



# Réseau de neurones dans Pac-Man

Francis BISSON  
Kenny CÔTÉ  
Benjamin HÉROUX-CLÉROUX

2009-05-25

# Plan de la présentation

- Description de la problématique
- Simplification du problème
- Choix de l'approche
- Stratégie d'implémentation
- Démonstration
- Expérimentations et résultats
- Conclusion

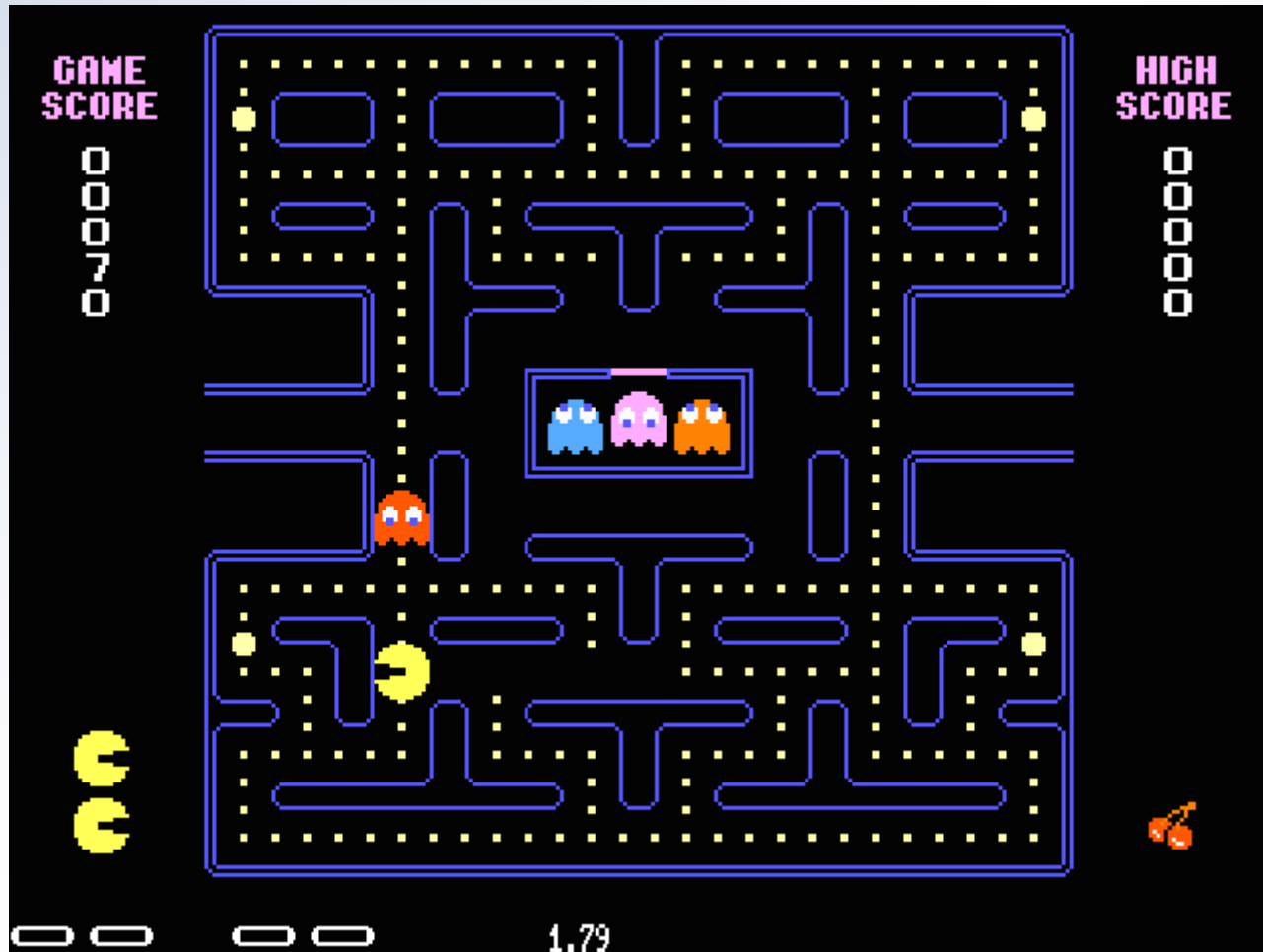
# Plan de la présentation

- Description de la problématique
- Simplification du problème
- Choix de l'approche
- Stratégie d'implémentation
- Démonstration
- Expérimentations et résultats
- Conclusion

