



# Retour d'expérience sur le premier atelier Zeste de Code

---

9 mars 2020



# Table des matières

1.	Un premier atelier à l'organisation... tendue . . . . .	1
1.1.	Chronologie de l'atelier et de son organisation . . . . .	1
1.2.	Quelques documents pour illustrer nos propos . . . . .	4
2.	Une première expérience et déjà de nouvelles idées . . . . .	7
2.1.	Simplifier le code source en s'inspirant d'Arduino . . . . .	7
2.2.	Configurer les logiciels et coder le jeu en français . . . . .	8
2.3.	Enseigner les bases de la programmation pendant la création du jeu vidéo . . . . .	8
2.4.	Commencer plus tôt la communication sur l'événement . . . . .	9
2.5.	Diriger les participants les plus inexpérimentés vers Scratch . . . . .	9
3.	Comment organiser un atelier de programmation ? Un petit guide sans prétention . . . . .	9
3.1.	En amont de l'événement . . . . .	9
3.2.	Le jour de l'événement . . . . .	12
3.3.	Après l'événement . . . . .	14
	Contenu masqué . . . . .	14

Le mercredi 24 octobre 2018 était une grande première pour l'association Zeste de Savoir car celle-ci a organisé son premier événement grand public [IRL](#) ! Nommé « Zeste de Code, la programmation sans pépins ! », cet atelier d'apprentissage de la programmation s'est tenu à Orsay, au sud de Paris, dans le cadre de la Semaine européenne du Code grâce au soutien de [Meet and Code](#) [↗](#) . Une quinzaine de participants ont pu ainsi réaliser, selon leur choix, un [Flappy Bird](#) [↗](#) en Scratch ou un [Snake](#) [↗](#) en Python !

Cette première expérience est globalement un succès, malgré quelques améliorations à apporter pour les ateliers. Dans cet article, nous vous proposons de revivre avec nous cette première expérience et de réfléchir aux pistes d'amélioration pour l'organisation d'un prochain atelier Zeste de Code ! Si vous êtes intéressés pour organiser vous-même un atelier de programmation près de chez vous, nous vous mettons à disposition un petit guide à la fin de l'article, inspiré de cette première expérience.

## 1. Un premier atelier à l'organisation... tendue

### 1.1. Chronologie de l'atelier et de son organisation

#### 1.1.1. Genèse de ce premier atelier

Fin juillet 2018, @Dwayn publie [un article](#) [↗](#) portant sur l'organisation par l'association Zeste de Savoir d'un événement [IRL](#) autour de la programmation. Cet article propose notamment de profiter de la semaine européenne du code pour décrocher une subvention à l'organisme Meet and Code et organiser un premier atelier de programmation pour débutants. Très vite, de

## 1. Un premier atelier à l'organisation... tendue

nombreux agrumes s'intéressent au projet, et se regroupent sur [le sujet dédié](#) ; il s'en suit une pulpeuse discussion où plusieurs idées émergent au fil des commentaires.

Avoir des idées, c'est bien ; avoir du concret, c'est mieux ! Alors que nous sommes début septembre, et que la date limite pour déposer la demande de subvention approche à grand pas, la communauté n'a toujours rien décidé de concret. @firm1 [réveille un peu le sujet](#) et quelques membres s'activent ; en quelques jours, @Bibou trouve une salle pour organiser l'événement au sein de son université. Avec l'aide de @Situphen, @Amaury et @entwanne, une demande de subvention est rédigée et envoyée ; deux jours après, la réponse tombe : 500 € de subvention nous sont accordés !

Cette subvention en poche, il nous faut terminer de régler l'organisation ; @Bibou s'en charge comme un chef en allant effectuer toutes les démarches auprès de son université pour obtenir une salle informatique le **24 octobre**, et ainsi permettre d'organiser cet atelier très attendu ! Il s'occupe aussi de contacter des assurances pour l'association, afin d'être couverts le jour de l'événement, ainsi que d'imprimer des prospectus et des affiches puis de les distribuer dans plusieurs endroits à Orsay.

Neuf jours avant l'événement, les premières demandes d'inscription commencent à arriver dans notre boîte de courriel. @Situphen prépare donc un message-type personnalisable pour répondre au demande, ainsi qu'une fiche d'autorisation parentale et un tableur pour recenser tous les inscrits. Au jour le jour, @Situphen et @Amaury répondent aux différentes demandes. C'est dans les tous derniers jours que la liste des participants s'allonge pour arriver au nombre de quinze participants !

### 1.1.2. Et la date arriva

Nous voici quelques jours avant l'atelier et le plus important n'est pas prêt : les objectifs et les versions finales des jeux à proposer aux participants ! Pour le Flappy Bird, @Bibou s'occupe de rédiger les objectifs et @Situphen se familiarise avec Scratch ; pour le Snake, @TAlone fait un travail remarquable en concevant les différents objectifs à atteindre pour les participants et la version finale du jeu.

