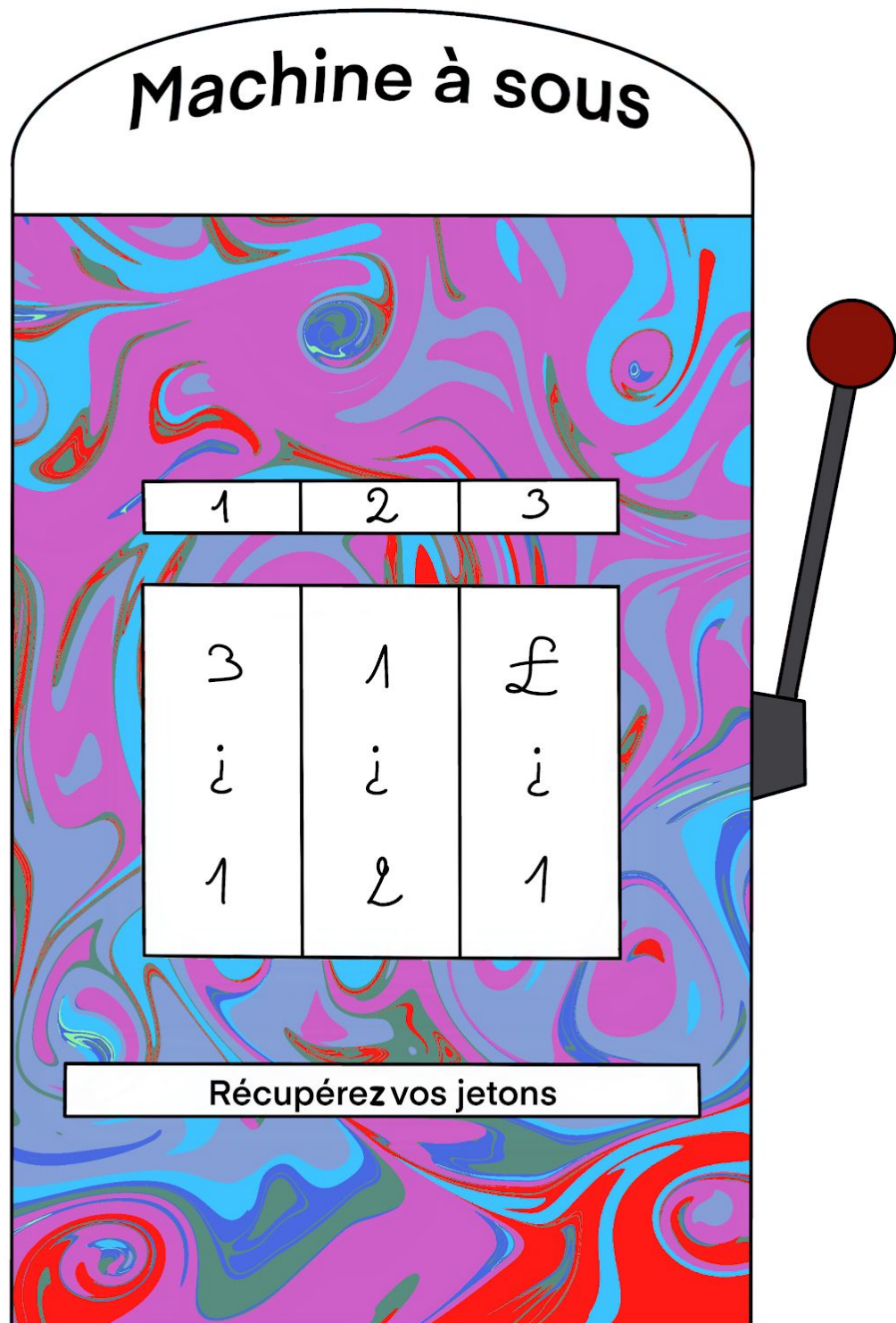


Machine à sous virtuelle



Ce programme a été créé dans l'optique de reproduire le fonctionnement d'une machine à sous virtuellement. Au cours de son déroulement l'utilisateur pourra interagir plusieurs fois avec le code et ainsi influencer grandement le déroulement de la partie. Cette machine étant virtuelle n'a pas vraiment d'attache matérielle mais bénéficie tout de même d'équivalents virtuels pour ces rouleaux, écran et compteur. Le programme propose même d'échanger les jetons restants contre des "gains" matériels, ce qui n'est bien sûr pas réellement réalisable dans ce contexte.

