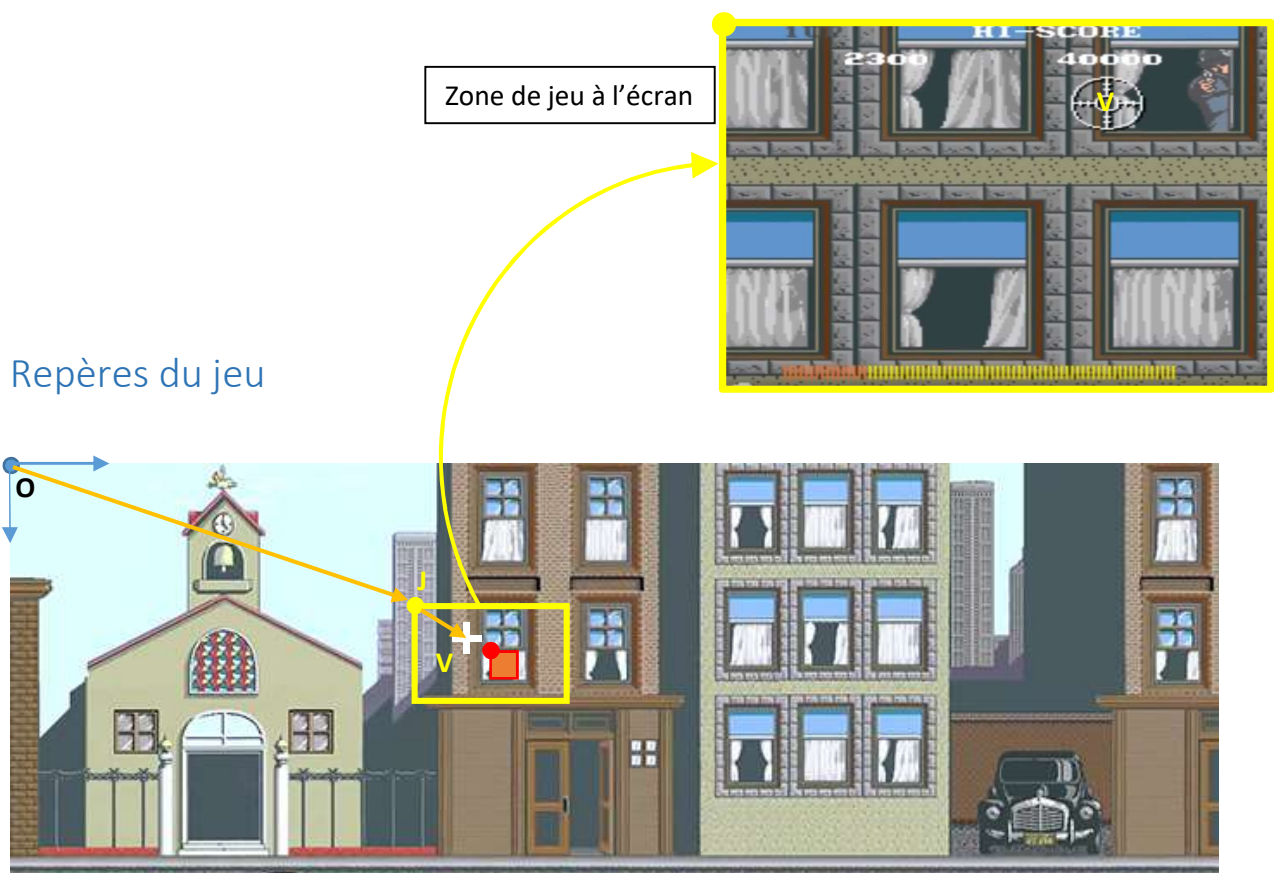
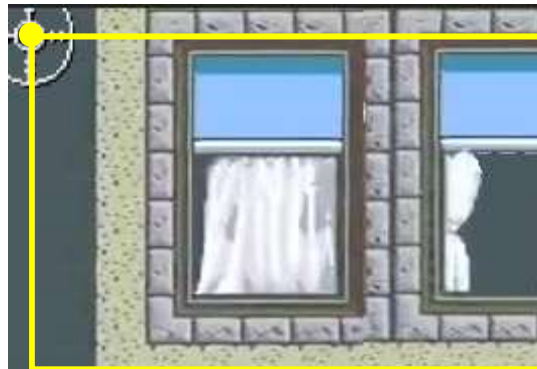


Empire City

Empire city (surnom de New York) représente un jeu de tir sniper à l'époque de la prohibition. Après une décennie de jeux exploitant un fond fixe, Empire City met en scène une zone large de jeu avec un timing bien dosé mettant ainsi en place un gameplay assez addictif. L'écran grâce au déplacement du viseur du sniper peut se déplacer (scrolling) en vertical et en horizontal !! Incroyable. C'est cet effet que nous allons mettre en place dans ce tutoriel.