Le jour de l'évènement, un des organisateurs, @Bibou, est déjà sur place ; il s'est chargé des premières vérifications quant à la salle, en plus de s'être arrangé avec l'administrateur réseau pour avoir des sessions pour nos jeunes agrumes. Quelque temps avant l'évènement, @TAlone le rejoint et ils partent tous deux installer le fléchage dans les bâtiments, mais aussi depuis l'arrêt de bus. Une petite anecdote : tous deux venant en transports, ils n'en installent pas du côté du parking des voitures, ce qui ne sera pas sans déranger certains participants (ne sachant pas utiliser des coordonnées GPS par ailleurs, mais passons).

Après un court repas, les deux organisateurs déjà présents sont prêts à accueillir @Situphen ainsi qu'@Amaury, afin d'effectuer les derniers préparatifs ; en particulier, l'installation de Pip et de Pygame sur les postes informatiques. Les organisateurs terminent de réviser leurs copies quand les premiers participants arrivent et sont conviés à s'installer chacun à un poste de façon temporaire – l'organisation initiale prévoyant une salle réservée Python, et une salle réservée Scratch.

Quelques minutes après l'heure prévue, les organisateurs présentent rapidement l'association Zeste de Savoir et ses objectifs, et décrivent les objectifs de la programmation à ceux n'en ayant

## 1. Un premier atelier à l'organisation... tendue

jamais fait. Après cette présentation, l'atelier commence par une répartition des participants entre les deux langages proposés, selon les choix de chacun. À la surprise des organisateurs, un seul des participants choisi l'atelier Scratch ; le choix est donc fait de le laisser dans la même salle que les participants en Python : le chanceux aura droit à un professeur particulier durant toute la durée de l'atelier.

### 1.1.3. Déroulement de l'atelier

L'atelier commence donc pour nos pythonistes avec une initiation aux outils de programmation ([Spyder](#) ☞ aura été utilisé pendant cet atelier, car il était disponible sur les postes locaux). Très vite, les organisateurs enchaînent sur une présentation des bases du langage Python (structures conditionnelles, boucles, etc), qui durera plus de deux heures, alors que moins d'une heure devait y être allouée . Cette partie étant très importante, il était nécessaire, avant de s'attaquer à un projet plus complet, que les participants travaillent ces notions de bases, utiles pour toute réalisation informatique.

Ces deux heures passées, nos participants enchaînent sur le goûter, qui semble avoir vraiment plu à l'ensemble d'entre eux, étant un moment de détente après un début d'après-midi chargé. Pendant ce temps, les animateurs se relayent dans la salle pour installer les tuiles graphiques ainsi que la documentation des fonctions pour le Snake, oubliés lors de la préparation. Le jeu final est testé sur les postes informatiques, afin de valider son fonctionnement avant de faire remonter les participants.

L'heure de la pratique a enfin sonné ; alors qu'il ne reste plus qu'une heure et demie, les participants commencent à programmer un Snake en Python avec la bibliothèque fournie (disponible sur [le dépôt Zeste de Code](#) ☞ pour les curieux). Cette bibliothèque, écrite par @TAlone quelques jours avant l'atelier, censée simplifier le développement du jeu, a permis aux plus avancés de concevoir un jeu constitué d'un serpent bougeant à l'écran... sans pommes.

En effet, en raison du temps très court, et de la complexité ressentie de la bibliothèque (difficile d'écrire une abstraction pour débutants quand on ne l'est pas soi-même), nos participants n'ont eu le temps de terminer toutes les étapes prévues pour la réalisation du jeu. Malgré ces difficultés, et quelques contretemps qui n'ont que très peu impactés les participants directement, cette journée est ressentie positivement par la plupart d'entre eux, qui rentrent chez eux avec le sourire. Certains demandent même un retour pour pouvoir continuer le jeu chez eux... Demande que nous avons failli à survenir, ayant pris bien trop de temps à rédiger ce compte-rendu ainsi que celui des élèves. Plusieurs mois après, est-il décent de les recontacter avec de quoi continuer, alors que nous avons échoué, faute d'anticipation suffisante et de temps pris, à leur envoyer dans des délais raisonnables ? La question trouve pour la moment une réponse négative de notre côté.

Aucun problème n'a été posé lors du départ des participants, chacun étant parti plus ou moins à l'heure avec l'accompagnateur prévu. Il est alors temps pour l'organisation de ranger le goûter, ainsi que d'éteindre les postes informatiques. La nuit tombée, chacun rentre chez soi, et trois organisateurs, @Amaury, @Situphen et @TAlone font un débriefing rapide de la journée dans le RER ; débriefing qui, complété des retours de @Bibou, est l'ébauche de l'article que vous lisez en ce moment.