# Présentation de la problématique

- Jeu vidéo inventé en 1980 chez Namco au Japon
- Pac-Man doit manger les points sans se faire attraper par les fantômes
- Les *power pellets* (4) permettent à Pac-Man de manger les fantômes durant une période de temps limitée
- Les fantômes morts réapparaissent après un certain temps
- Des items spéciaux (fruits) donnent des points bonus
- Pac-Man peut se téléporter d'un côté à l'autre de l'écran

# Présentation de la problématique



# Plan de la présentation

- Description de la problématique
- Simplification du problème
- Choix de l'approche
- Stratégie d'implémentation
- Démonstration
- Expérimentations et résultats
- Conclusion

## Simplification du problème

- Changements de direction aux intersections seulement
- Pas de *power pellets* (donc fantômes invincibles)
- Pas de points bonus
- Une seule vie pour Pac-Man

# Plan de la présentation

- Description de la problématique
- Simplification du problème
- Choix de l'approche
- Stratégie d'implémentation
- Démonstration
- Expérimentations et résultats
- Conclusion



# Choix de l'approche

- Besoin d'un algorithme pouvant fournir une réponse rapidement (jeu avec décisions en temps réel)
- Environnement dynamique (fantômes, points)
- → Réseau de neurones

# Plan de la présentation

- Description de la problématique
- Simplification du problème
- Choix de l'approche
- **Stratégie d'implémentation**
- Démonstration
- Expérimentations et résultats
- Conclusion