Une machine à sous est constituée de trois rouleaux possédants différents symboles, certains représentés plus que d'autres. Ici les trois rouleaux sont des tableaux constitués de 10 arguments :

- ['£',2,3,'\$',1,3,'¿','¥',3,2]
- [3,2,'¥',1,'¿',2,2,'\$',1,'£']
- [2,'£','¿',1,'\$',3,1,1,3,'¥']

Ici l'utilisateur va se voir afficher les rouleaux complets et leurs numéros ainsi que les règles du jeu pour pouvoir penser à ce qu'il va miser et comment il va jouer :

- `print("\033[44m'+\"1\", \"2\", \"3\"+\033[49m')`
- `print("\033[47m')`
- `print((tab1[0]),(tab2[0]),(tab3[0]))`
- `print("\033[34m', \"\033[4m\"")`
- `print((tab1[1]),(tab2[1]),(tab3[1]))`
- `print("\033[0m', \033[47m')`
- `print((tab1[2]),(tab2[2]),(tab3[2]))`
- `print("\033[48;2m')`
- *#affiche le tableau 3x3 qui sera affiché et la ligne bleu qui sera lue*
- `print((tab1[3]),(tab2[3]),(tab3[3]))`
- `print((tab1[4]),(tab2[4]),(tab3[4]))`
- `print((tab1[5]),(tab2[5]),(tab3[5]))`
- `print((tab1[6]),(tab2[6]),(tab3[6]))`
- `print((tab1[7]),(tab2[7]),(tab3[7]))`
- `print((tab1[8]),(tab2[8]),(tab3[8]))`
- `print((tab1[9]),(tab2[9]),(tab3[9]))`
- `print("\033[44m'+\"Vos gains de jetons dépendent de la combinaison obtenue en ligne bleue(2ème ligne).\n\"")`
- `print(\"Vous gagnez : \n_3 jetons pour 3'£', \n_4 jetons pour 3'$', \n_5 jetons pour 3'¥', \n_6 jetons pour 3 '¿', \n_3 pour 3 autres symboles identiques \n_1 jeton pour 2 symboles identiques \n Vous perdez 1 jetons si aucun symbole n'est identique.\"+\033[49m')`

Lorsque la machine virtuelle est lancée, les rouleaux doivent tourner sur eux même avant de s'arrêter aléatoirement. Cela peut être comparé virtuellement à un déplacement uniforme de tous les arguments de ce tableau un nombre de fois aléatoire. Pour cela on utilise l'importation de la capacité de trouver un nombre aléatoire :

- `from random import randint`

L'utilisateur va devoir rentrer le nombre 'n' de jetons qu'il mise soit le nombre de rounds qu'il souhaite que la machine effectue. La somme de jetons après chaque round 'N' prend la

même valeur car aucun rounds n'a encore été joué. Ensuite va commencer la première boucle principale :

- `n=int(input("\033[3m"+"combien de jetons voulez- vous miser ? \n"))`
- `N=n`
- `for i in range (0,n):`
 - *Suite du code*

On utilise ensuite des fonctions `jeu()` afin que les arguments des roues se décalent x-fois vers la droite, x étant un nombre aléatoire d'un intervalle :

- `def jeu1():`
 - `for a in range (0,randint(20,80)):`
 - `tab1.insert(0,tab1.pop())`

Après est utilisé la fonction `tableau()` pour afficher le tableau 3x3 (soit l'écran de la machine) de fin de round en mettant en valeur la ligne du milieu en bleu, la ligne qui sera déterminante pour la suite :