## 1.2. Quelques documents pour illustrer nos propos

i

Nous vous mettons à disposition plusieurs documents pour illustrer nos propos. Vous pouvez retrouver le code source complet du jeu Snake ainsi que les fichiers LaTeX utilisés pour la présentation de l'atelier sur notre dépôt Zeste de Code, tel qu'il était le jour de l'atelier [↗](#).

### 1.2.1. Répartition des âges des participants

Voici la répartition des participants (10 garçons et 3 filles), dont l'âge médian était 15 ans.

Répartition des âges des participants

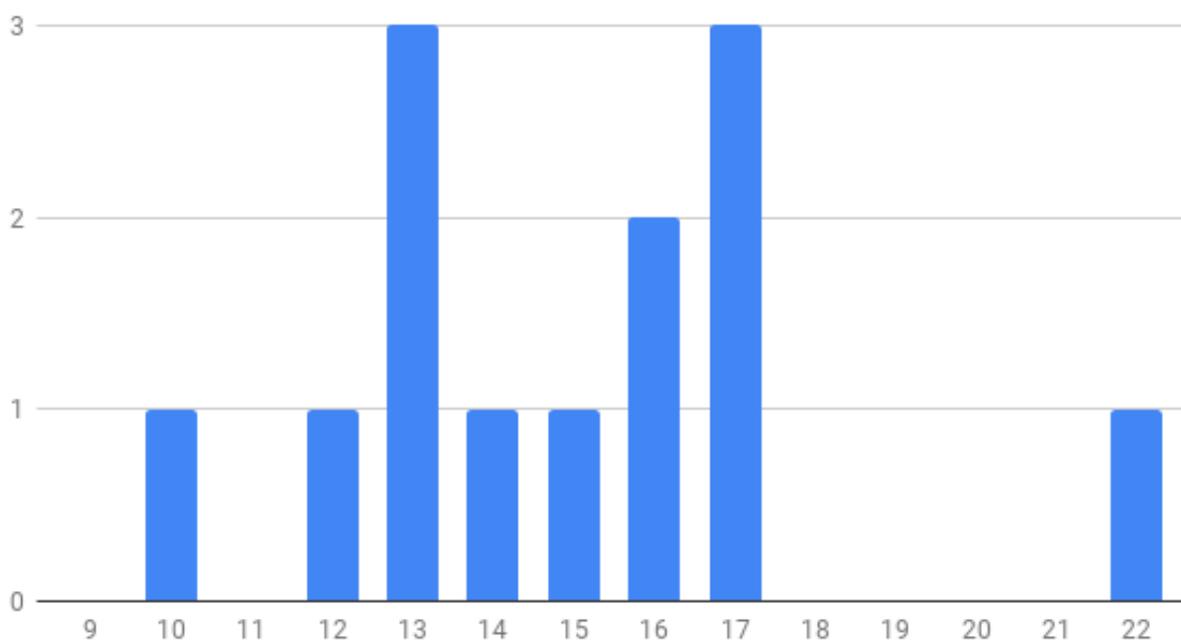


FIGURE 1.1. – Répartition des âges des participants

### 1.2.2. Version finale du jeu Snake

Voici le code source du jeu Snake (en balise secret) ainsi qu'une capture d'écran du jeu.

