4

- Remplie au fil de l'entraînement
- Prends son nom de la fonction $Q(s,a)$

$$Q(s,a) = Q(s,a) + \alpha (R(s,a) + \gamma \max(Q'(s',a')) - Q(s',a'))$$



GÉNÉRER UNE PRÉDICTION (ApR) :

3 - Jouer des parties

Actions:    

Start	0,7	0,5	1,2	0,6
Nothing/Blank	0,8	1,5	1,4	0,4
Bomb	0,3	0,7	1,8	0,4
Power-up	0,8	0,1	1,3	1,1
End	0,7	0,5	0,9	1,4

- Réseau d'apprentissage utilisé pour le test
- Prédiction en fonction de l'environnement

Application au casse brique



Premier instinct:

- Modèle simpliste a trois états

Pas assez fin!

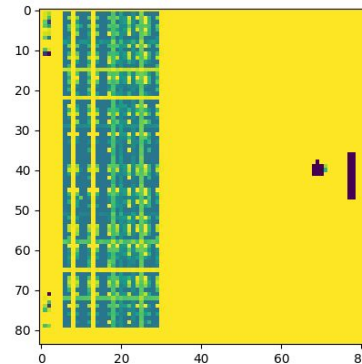
Application au casse brique



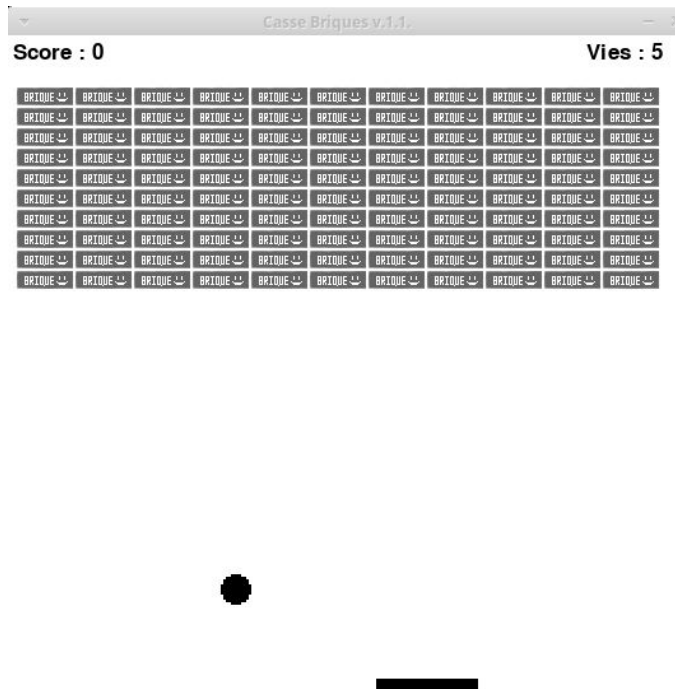
Seconde hypothèse

- Utilisation d'images du jeu

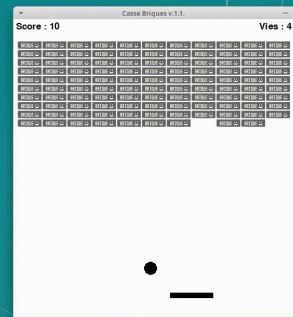
Plus lourd mais plus efficace!