Voici le décor complet du jeu faisant une résolution de 2000x686 en tout. En graphisme, nous rappelons que le l'axe des y est orienté de haut en bas. Par conséquent, le centre du repère se situe en haut à gauche (rond bleu). Nous représentons par un cadre jaune de 400x300 la zone d'affichage du jeu. Sa position dans le repère du décor est définie par la position de son point haut-gauche symbolisé par un rond jaune. La position d'un bandit (en rouge) est de la même manière définie dans le repère du décor. La position du viseur (V : croix blanche) du tireur est cependant définie dans les coordonnées du repère écran. Ainsi lorsque $\text{viseur_xecran} = \text{viseur_yecran} = 0$, le viseur est positionné en haut à gauche de l'écran du jeu comme dans l'image ci-dessous :



Test d'intersection

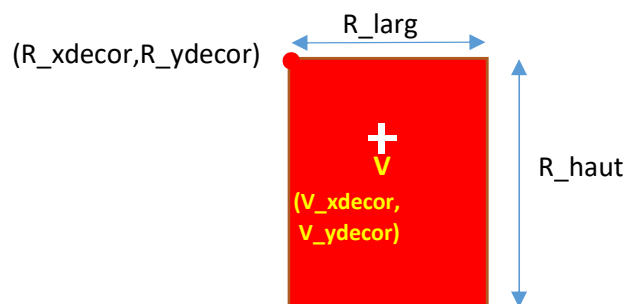
Comment tester si notre joueur tire bien sur le bandit ou à côté ? Tout d'abord, avant de faire les tests, nous allons ramener les coordonnées du viseur dans le repère du décor.

Si J désigne la position de la zone jaune dans le repère du décor et V la position du viseur dans le sous-repère de l'écran (zone jaune), nous avons : $V_{decor} = J_{decor} + V_{ecran}$

Pour calculer cela, nous allons faire :

$V_{xdecor} = J_{xdecor} + V_{xecran}$ et la même chose pour la composante y .

Maintenant que nous avons les coordonnées du viseur dans le repère principal (decor), il nous suffit de savoir si le point V appartient au rectangle rouge défini par $(R_{xdecor}, R_{ydecor}, R_{larg}, R_{haut})$:



Le test est assez simple. L'abscisse du bord gauche est donnée par R_{xdecor} . L'abscisse du bord droit est donnée par $R_{xdecor} + R_{larg}$. Il suffit de comparer V_{xdecor} à ces deux valeurs pour savoir si le Viseur est bien dans la zone rouge horizontalement parlant. Il faut ensuite tester les ordonnées pour déterminer si V_{ydecor} est bien compris entre R_{ydecor} et $R_{ydecor} + R_{haut}$.

TODOLIST

- Nous définissons la zone à extraire dans le décor grâce à la ligne suivante :

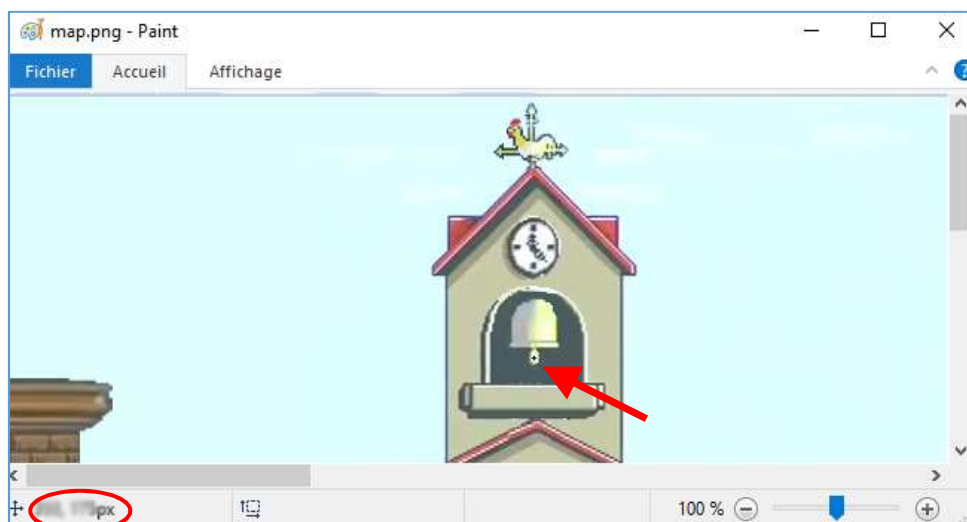
```
zonejaune = pygame.Rect( 0, 0, screenWidth, screenHeight )
```