👁 Code source du jeu Snake

## 1. Un premier atelier à l'organisation... tendue

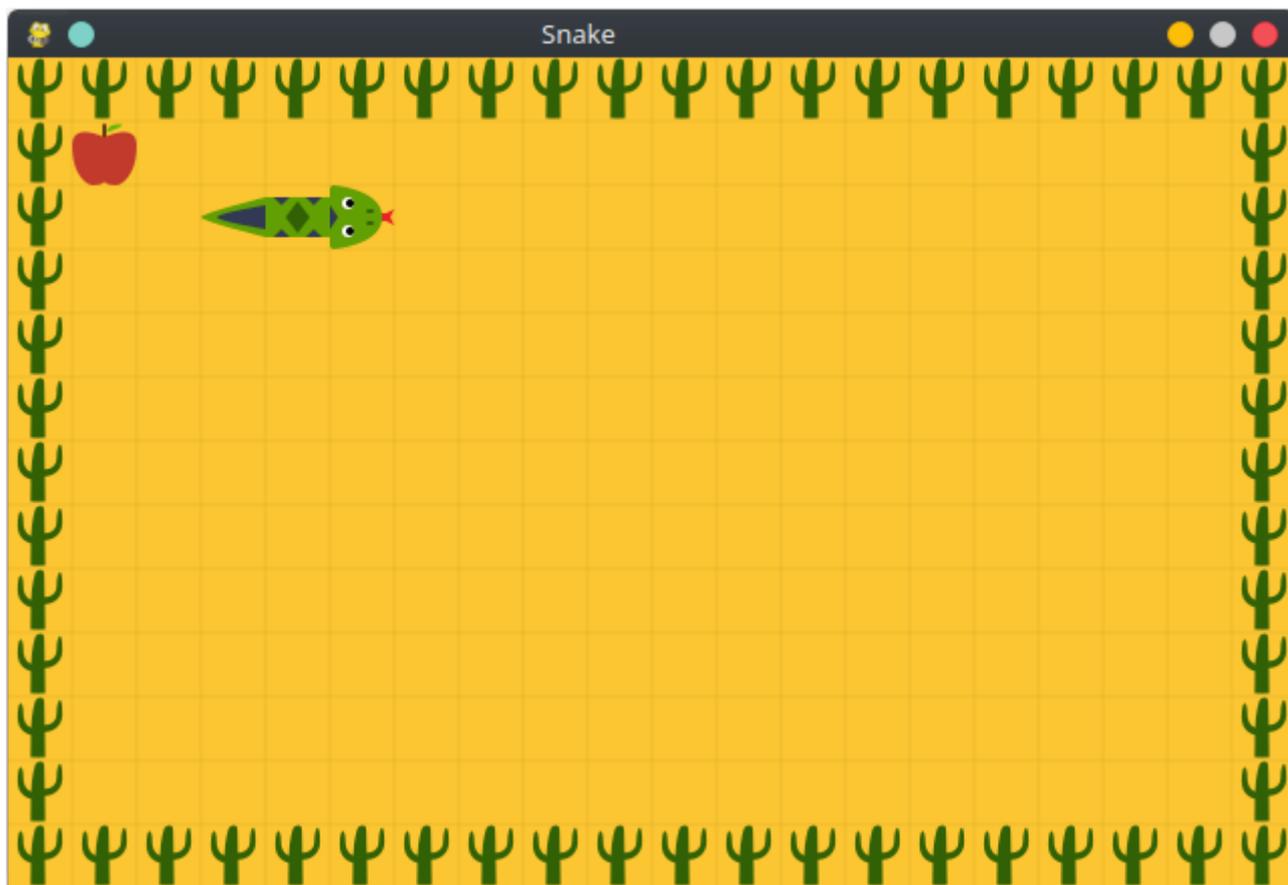


FIGURE 1.2. – Capture d'écran du jeu Snake

### 1.2.3. Affiche de l'événement

Voici l'affiche créée par @Bibou pour communiquer sur l'événement aux alentours d'Orsay, notamment grâce à la mairie.



# CODING GOÛTER

*Gratuit*

MERCREDI 24 OCTOBRE  
14h30 - 18h30 à ORSAY



Vous avez entre 12 et 24 ans ?

Apprenez à programmer  
en partant de zéro  
&  
Découvrez les études et  
métiers de l'informatique

Réservation obligatoire : [zestedecode@gmail.com](mailto:zestedecode@gmail.com)

Toutes les infos : [zestedesavoir.com](http://zestedesavoir.com)

- de 12 ans bienvenus si accompagnés  
Lieu : Université Paris Sud, PUIO bat. 640,  
rue Joliot Curie, 91400 Orsay  
Arrêt de bus université Paris Saclay



[zestedesavoir.com](http://zestedesavoir.com)  [@ZesteDeSavoir](https://twitter.com/ZesteDeSavoir)

Zeste De Savoir. Association à but non lucratif loi 1901



## 2. Une première expérience et déjà de nouvelles idées

### 2.1. Simplifier le code source en s'inspirant d'Arduino

Nous avons constaté lors de ce premier atelier que le code source du jeu est trop complexe pour des débutants. Malgré la mise à disposition aux participants d'un petit guide pour expliquer l'utilisation des fonctions, certaines instructions nécessaires au fonctionnement du programmation n'était pas bien comprises. Il y avait donc pour certaines parties un effet magique « Je ne sais pas ce que ça fait mais on me dit que c'est nécessaire donc je copie-colle. » qu'il faut éviter !

Nous allons retravailler le code source du jeu pour le simplifier, et nous allons nous inspirer du code source des programmes Arduino utilisés régulièrement dans l'enseignement. La structure de ces programmes est très simple, il s'agit de proposer deux fonctions :

- une fonction d'initialisation, dont le code est exécuté une seule fois au début du programme ;
- une fonction de boucle, dont le code est exécuté infiniment jusqu'à l'arrêt du programme.