- `def tableau():`
 - `print(" ")`
 - `print("\033[44m'+"1","2","3"+'\033[49m')`
 - `print("\033[47m')`
 - `print((tab1[0]),(tab2[0]),(tab3[0]))`
 - `print("\033[34m','\033[4m")`
 - `print((tab1[1]),(tab2[1]),(tab3[1]))`
 - `print("\033[0m','\033[47m')`
 - `print((tab1[2]),(tab2[2]),(tab3[2]))`
 - `print("\033[49m')`
 - `print(" ")`

Le code demande ensuite à l'utilisateur de choisir s'il veut refaire tourner l'une des roues ou non. Il prend sa décision en pouvant voir le résultat qu'il possède déjà et ainsi choisir laquelle lui est bénéficiaire :

- `print((("\033[3m"+"voulez-vous refaire tourner une roue pour un jeton? \nSi oui, laquelle ?(non/1/2/3)"))`
- `an = input()`

S'il répond négativement ou une réponse non connue du programme aucune roue ne tournera et le programme passera à l'analyse de la ligne en bleu directement :

- `if an == "non":`
 - `print(" ")`
- `else:`
 - `print("considéré comme un 'non'.")`

Si l'utilisateur réponds par l'affirmative (un des numéro des rouleaux) alors il perdra un jeton de plus et cette roue tournera un nombre de fois aléatoire donnant ainsi une nouvelle ligne :

- `elif an == "3":`
 - `N=N-1`
 - `jeu3()`

- `tableau()`

Ensuite la ligne bleue va être lue et en fonction des symboles la somme 'N' va se voir attribuer un gain ou une perte en fonction du résultat, le résultat affiché grâce à la fonction 'jetons()' qui fait office de compteur :

- `def jetons():`
 - `print("\033[42m'+\"vous possédez donc \",N,\" jetons.\"+\033[49m')`
- `elif tab1[1]==tab2[1]==tab3[1]=='X':`
 - `print("c'est gagné, vous gagnez 5 jetons !")`
 - `N=N+5`
 - `jetons()`

Le code est écrit de telle façon que si la somme 'N' est nulle ou égale à 0 la boucle s'arrête et la partie étant considérée perdue le programme ne continue pas.

- `if N < 0 :`
 - `print("perdu :(")`
 - `print("veuillez racheter des jetons si vous voulez rejouer.")`
 - `break`

Si après les n-rounds il reste des jetons à l'utilisateur, le programme va lui dire combien il lui en reste et le programme pourra passer à la deuxième grande boucle :

- `if N>0:`
 - `print("il vous reste \",N,\" jeton(s).")`

La deuxième grande boucle commence si et seulement si 'N' est positif. Cette boucle pour rejouer va se répéter tant que 'N' est supérieur à zéro et que l'utilisateur le souhaite :

- `while N>0:`
 - `print("\033[3m\"+\"Voulez-vous rejouer une partie ? \nVos gains seront augmentés sauf pour deux symboles identiques mais vos pertes le seront aussi.(oui/non)\")`
 - `answer = input()`

Si l'utilisateur ne répond pas "oui" la boucle sera définitivement arrêtée et la partie va s'arrêter :

- `if answer == "non":`
 - `print("Vous possédez finalement \",N,\" jeton(s).")`
 - `print("Vous pourrez échanger vos jetons restants contre des gains si vous le souhaitez.")`
 - `break`
- `else:`
 - `print("Considéré comme 'non'")`
 - `print("Vous possédez finalement \",N,\" jeton(s).")`
 - `print("Vous pourrez échanger vos jetons restants contre des gains si vous le souhaitez.")`
 - `break`

Si l'utilisateur répond "oui", il a alors le choix de combien de ses jetons restants il veut miser. Ce nombre 'j' n'existe que s'il est plus petit que 'N'. S'il est plus grand, le programme ne peut continuer et la question est reposée. Si l'utilisateur a correctement répondu, la partie va continuer quasiment comme lors de la boucle précédente à part quelques différences :

- *elif answer == "oui":*
 - *j=int(input("combien de vos jetons voulez vous jouer ? "))*
 - *if j<=N:*
 - *for i in range (0,j):*
 - *jeu1()*
 - ...
- *else:*
 - *print("\033[41m' + "impossible, réessayer"+'\033[49m')*