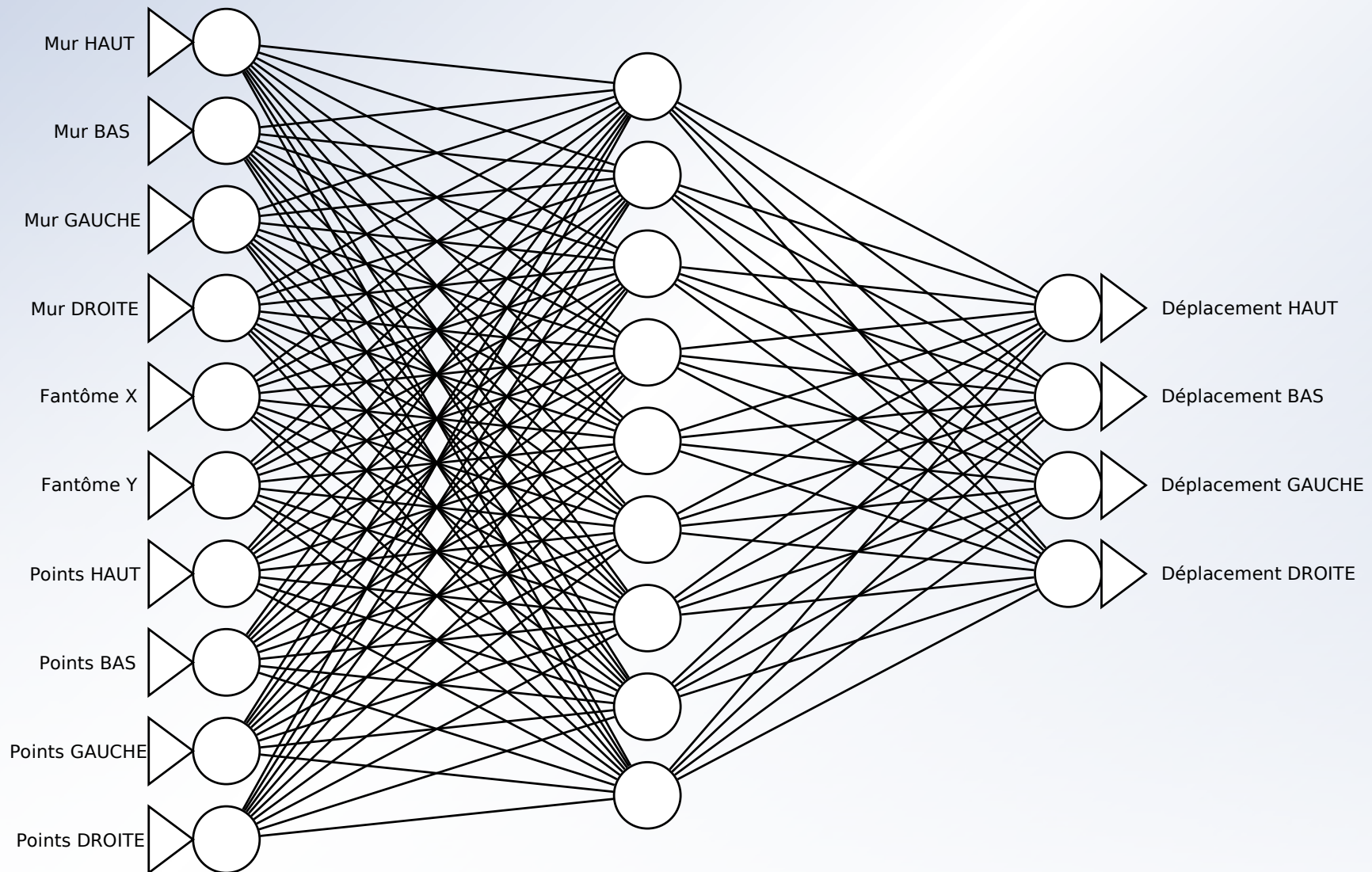
# Stratégie d'implémentation

- Programmé en C++
- Librairie graphique SDL (LGPL, multiplateforme)
- Technique d'I.A. pour les fantômes :
  - Semi-aléatoire
- Technique d'I.A. pour Pac-Man :
  - Réseau de neurones
  - Apprentissage supervisé

# Stratégie d'implémentation (topologie du réseau)

- 10 neurones en entrée
  - (4) Murs  $\uparrow, \downarrow, \leftarrow, \rightarrow$
  - (2) Distance de Manhattan du fantôme le plus proche de Pac-Man (rayon de vision)
  - (4) Nombre de points  $\uparrow, \downarrow, \leftarrow, \rightarrow$
- 9 neurones cachés
- 4 neurones en sortie
  - Direction à prendre  $\uparrow, \downarrow, \leftarrow, \rightarrow$

# Stratégie d'implémentation (topologie du réseau)



## Stratégie d'implémentation (paramètres d'apprentissage)

- Apprentissage par rétropropagation
- Taux d'apprentissage : 0.1
- Fonction d'activation : fonction sigmoïde
- Limite de 50 000 époques d'apprentissage
- Données d'apprentissage colligées à partir des actions d'un joueur humain

# Plan de la présentation

- Description de la problématique
- Simplification du problème
- Choix de l'approche
- Stratégie d'implémentation
- **Démonstration**
- Expérimentations et résultats
- Conclusion