Voici ce que ça donne :

```
1 void setup() {
2   // Une fonction d'initialisation
3 }
4
5 void loop() {
6   // Une fonction qui boucle
7 }
```

Notre code dans sa version minimale serait composé de ces deux seules fonctions. Lancé sans modification, il afficherait simplement une fenêtre vide. Chaque participant pourra alors créer au fur et à mesure la logique du jeu avec des fonctions créées par nos soins. Ce sont ces fonctions que nous allons retravailler pour les rendre plus compréhensibles. Nous allons aussi simplifier la gestion des événements.

En attendant d'arriver à une nouvelle version du jeu, nous pouvons d'or et déjà vous montrer le squelette du code source :

```
1 from zestedecode import Game
2
3 def setup(game):
4     # Une fonction d'initialisation
5     pass
6
7 def loop(game, events):
8     # Une fonction qui boucle
9     pass
10
```

## 2. Une première expérience et déjà de nouvelles idées

```
11 Game(setup, loop).play()
```

### 2.2. Configurer les logiciels et coder le jeu en français

Lors de ce premier atelier, les ordinateurs étaient configurés par défaut en anglais, et par conséquent il est en état de même pour les interfaces des logiciels. Aussi, les fonctions et les variables du code étaient en anglais. Nous avons constaté que l'utilisation de l'anglais est un frein pour de nombreux participants.

Il faudra donc lors du prochain atelier vérifier que les ordinateurs, ou, *a minima*, les logiciels, peuvent être configurés en français. Bien que l'utilisation de l'anglais pour le développement logiciel soit encouragé dans le monde professionnel, nous pensons que l'anglais n'a pas lieu d'être pour une initiation à la programmation avec un public adolescent. Une question reste en suspens, devons-nous utiliser des accents pour les variables et les fonctions en Python 3 ?

Nous pouvons reprendre la version minimale puis traduire les variables et les fonctions pour arriver à un code plus pédagogique :

```
1 from zestedecode import Jeu
2
3 def démarrage(jeu):
4     # Une fonction d'initialisation
5     pass
6
7 def boucle(jeu, événements):
8     # Une fonction qui boucle
9     pass
10
11 Jeu(démarrage, boucle).jouer()
```

### 2.3. Enseigner les bases de la programmation pendant la création du jeu vidéo

Nous en avons parlé dans la première partie de l'article, l'atelier ne s'est pas déroulé comme prévu : la présentation des bases du langage Python a duré deux heures au lieu d'une heure, ce qui n'a pas laissé assez de temps pour la programmation du jeu Snake. Faut-il réellement séparer l'apprentissage du langage et la conception du jeu vidéo ? Après cette première expérience, nous pensons que non. Nous allons réfléchir pour enseigner les bases du langage par la création du jeu vidéo. Si vous avez un avis sur la question, n'hésitez pas à nous en faire part !

### 3. Comment organiser un atelier de programmation ? Un petit guide sans prétention

#### 2.4. Commencer plus tôt la communication sur l'événement

Afin de faire connaître l'atelier, nous avons diffusé une petite campagne de communication dans les collèges et lycées des environs d'Orsay, ainsi que dans les rues à proximité et dans l'université qui nous accueillait. En soi, nous avons obtenus suffisamment d'inscriptions, mais ce point a jusqu'à la fin été assez tendu. En effet, suite à l'attente des subventions et, surtout, à celle de l'organisation de ce premier événement, nous avons lancé la campagne très, très tard : dix jours (!) avant l'atelier ; cette fois, nous avons eu de la chance, avec un pic d'inscription la veille (plus de la moitié de tous les inscrits, en fait), mais à l'avenir, il faudrait prévoir un peu plus large afin d'être tranquille.

L'idée de communiquer auprès de professeurs directement, et pas juste dans les lycées, a également été mentionnée. L'Éducation Nationale peut être un peu tatillonne à ce sujet, mais ça pourrait être une possibilité intéressante dans la limite du possible.

#### 2.5. Diriger les participants les plus inexpérimentés vers Scratch

Un seul participant a choisi l'activité utilisant Scratch. Probablement que la présentation de cette partie a été moins convaincante : il faudra veiller à ce que les deux présentations (Python et Scratch) soient aussi attrayantes l'une que l'autre.

### 3. Comment organiser un atelier de programmation ? Un petit guide sans prétention

*i*

Ce guide n'a pas la prétention de détenir la vérité ou d'être exhaustif sur le sujet ! Il décrit simplement la manière dont nous organiserons notre atelier de programmation la prochaine fois. Il garde ce qui a fonctionné lors du premier événement Zeste de Code et tire aussi les conclusions des petits couacs. Nous ne nous prétendons pas experts en la matière, soyez libres de suivre certains de nos conseils et d'en ignorer d'autres !