Ensuite, cette zone jaune est copiée puis collée dans la zone de dessin par la fonction habituelle :

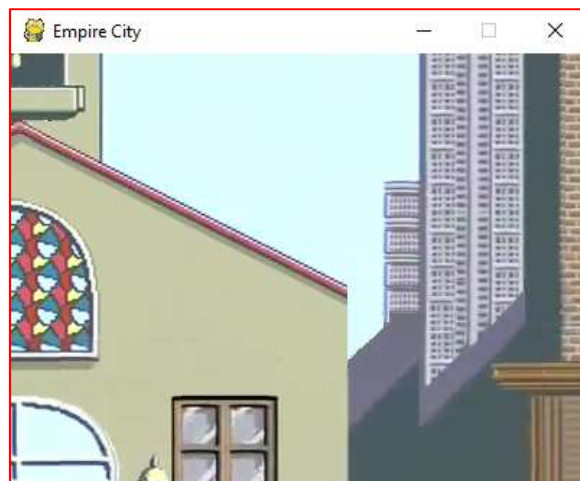
```
screen.blit(fond,(0,0),area = zonejaune)
```

La coordonnée (0,0) indique qu'il faut coller la zone jaune en haut à gauche de la fenêtre de jeu. Comme le rectangle de copie est de la même largeur et hauteur que la fenêtre d'affichage, la fenêtre de jeu est entièrement remplie.

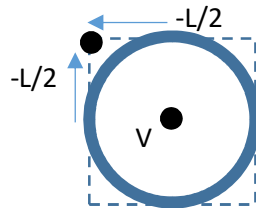
Nous allons positionner la fenêtre de jeu de telle sorte que le bas de la cloche soit en haut à gauche de la fenêtre de jeu. Ouvrez l'image map.png dans mspaint (clic droit sur le fichier PNG puis choisissez modifier). Dans l'éditeur, déplacez votre curseur pour sur le bas de la cloche. En bas à droite, le logiciel nous indique la coordonnée en pixel :



Notez ces valeurs et modifiez la définition de zonejaune pour obtenir ceci lors du lancement :

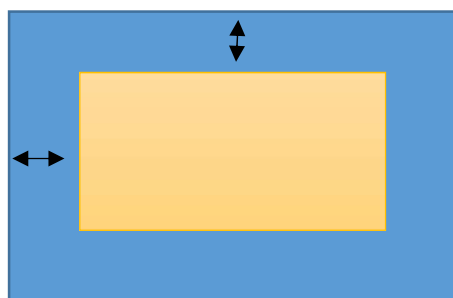


- Créez maintenant des variables J_xdecor , J_ydecor et initialisez pour que le jeu commence sur le devant de l'église. Pensez à modifier la définition de zonejaune
- Chargez l'image du viseur dans un sprite. Créez des variables V_xecran et V_yecran et initialisez-les pour que le viseur se situe au milieu de la fenêtre de jeu. Dessinez le viseur à l'écran. Attention, pour que le sprite du viseur soit centré correctement sur le point V, il faut le décaler de la moitié de sa largeur :



- Avec l'appui des touches clavier $\downarrow \rightarrow \leftarrow \uparrow$ modifiez la position du viseur. A ce niveau, le décor ne bouge pas encore
- Mise en place du scrolling. Nous allons définir dans la fenêtre une zone libre en orange dans laquelle le curseur évolue librement et le scrolling n'est pas actif. Dès que le centre du viseur entre dans la zone bleu, il faut modifier son x/y pour le remettre dans la zone orange (comme dans le cas de la collision du palet sur la raquette de pong). Ensuite, il faut modifier les valeurs de J_xdecor et J_ydecor en conséquence pour que le décor bouge (et non le viseur) à la place du viseur.