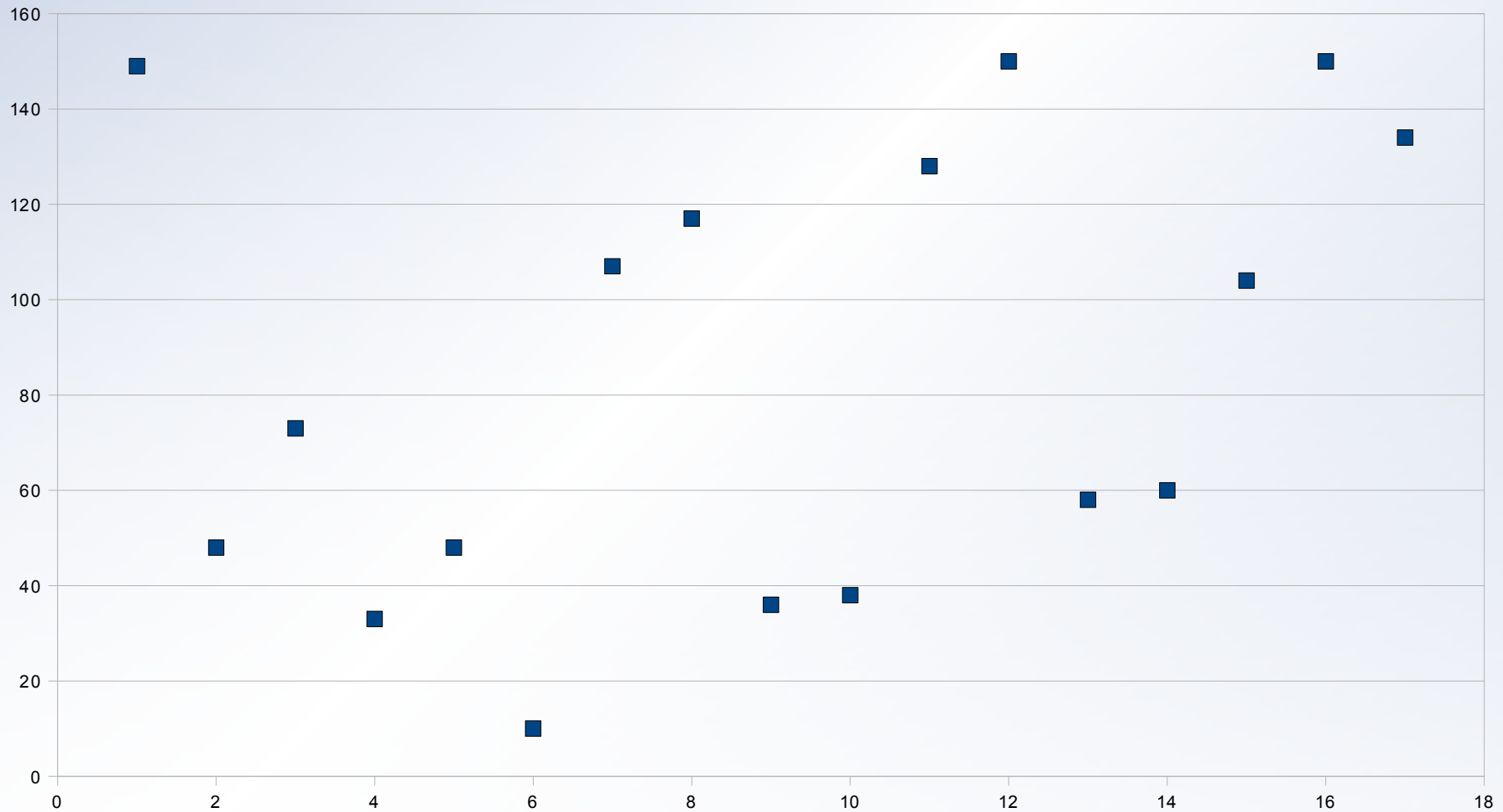
# Plan de la présentation

- Description de la problématique
- Simplification du problème
- Choix de l'approche
- Stratégie d'implémentation
- Démonstration
- **Expérimentations et résultats**
- Conclusion