#### 3.1. En amont de l'événement

##### 3.1.1. Former une petite équipe motivée

- Deux ou trois, c'est peu. Un imprévu est vite arrivé et se retrouver tout seul pour animer un atelier de programmation, c'est difficile ! Il faut néanmoins rester à taille humaine ; plus on est nombreux et moins il est facile de communiquer.
- Vous êtes seul ? Nombreux sont les internautes prêts à vous rejoindre dans l'aventure et nombreuses sont les structures qui peuvent vous aider. Lancez-vous et créez un sujet sur le forum pour trouver des membres motivés ! Ouvrez la porte de votre mairie, de la médiathèque du quartier ou encore d'une association dédiée à l'informatique : ils pourront vous aider !

### 3. Comment organiser un atelier de programmation ? Un petit guide sans prétention

#### 3.1.2. Rechercher des subventions

- Organiser un événement est rarement gratuit, bien qu'il soit possible d'en minimiser les frais. De nombreuses organisations, tel Meet and Code par exemple, proposent des subventions pour l'organisation de projets éducatifs et l'atelier de programmation remplit souvent les critères. Profitons-en !
- Pour avoir une subvention, il est nécessaire de remplir un dossier décrivant l'événement proposé. Cela prend du temps et se prépare à l'avance, pensez-y !

#### 3.1.3. Définir la forme de l'événement

- À qui s'adresse l'événement ? Les ateliers de programmation sont souvent pensés et créés pour les enfants et les adolescents, mais rien n'empêche d'innover et d'en proposer à un public différent !
- Il faut ensuite trouver un lieu pour organiser l'événement, si possible avec des ordinateurs présents sur place pour simplifier l'organisation. Si vous êtes lycéen, étudiant ou enseignant, vous pouvez demander à votre établissement s'il est possible d'utiliser une salle d'enseignement sur un créneau.
- Vous devez vous mettre d'accord sur une date et un créneau horaire pour l'événement. Cela dépend bien entendu du lieu où vous organisez l'atelier. Il faut choisir un créneau pendant lequel l'équipe organisatrice et les participants seront présents ! Pour des enfants ou des adolescents, il faut penser aux temps extrascolaires : le mercredi après-midi, le samedi, le dimanche ou les vacances scolaires.
- Il faut aussi définir une durée ! Ni trop court pour que les participants aient le temps d'apprendre de nouvelles choses ; ni trop long pour laisser le cerveau se reposer. N'hésitez pas à prévoir une pause au milieu de votre événement, voire même plusieurs selon les envies ! Vous pouvez aussi réfléchir à organiser l'événement sur plusieurs jours consécutifs ou de manière récurrente sur plusieurs semaines.
- Il faut définir un nombre limite de participants qui soit en accord avec les capacités du lieu mais aussi le nombre d'organisateurs présents pour animer l'atelier.

#### 3.1.4. S'occuper des formalités

- Il est nécessaire d'avoir une assurance pour couvrir les accidents pouvant se produire, qu'il s'agisse de personnes ou de matériel. Dans le cadre des événements Zeste de Code, l'assurance est prise en charge par l'association Zeste de Savoir.
- En fonction du lieu que vous avez trouvé, il y aura peut-être des formalités administratives avec des papiers à signer. En cas de doute, demandez au responsable des locaux !

#### 3.1.5. Communiquer sur l'événement

- Pour que les participants viennent à l'événement, il faut les prévenir. Rien de bien nouveau, mais c'est un point qu'il ne faut pas prendre à la légère ! Vous pouvez contacter les journaux locaux pour qu'ils diffusent votre événement à leurs lecteurs. Vous pouvez imprimer des affiches et les coller sur les panneaux d'affichage libre. Vous pouvez demander à la mairie, aux établissements scolaires ou aux maisons de quartier de parler de votre événement à des participants potentiels ; n'hésitez pas à leur laisser des affiches. Enfin,

### 3. Comment organiser un atelier de programmation ? Un petit guide sans prétention

vous pouvez utiliser les réseaux sociaux en diffusant l'événement sur des pages ou des groupes dédiés à la ville ou aux alentours de l'événement.

- À chaque fois que vous communiquez autour de l'événement, il faut une courte description de l'événement, le lieu, la date, l'heure et un moyen de contact. Pour ce dernier, le mieux est probablement d'opter pour une adresse de courriel. Concernant le lieu, n'hésitez pas à préciser les transports en commun les plus proches du lieu !
- Attention, un événement ne se diffuse pas rapidement ! Il faut laisser le temps aux participants potentiels de prendre connaissance de celui-ci et de s'organiser pour pouvoir y participer.