Application au casse brique

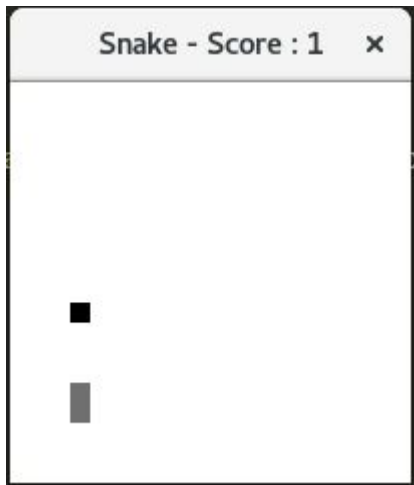


```
python q_learning.py homemade train 104x56  
  
trainable params: 676,915  
non-trainable params: 0  
  
2019-05-03 09:22:01.780038: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA  
2019-05-03 09:22:01.802016: I tensorflow/core/platform/profile_utils/cpu_utils.cc:94] CPU Frequency: 2000000000 Hz  
2019-05-03 09:22:01.802057: I tensorflow/compiler/xla/service/service.cc:150] XLA service 0x34620600 executing computations on platform Host. Devices:  
2019-05-03 09:22:01.802078: I tensorflow/compiler/xla/service/service.cc:150] StreamExecutor device (0):   
y: nondimname, nondimname  
/home/killy/em/game/ai/trainer/lib/python3.6/site-packages/skinage/transforms/warps.py:110: UserWarning  
g: Anti-aliasing will be enabled by default in skinage 0.15 to >  
warn:Anti-aliasing will be enabled by default in skinage 0.15 to >  
state: observe, episode: 0, score: 0, global_step: 640, avg loss: 0.0, step: 640, memory length: 640  
state: observe, episode: 1, score: 0, global_step: 980, avg loss: 0.0, step: 244, memory length: 884  
state: observe, episode: 2, score: 0, global_step: 980, avg loss: 0.0, step: 98, memory length: 980  
state: observe, episode: 3, score: 10, global_step: 1005, avg loss: 0.0, step: 115, memory length: 1005  
state: observe, episode: 4, score: 0, global_step: 1179, avg loss: 0.0, step: 84, memory length: 1179  
state: observe, episode: 5, score: 10, global_step: 1375, avg loss: 0.0, step: 150, memory length: 1375  
state: observe, episode: 6, score: 20, global_step: 1495, avg loss: 0.0, step: 129, memory length: 1495  
state: observe, episode: 7, score: 60, global_step: 2222, avg loss: 0.0, step: 727, memory length: 2222  
state: observe, episode: 8, score: 20, global_step: 2591, avg loss: 0.0, step: 369, memory length: 2591  
state: observe, episode: 9, score: 60, global_step: 3028, avg loss: 0.0, step: 437, memory length: 3028  
state: observe, episode: 10, score: 0, global_step: 3102, avg loss: 0.0, step: 154, memory length: 3102  
state: observe, episode: 11, score: 20, global_step: 3350, avg loss: 0.0, step: 160, memory length: 3350  
state: observe, episode: 12, score: 0, global_step: 3457, avg loss: 0.0, step: 107, memory length: 3457  
state: observe, episode: 13, score: 20, global_step: 3646, avg loss: 0.0, step: 393, memory length: 3646  
state: observe, episode: 14, score: 100, global_step: 4652, avg loss: 0.0, step: 804, memory length: 4652  
state: observe, episode: 15, score: 40, global_step: 4846, avg loss: 0.0, step: 194, memory length: 4846  
state: observe, episode: 16, score: 0, global_step: 5074, avg loss: 0.0, step: 228, memory length: 5074  
state: observe, episode: 17, score: 60, global_step: 5406, avg loss: 0.0, step: 332, memory length: 5406  
state: observe, episode: 18, score: 10, global_step: 5595, avg loss: 0.0, step: 160, memory length: 5595  
state: observe, episode: 19, score: 0, global_step: 5825, avg loss: 0.0, step: 230, memory length: 5825  
state: observe, episode: 20, score: 20, global_step: 6010, avg loss: 0.0, step: 120, memory length: 6010  
state: observe, episode: 21, score: 50, global_step: 6746, avg loss: 0.0, step: 730, memory length: 6746  
state: observe, episode: 22, score: 60, global_step: 7230, avg loss: 0.0, step: 476, memory length: 7230  
state: observe, episode: 23, score: 40, global_step: 7500, avg loss: 0.0, step: 370, memory length: 7500  
state: observe, episode: 24, score: 60, global_step: 8129, avg loss: 0.0, step: 531, memory length: 8129  
state: observe, episode: 25, score: 40, global_step: 8240, avg loss: 0.0, step: 301, memory length: 8240  
state: observe, episode: 26, score: 20, global_step: 8586, avg loss: 0.0, step: 156, memory length: 8586  
state: observe, episode: 27, score: 30, global_step: 8756, avg loss: 0.0, step: 170, memory length: 8756  
state: observe, episode: 28, score: 40, global_step: 9236, avg loss: 0.0, step: 480, memory length: 9236  
state: observe, episode: 29, score: 30, global_step: 9604, avg loss: 0.0, step: 360, memory length: 9604  
state: observe, episode: 30, score: 20, global_step: 9827, avg loss: 0.0, step: 220, memory length: 9827  
state: explore, episode: 31, score: 100, global_step: 10050, avg loss: 0.02569771785079478, step: 718, memory length: 10050  
state: explore, episode: 32, score: 40, global_step: 10935, avg loss: 0.036790316962053660, step: 385, memory length: 10935  
state: explore, episode: 33, score: 0, global_step: 11070, avg loss: 0.036215892114243471, step: 135, memory length: 11070  
state: explore, episode: 34, score: 10, global_step: 11276, avg loss: 0.03919099455091974, step: 206, memory length: 11276
```





Première hypothèse:



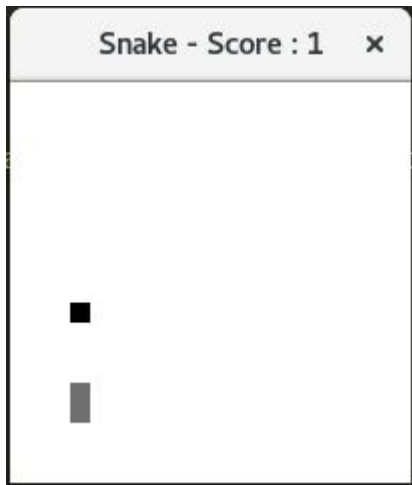
- Etat par grille de jeu
 - Position de la tête
 - Position de la nourriture
- Résultats obtenus
 - Peu concluant

[illegible]



Application au Snake

Seconde hypothèse:



- Etat par “Radar”
 - Présence d’un danger proche
 - Direction de la nourriture
- Résultats obtenus
 - Résultats non satisfaisants
 - Revoir le modèle Keras

Bilan





Merci de votre attention :)

L'équipe Nikmabe.