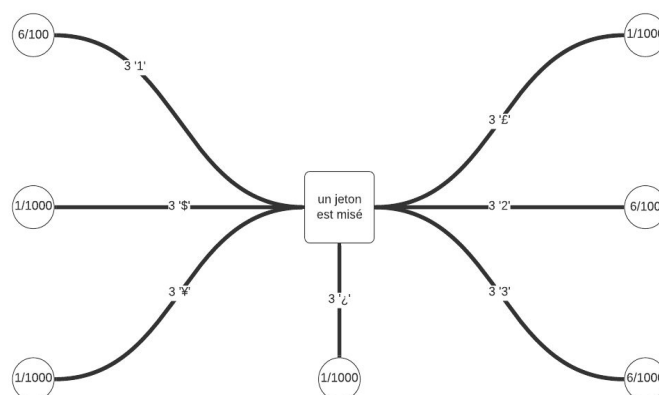
Les différences entre les deux boucles principales se font sur l'augmentation des gains et des pertes à chaque round de 1 sauf quand il y a deux symboles identiques qui reste fixée à 1. Cette différence est possible grâce à la variable 'a', cette variable initialisée à 0 va augmenter à chaque round après lecture de la ligne bleue. Ce changement n'influe pas la possibilité de refaire tourner un des rouleaux :

- *elif tab1[1]==tab2[1]==tab3[1]=='¿':*
 - *print("c'est gagné, vous gagnez",7+a,"jetons !")*
 - *N=N+7+a*
 - *jetons()*

Après chaque round, si la somme 'N' est positive, le joueur aura le choix de rejouer certains de ces jetons. L'initialisation de 'a' étant avant la boucle 'while', les pertes et les gains vont augmenter à partir de ceux du round précédent. À la fin de la partie, le programme souhaite une bonne journée à l'utilisateur. À ce niveau là, pour pouvoir rejouer il faut relancer le programme et ainsi miser de nouveaux jetons :

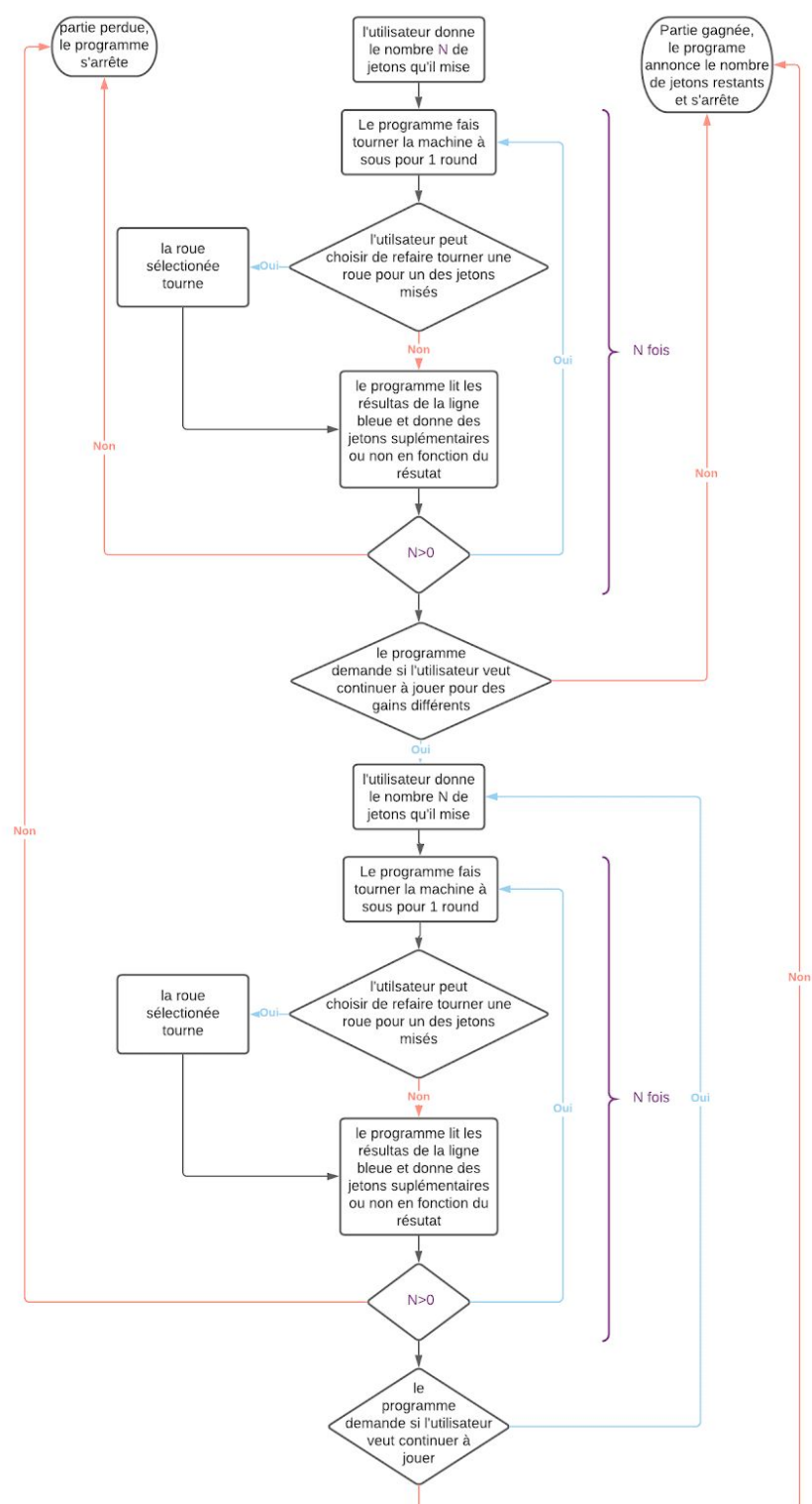
- *print("Passez une bonne journée :) !!!")*