#### 3.1.6. Préparer l'atelier de programmation

- Quel atelier de programmation souhaitez-vous mettre en place ? Pour le premier Zeste de Code, nous avons choisi de proposer la création d'un jeu Snake en Python ou la création d'un jeu similaire à Flappy Bird en Scratch. Le choix du langage est un point très important, certains langages sont plus accessibles au débutants que d'autres. Par exemple, prendre un langage comme le langage C demande au préalable d'expliquer les différents types de variable, cela peut prendre du temps et être une barrière de plus à franchir pour les participants qui débutent ! Le langage ne fait pas tout, le choix du jeu est important. Dans l'idéal, il faut permettre aux participants d'avoir une version jouable du jeu à la fin de l'atelier.
- Si vous comptez utiliser les ordinateurs présents sur le lieu de l'événement, il faut bien penser à vérifier que tous les logiciels nécessaires sont déjà installés ou s'installeront facilement le jour de l'atelier !
- Il est intéressant d'avoir une liste d'objectifs à atteindre dans la création du jeu. Libre à vous de garder cette liste pour les animateurs de l'atelier ou de la donner à chaque participant pour les laisser en autonomie. Cette liste permet de guider les participants étape par étape jusqu'à avoir une version jouable du jeu. Tous les participants n'avancent pas au même rythme, prévoyez quelques petits bonus à ajouter pour personnaliser le jeu !
- Chaque animateur devrait prendre le temps de s'approprier le jeu avant l'atelier. Plus vous connaissez le jeu, mieux vous pourrez aider les participants ! Aussi, essayez de vous placer dans la peau des participants et anticiper leurs difficultés éventuelles. Anticiper la difficulté d'un jeu n'est pas simple, n'hésitez pas à demander à des personnes de votre entourage de le tester et de vous faire un retour.
- Organiser un atelier de programmation, ce n'est pas seulement coder ! Il faut prévoir le déroulement de la journée : l'accueil des participants, la présentation de l'atelier, l'atelier en lui-même, la conclusion de l'atelier puis le départ des participants. N'hésitez à écrire un planning prévisionnel en estimant la durée que chaque moment de l'événement ! Cela vous permettra à tout moment de savoir si vous êtes plutôt en avance ou en retard sur vos estimations. Vous trouverez plus d'informations dans la partie "Le jour de l'événement".

#### 3.1.7. Recueillir les demandes de participation

- Vous allez progressivement recevoir des demandes de participation. Pour ne pas réinventer la roue à chaque fois, il peut être intéressant de créer un modèle-type pour les réponses, sans tomber dans la réponse automatique bien entendu !
- Pour les participants mineurs, il faut demander une autorisation parentale aux responsables légaux autorisant la participation à l'atelier de programmation. Vous pouvez

### 3. Comment organiser un atelier de programmation ? Un petit guide sans prétention

proposer aux participants de vous envoyer cette autorisation remplie et signée par courriel ou de vous la donner le jour l'événement, le choix est le vôtre !

- À la fin de l'événement, certains participants mineurs voudront peut-être partir seul. Pour éviter tout problème, il peut être intéressant de demander au préalable aux responsables légaux s'ils autorisent la sortie non accompagnée. Cela peut se rajouter simplement dans l'autorisation parentale.
- Lors de l'événement, vous serez peut-être amenés à prendre des photos. Si vous voulez diffuser ces photos et que les visages des participants apparaissent dessus, il vous faut leur autorisation pour ne pas violer leur droit à l'image. Sinon, vous pouvez tout simplement masquer les visages avant de les diffuser.
- En cas d'accident, notamment pour un participant mineur, il est très utile de pouvoir contacter un responsable légal. Le courriel de réponse est l'endroit idéal pour demander un numéro en cas d'urgence.
- Si jamais vous organisez un goûter pendant l'événement, n'hésitez pas à demander aux participants leurs allergies alimentaires. Là encore, le courriel de réponse est le moment idéal.
- Il est intéressant de tenir une liste des participants avec leur nom, leur âge, leur numéro de contact en cas d'urgence et leurs allergies alimentaires. Vous pouvez aussi noter pour les participants mineurs si vous avez reçu l'autorisation parentale et s'ils ont l'autorisation de sortir sans accompagnateur. Pensez à l'apporter le jour de l'événement !

## 3.2. Le jour de l'événement

### 3.2.1. Préparer les lieux

- Venez en avance ! Si l'atelier commence en début d'après-midi, préférez arriver sur place en fin de matinée : vous avez des choses à préparer avant le début de l'atelier. Enfin, rappelez-vous qu'un imprévu peut toujours pointer le bout de son nez !
- Allumez autant d'ordinateurs qu'il y a besoin, en fonction du nombre de participants inscrits. Installez les logiciels dont vous avez besoin ou vérifiez qu'ils fonctionnent correctement. Dans l'idéal, un participant devrait être en mesure de s'asseoir et commencer à coder tout de suite ! Si vous avez prévu de projeter un diaporama, profitez-en pour les ouvrir sur un ordinateur et allumer le vidéoprojecteur.
- Si le lieu où se déroule l'atelier est un labyrinthe, balisez avec des affiches les trajets que les participants sont les plus à même de prendre : de l'entrée du bâtiment jusqu'à la salle ; de l'arrêt de bus jusqu'au bâtiment ; du parking jusqu'au bâtiment. Dans la mesure du possible, essayez d'imprimer les affiches avant le jour de l'atelier, vous n'êtes jamais à l'abri d'une panne d'imprimante.