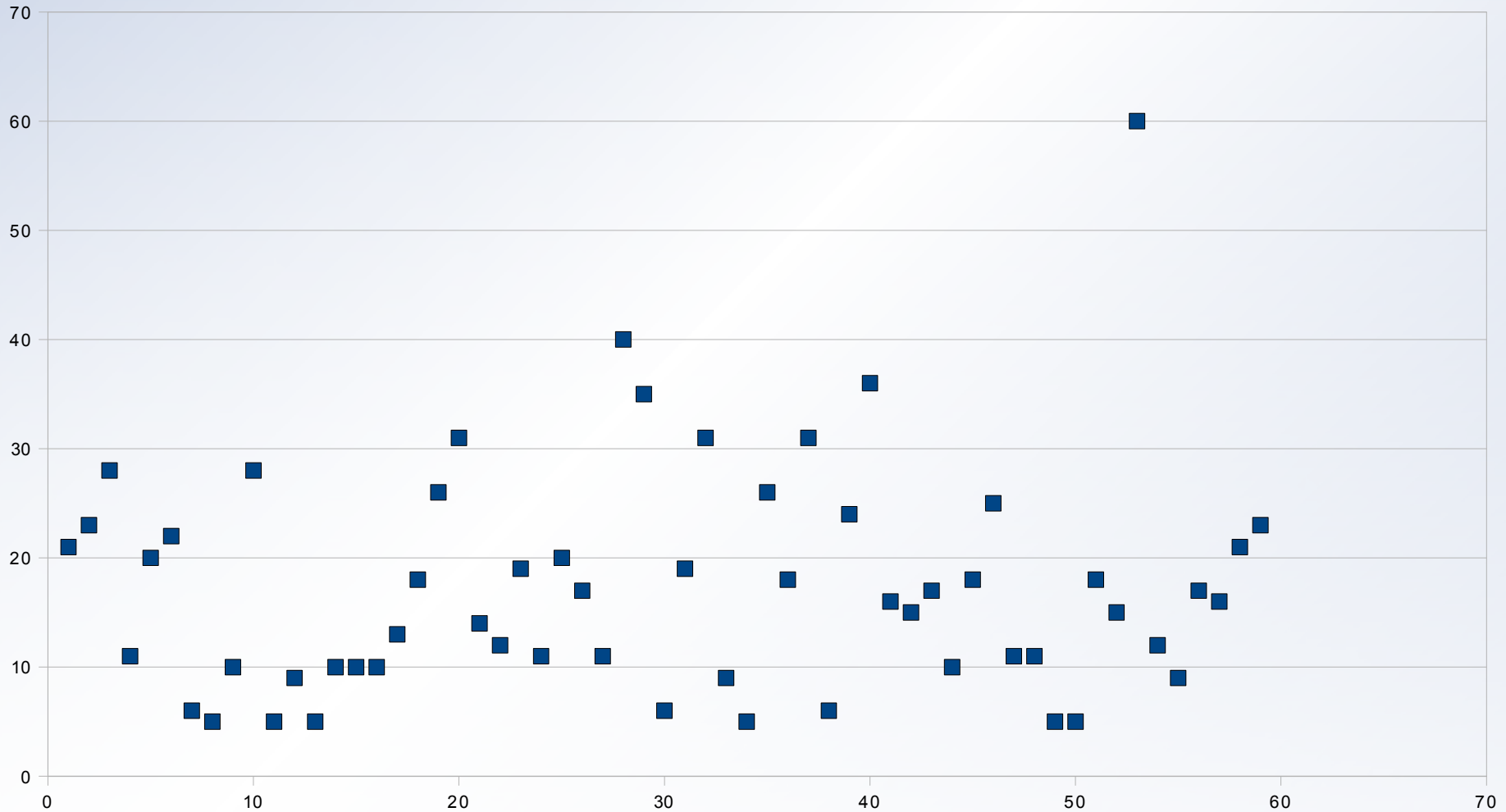
# Expérimentations et résultats (joueur humain)