La zone orange est définie par rapport à la distance au bord de la fenêtre de jeu, distance symbolisée par les flèches sur le schéma. Vous choisirez une valeur pour la distance que vous trouverez pertinente en testant.



- Attention, le joueur peut maintenant scroller dans toutes les directions et sortir de la zone de la map !!! Lorsque le scrolling crée des valeurs erronées pour J_xdecor et J_ydecor , corrigez-les pour éviter tout problème !

- Choisissez un sprite parmi les fichiers « bandit_rue ». Tirez sa position en x aléatoirement. Au bout de trois secondes, faites apparaître le bandit sur le trottoir.

- ♣ `int(pygame.time.get_ticks()/1000)` retourne le temps en seconde

- ♣ Dès le démarrage du jeu, sauvegardez la valeur de l'horloge dans une variable T0

- Gérez le tir en détectant l'appui sur la barre espace. Si vous détectez l'intersection entre les coordonnées du tir et la zone du bandit, le méchant disparaît. Faites réapparaître un nouveau méchant 3 secondes après la mort du précédent.

- Après chaque tir, faites dévier la position du viseur pour un peu plus de réalisme

- ♣ La fonction `random.random()` retourne un numérique entre 0 et 1

- Lorsque le viseur est trop éloigné du méchant, faites apparaître les sprites flèche gauche ou droite pour aider le joueur à localiser le bandit

- Repérez quatre coordonnées de fenêtres et choisissez un sprite « bandit_window ». Faites apparaître aléatoirement un bandit sur le trottoir et un bandit à la fenêtre avec des probabilités de 50/50. Dans le cas du bandit fenêtre, choisissez au hasard une des quatre fenêtres présélectionnées.

PROJET

Il est possible de poursuivre ce jeu comme en projet. Il reste pas mal d'options à mettre en place afin de gérer les diverses sprites fournis dans les supports :

- Affichez le nombre de balles restantes. On démarre la partie avec 12 balles. Affichez-les en bas à gauche en mettant un sprite balle par balle en stock. Un tir consomme une balle, un méchant tué fait gagner 2 balles. S'il n'a plus de balle, le joueur ne peut plus tirer

- Les sprites « bandit_window » peuvent uniquement être positionnés à une fenêtre car ils sont coupés en bas et sur un côté

- Les sprites « bandit_mur » sont coupés sur un côté mais ni en bas ni en haut. Ils ne peuvent s'accoler qu'à un mur

- Sprite « bandit_egout » et oui tous les coups sont permis dans la mafia !
- Sprite « woman » : attention, si le joueur tire sur la femme en rouge, il perd trois balles
- Sprite « bandit_appui » sont en général postés au milieu d'une fenêtre ou sur un toit
- Utilisez tous les sprites bandits disponibles dans chaque catégorie
- Au bout de 3 secondes maintenant le méchant tire et fait perdre une balle/vie au joueur, il tire toutes les 3 secs tant qu'il n'est pas mort. Affichez un compteur en bas au milieu de l'écran de jeu qui indique le nombre de secondes restantes avant le tir du bandit
- Si le joueur n'a plus de balles, affichez « DEAD » en gros caractères au centre de l'écran en blanc
- Ecran esquive : en appuyant sur ENTER, on affiche le sprite « hide » sur la totalité de la surface de jeu. Le joueur ne reçoit pas de dégât même si le bandit tire. Par contre, la visibilité est fortement diminuée !!
- Un tir fait afficher le sprite « bang » pendant 2 secondes à l'écran. Si le joueur tire plusieurs fois, il y aura plusieurs sprites « bang » à l'écran. Chaque bang vit deux secondes avant de disparaître
- Pour les bandits autres que ceux du trottoir, repérez un maximum d'endroits possibles pour varier les emplacements
- Le sprite « bandit_rue3 » permet de faire un bandit qui court, il suffit de décaler sa position en x au fil du temps
- ♣ Pour traiter ce projet sereinement, il vous ait conseillé d'utiliser un maximum les listes qui doivent vous faciliter la vie. Par exemple, l'ensemble des positions possibles pour les tireurs de fenêtre peut être stockée dans une liste. Ensuite, vous pouvez associer à chaque catégorie un entier : 1 pour FENETRE, 2 pour MUR, 3 pour TOIT... Et si vous stockez vos listes dans une liste !! Alors sprites[TOIT] vous donnera la liste des sprites des tireurs de toit et pos[TOIT] la liste des positions possibles pour les tireurs sur le toit ... Magique non !