



## Piscine Unity - d02

Audio, Animation et Communication entre les scripts

Staff [staff@staff.42.fr](mailto:staff@staff.42.fr)

*Résumé: Ce document contient le sujet du jour 02 de la piscine Unity de [42](#).*

# Table des matières

<b>I</b>	<b>Préambule</b>	<b>2</b>
<b>II</b>	<b>Consignes</b>	<b>3</b>
<b>III</b>	<b>Exercice 00 : Point and click</b>	<b>5</b>
<b>IV</b>	<b>Exercice 01 : Plus on est de fous</b>	<b>6</b>
<b>V</b>	<b>Exercice 02 : Batiments et villages</b>	<b>7</b>
<b>VI</b>	<b>Exercice 03 : De l'action</b>	<b>8</b>
<b>VII</b>	<b>Exercice 04 : Joueur vs IA</b>	<b>10</b>

# Chapitre I

## Préambule

Aujourd'hui vous allez vous transporter dans l'univers féérique et violent de Warcraft. Des humains, des orcs et surtout du sang. Vous apprendrez les rudiments et mécaniques du RTS.

Au programme donc, du combat, des unités dévouées et prêtes à tout pour servir le grand maître que sera votre souris. Des villages pillés, réduits en ruines et en cendres. Pour vous accompagner dans votre quête rien de tel que le menestrel Youtube qui fera rugir les tambours de guerre pendant votre recherche :

- [Une bande son qu'elle est bien](#)
- Bon et parce que tout ça c'est quand même un peu violent, voilà de quoi [compenser un peu](#).

Bon courage à vous tous et que votre journée soit victorieuse.

# Chapitre II


## Consignes

- Seule cette page servira de référence : ne vous fiez pas aux bruits de couloir.
- Le sujet peut changer jusqu'à une heure avant le rendu.
- Si aucune information contraire n'est explicitement présente, vous devez assumer les versions de langages suivantes :
  - Ruby `>= 2.3.0`
  - pour d09 Rails `> 5`
  - mais pour tous les autres jours Rails `4.2.7`
  - HTML 5
  - CSS 3
- Nous vous interdisons FORMELLEMENT d'utiliser les mots clés `while`, `for`, `redo`, `break`, `retry`, `loop` et `until` dans les codes sources Ruby que vous rendrez. Toute utilisation de ces mots clés est considérée comme triche (et/ou impropre), vous donnant la note de -42.
- Les exercices sont très précisément ordonnés du plus simple au plus complexe. En aucun cas, nous ne porterons attention ni ne prendrons en compte un exercice complexe si un exercice plus simple n'est pas parfaitement réussi.
- Attention aux droits de vos fichiers et de vos répertoires.
- Vous devez suivre la procédure de rendu pour tous vos exercices : seul le travail présent sur votre dépôt GIT sera évalué en soutenance.
- Vos exercices seront évalués par vos camarades de piscine.
- Vous ne devez laisser dans votre répertoire aucun autre fichier que ceux explicitement spécifiés par les énoncés des exercices.
- Vous avez une question ? Demandez à votre voisin de droite. Sinon, essayez avec votre voisin de gauche.
- Votre manuel de référence s'appelle `Google` / `man` / `Internet` / ....
- Pensez à discuter sur le forum Piscine de votre Intra, ou Slack, ou IRC...

- Lisez attentivement les exemples. Ils pourraient bien requérir des choses qui ne sont pas autrement précisées dans le sujet...
- Par pitié, par Thor et par Odin ! Réfléchissez nom d'une pipe !

# Chapitre III

## Exercice 00 : Point and click


	Exercice : 00
Exercice 00 : Point and click	
Dossier de rendu : <i>ex00/</i>	
Fichiers à rendre : <b>Le dossier de votre projet.</b>	
Fonctions interdites : <b>Aucune</b>	

Dans ce premier exercice vous devez mettre en place un personnage sur la map fournie dans les assets. On doit pouvoir le déplacer en cliquant sur un endroit sur la map qui fera office de destination. Il ne doit pas se téléporter mais avancer jusqu'à sa destination. Le sprite doit également être orienté en direction de sa destination.

Rendez votre personnage un peu plus vivant en lui ajoutant un son lorsqu'il commence à se déplacer. Ajoutez lui également une animation de marche qui ne se déclenchera que lorsqu'il marche donc.

# Chapitre IV

## Exercice 01 : Plus on est de fous

	Exercice : 01
Exercice 01 : Plus on est de fous	
Dossier de rendu : <i>ex01/</i>	
Fichiers à rendre : <b>Le dossier de votre projet.</b>	
Fonctions interdites : <b>Aucune</b>	


Un personnage c'est bien, plusieurs c'est mieux. Vous allez donc mettre en place plusieurs personnages (avec les memes caractéristiques) et faire en sorte qu'on puisse les selectionner avant de leur donner un ordre.