Score par partie



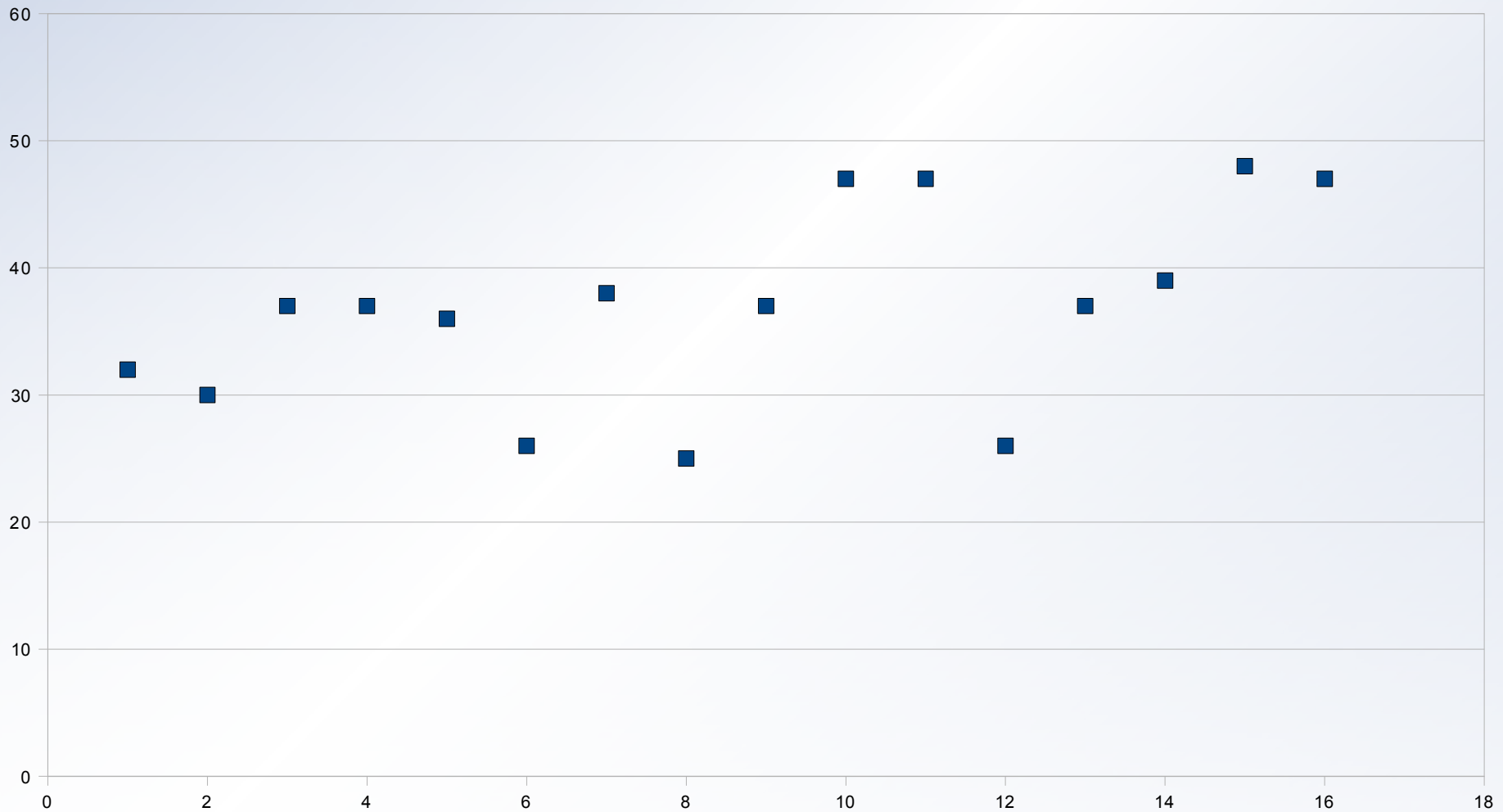
# Expérimentations et résultats (joueur aléatoire)

Score par partie



# Expérimentations et résultats (joueur intelligent)

Score par partie



# Plan de la présentation

- Description de la problématique
- Simplification du problème
- Choix de l'approche
- Stratégie d'implémentation
- Démonstration
- Expérimentations et résultats
- Conclusion

# Conclusion

- Le joueur intelligent en général meilleur que le joueur aléatoire
- Le comportement semi-aléatoire des fantômes est difficile à prédire
- La présence de bruit dans les données d'apprentissage peut causer du surapprentissage (*overfitting*)
- Un trop petit nombre de données d'apprentissage n'offre pas un bon pouvoir de généralisation

## Conclusion (améliorations éventuelles)

- Prétraiter les données d'apprentissage
- Augmenter la dimension de l'ensemble d'apprentissage
- Tenir compte de la grille de jeu au complet
- Utiliser la position absolue des fantômes
- Altérer la topologie du réseau de neurones à l'aide d'un algorithme génétique