Les probabilités d'obtenir trois symboles identiques en ligne bleue varient entre 6/1000 et 1/1000, rendant la capacité de gagner beaucoup de jetons maigre.



Combinaison	£	2	3	\$	1	€	¥	Gagné	Perdu
Rouleur 1	1/10	2/10	3/10	1/10	1/10	1/10	1/10		
Rouleur 2	1/10	3/10	1/10	1/10	2/10	1/10	1/10		
Rouleur 3	1/10	1/10	2/10	1/10	3/10	1/10	1/10		
Probabilité totale par combinaison	= 1/1000	= 6/1000	= 6/1000	= 1/1000	= 6/1000	= 1/1000	= 1/1000	= 22/1000	= 978/1000

Organigramme du fonctionnement de la machine à sous virtuelle



Analyse informelle du jeu :

On part du principe que l'utilisateur joue pour gagner le plus possible de gains. Pour cela si la ligne bleue est composée de trois mêmes symboles il ne fera pas retourner la roue, si elle est composée de 2 mêmes symboles il fera tourner celle avec le symbole différent et si les trois symboles sont différents alors il va faire tourner les roues pour espérer avoir deux symboles, sachant que tous les caractères spéciaux sont représentés une fois par roue et que les chiffres de façon inéquitable (trois '3' dans le rouleau 1, un seul dans le rouleau 2 par exemple)