On doit donc pouvoir cliquer gauche sur un personnage pour le selectionner. Ceci l'ajoute à la selection active. Si un ou des personnages sont déjà présent dans la selection active, alors elle est vidée. On doit pouvoir néanmoins rajouter un personnage sans vider la selection active en cliquant gauche avec la touche **control** enfoncée.

Si on clique gauche ailleurs que sur un personnage alors toute la selection va en direction du point visé comme lors de l'exercice précédent. Si on clique droit la selection est vidée.

# Chapitre V

## Exercice 02 : Batiments et villages

	Exercice : 02
Exercice 02 : Batiments et villages	
Dossier de rendu : <i>ex02/</i>	
Fichiers à rendre : <b>Le dossier de votre projet.</b>	
Fonctions interdites : <b>Aucune</b>	

Le but de cet exercice va être de mettre en place deux villages ennemis : l'un Orc et l'autre Humain. Les deux villages sont identiques, toutefois les sprites les représentant seront bien entendus différents.

Un village est composé d'un hôtel de ville et de 4 bâtiments de votre choix. Le nombre de points de vie des bâtiments est à votre discretion, toutefois l'hôtel de ville doit en avoir significativement plus que les 4 autres.


L'hôtel de ville de chaque village va spawner un nouveau guerrier humain ou orc, selon sa race, toutes les 10 secondes devant sa porte. Ce n'est pas grave si les personnage se superposent.

Les guerriers du joueur doivent pouvoir être déplacés conformément à l'exercice précédent. Les guerriers ennemis restent statiques pour l'instant puisque nous aborderons les combats dans le prochain exercice.



# Chapitre VI

## Exercice 03 : De l'action

	Exercice : 03
Exercice 03 : De l'action	
Dossier de rendu : <i>ex03/</i>	
Fichiers à rendre : <b>Le dossier de votre projet.</b>	
Fonctions interdites : <b>Aucune</b>	

Tout est en place il ne manque plus que d'un peu d'action. Vos personnages vont donc enfin pouvoir taper tout ce qui bouge, voire ce qui ne bouge pas.

Ajoutez donc la possibilité de pouvoir attaquer une cible. On doit pouvoir cliquer sur une cible pour faire attaquer toute la selection active. Si les personnages sont trop loins pour attaquer ils s'approcheront jusqu'à une distance correcte pour le faire. Si la cible s'échappe ils devront la suivre à moins que vous ne lui donniez un ordre contraire.

Vos personnages peuvent attaquer les unités et les bâtiments ennemis. N'oubliez pas d'ajouter un son et une animation pour le combat. Lorsqu'une unité ou un bâtiment n'a plus de vie, celui-ci disparaît accompagné d'un son correspondant. La partie s'arrête lorsqu'un des deux hôtels de ville est détruit.

Attention : à chaque fois qu'un bâtiment qui n'est pas un hotel de ville est détruit, le temps de spawn d'unité de l'hôtel de ville du joueur concerné augmente de 2,5 secondes. Cela signifie donc que s'il ne reste que son hôtel de ville à un joueur, les unités spawneront toutes les 20 secondes.

Ajoutez également des informations dans la console pour savoir qui est attaqué et combien de vie il lui reste.

```
Orc Unit [50/100]HP has been attacked.
```

Faites également en sorte d'afficher un message lorsque la partie est terminée :


```
The Human Team wins.
```



Pour cet exercice, le camp ennemi est toujours inerte, ne perdez donc pas de temps à vouloir scripter une petite IA. Cela sera pour l'exercice suivant.

# Chapitre VII

## Exercice 04 : Joueur vs IA

	Exercice : 04
Exercice 04 : Joueur vs IA	
Dossier de rendu : <i>ex04/</i>	
Fichiers à rendre : <b>Le dossier de votre projet.</b>	
Fonctions interdites : <b>Aucune</b>	

Maintenant que votre jeu est prêt, il est temps de scripter un adversaire. Votre opposant va donc vouloir détruire tout ce qui est sur son passage, car il vous déteste personnellement. Sa première priorité sera donc de vous raser, et en particulier votre hôtel de ville. Cependant si ses unités se trouvent à proximité des vôtres, il ne pourra s'empêcher de laisser tomber son objectif un instant pour les défoncer.

De plus, si l'hôtel de ville de l'IA est attaqué par le joueur humain, l'IA rappelle des unités pour le défendre.

Les consignes précédentes forment le minimum acceptable pour votre IA. Une fois que ces comportements de base sont implementés correctement dans votre rendu, vous devez améliorer votre IA pour la rendre moins stupide et plus intéressante à affronter.