- Si l'utilisateur mise 10 jetons et suit la méthode précédente, il a environ % chances d'avoir encore des jetons à la fin de la première boucle principale. Le nombre de jetons restants est souvent inférieur à 5 car le joueur a plus de chance d'obtenir deux symboles identiques que trois pareils et encore moins trois symboles spéciaux pareils.
 - S'il rejoue tous ses jetons restants, les pertes et les gains vont augmenter rapidement et le joueur va perdre rapidement. En effet les pertes étant de 4 jetons au premier round, l'utilisateur ne peut continuer longtemps car il n'a plus de jetons rapidement. La plupart du temps il ne peut faire que 1 ou 2 rounds s'il n'obtient pas de triplé gagnant au premier round et ainsi perd la partie.
 - S'il ne rejoue pas ces jetons restants, l'utilisateur a en moyenne 3 jetons restant, n'ayant pas perdu la partie mais perdu tout de même plus de la moitié de la mise.
- Si le joueur mise 15 jetons et suit la même méthode que pour 10 jetons, il n'a presque aucune chance de perdre avec une moyenne de 4,5/5 de chances d'avoir encore des jetons à la fin des 15 rounds ne dépassant que rarement 10 jetons restants. Cependant si la moyenne est élevée c'est qu'il obtient facilement un reste de près de 10 jetons (% de chances) et un reste inférieur à 5 jetons (% de chances) .
 - S'il rejoue tous ses jetons restants, il a en moyenne ¼ de pouvoir réaliser tous les rounds et d'avoir encore des jetons à la fin. En effet, bien que les gains augmentent rapidement, les "gros" gains sont rares alors que les "grosses" pertes sont fréquentes. Si l'utilisateur a assez de chance pour pouvoir rejouer une deuxième fois, il ne parvient généralement pas à faire plus de deux rounds avant de perdre.
 - S'il choisit de ne pas rejouer, l'utilisateur a en moyenne 7 jetons restants, soit près de la moitié de la mise.
- Si le joueur mise 20 jetons et utilise la même méthode que les mises précédentes, il n'a presque aucune chance de perdre avec près de 4.8/5 de probabilité d'avoir encore des jetons à la fin des 20 rounds, généralement moins de 15 jetons restants.

- S'il rejoue tous ses jetons restants, il n'a que très peu de chances de pouvoir finir ses rounds et de pouvoir rejouer. Le nombre de rounds étant importants, la quantité de perte de jetons devient très vite très importante malgré des gains considérables. Ainsi l'utilisateur ne peut rejouer que s'il a eu des gains importants à presque chaque round ce qui est peu probable.
- S'il ne rejoue pas, l'utilisateur a en moyenne 12 jetons restant soit près de $\frac{3}{4}$ de la mise.

Le joueur n'a que peu de chance de sortir gagnant de la machine à sous. En effet, dès le début il va gagner facilement des jetons même sans forcément un triplé sur la ligne bleu :

- *elif tab1[1]==tab2[1] :*
 - *N=N+1*
 - *print("Vous gagnez 1 jetons !")*
 - *jetons()*
 - *a=a+1*

Ce qui va le conduire à souvent faire retourner un des rouleaux pour essayer d'avoir un meilleur résultat en ligne bleue car il aura l'impression d'en quelque sorte d'avoir la capacité de rembourser cette perte cette perte facilement du moment que les trois symboles en bleu ne sont pas tous différents. Pourtant, malgré un retournement d'une roue, cette combinaison est courante et le fait d'avoir fait retourner une roue augmente cette perte.

Ce schéma de perte facile se poursuit s'il a la capacité de rejouer. Ce qui est encore plus vicieux est que si dans cette boucle on ne paye pas plus cher pour refaire tourner un rouleau et on ne gagne pas plus pour deux symboles identiques , le reste des pertes et des gains, eux, augmentent, faisant de chaque tour de rouleau supplémentaire une capacité de perte très importante. Le joueur est ainsi berné par la capacité de gains importants et va rapidement perdre beaucoup de jetons à presque tous les rounds.

Un autre aspect de la machine à sous qui va "berner" l'utilisateur est le fait que le nombre de rounds correspond au nombre de jetons misés. En effet, on va logiquement penser que plus on mise de jetons, plus on a de chances de gagner, un nombre de round plus important étant la solution la plus logique pour obtenir de meilleures combinaisons plus régulièrement. Si ce n'est que les rouleaux tournant aléatoirement, cette chance n'est qu'une chance et est plus régulièrement transcrite en cas de partie continuée et rejouée par des pertes importantes se succédant.

Bien sûr, les rouleaux tournant aléatoirement, la possibilité d'enchaîner les triplés gagnant sans pertes est une possibilité mais très très infime.

changements de couleurs affichées sur la console trouvées sur :
[Affichage en couleur sur le terminal — documentation Sokoban 0.9.2.0 \(univ-grenoble-alpes.fr\)](#)
[ANSI color codes in Python · GitHub](#)