### 3.2.2. Animer l'atelier

- Très en avance, pile poil à l'heure ou légèrement en retard, les participants vont arriver au compte-goutte. Lors de l'accueil, notez au fur et à mesure les participants sur une feuille de présence. Pour les participants mineurs et si ce n'est pas déjà fait, récupérez les autorisations parentales ainsi que les autorisations (ou interdictions) de sortie non accompagnée. Si vous voulez prendre des photos, il faut aussi penser aux autorisations de droit à l'image. Une fois que la paperasse est faite, c'est l'occasion de discuter avec les participants avant le début de l'atelier !

### 3. Comment organiser un atelier de programmation ? Un petit guide sans prétention

- Tout le monde (ou presque) est arrivé ? C'est l'heure de la présentation ! Qui êtes-vous ? Quel est cet atelier ? Voici les deux questions auxquelles vous devrez répondre, avec un petit diaporama en support si vous le souhaitez.
- Il est maintenant de temps de mettre en place cet atelier longuement préparé ! Donner des conseils concrets peut être difficile tant un atelier peut prendre des formes différentes, néanmoins voici quelques questions d'ordre générale que vous pouvez vous poser ainsi que des éléments de réponse :
  - Est-ce que nous respectons le planning prévisionnel ? Le regarder chaque minute est inutile mais le faire chaque demi-heure environ permet d'anticiper le retard (ou l'avance) sur le programme !
  - Est-ce que nous sommes assez dynamique ? Si l'atelier se passe dans une salle de classe ou un pièce semblable, il faut éviter de rester près du tableau ou de la projection du diaporama si vous en avez un. Dans un événement comme celui-ci, vous voudrez probablement vous détacher de l'enseignement magistral. Ce n'est pas toujours simple surtout si vous avez suivi des études avec des enseignements magistraux, mais ça vaut le coup de tenter !
  - Est-ce que chaque participant comprend ? Un participant ne comprenant pas une partie de l'atelier ne va pas forcément se manifester, c'est donc à vous de repérer les participants en difficulté ! Vous pouvez par exemple passer dans les rangs pour questionner un à un les participants et les aider.
  - Si vous vous apercevez que tous les participants bloquent sur un concept ou une partie de l'atelier, il peut être judicieux de faire une explication générale pour le groupe. Si vous vous apercevez que l'atelier est trop compliqué, n'hésitez pas à l'adapter au mieux en cours de route !
- Si votre atelier dure plusieurs heures, par exemple de 14h à 18h, n'hésitez pas à prévoir une pause au milieu ! Cela permettra aux participants de laisser reposer leur cerveau et leurs yeux ainsi qu'à tout le monde de discuter d'une manière informelle. Vous pouvez aussi en profiter pour leur offrir un goûter.
- L'événement approche de sa fin, c'est le moment de conclure ! N'hésitez pas à valoriser le travail de chacun des participants pendant cet atelier. Vous pouvez profiter de ce temps pour distribuer un questionnaire et avoir un retour des participants sur le déroulement et le contenu de la journée. Si vous avez des autocollants ou des dépliants, n'hésitez pas à les mettre à disposition sur une table pour que les participants en prennent en partant. Par ailleurs, si vous avez des participants mineurs, gardez en tête qui a le droit de sortir non accompagné et qui doit attendre ses parents !

#### 3.2.3. Ranger les lieux

- Après avoir préparé les lieux puis animé l'atelier, c'est l'heure du rangement ! Rien de bien nouveau certes, mais c'est quand même à mettre dans l'emploi du temps de la journée. Aussi, n'oubliez pas d'enlever les affiches utilisées pour baliser le trajet si vous en avez utilisé !
- Enfin, vous pouvez prendre le temps entre animateurs de discuter du déroulement de la journée, ce qui s'est bien ou moins bien passé. Nous vous conseillons de noter brièvement ce dont vous discutez pour ne rien oublier !

### 3.3. Après l'événement

- Après la petite discussion entre animateurs le soir de l'événement, vous pouvez aller plus loin en témoignant de votre expérience et de vos idées sur un papier, dans une vidéo ou encore dans un article comme celui-ci. Vous aurez ainsi une trace écrite bien rédigée et c'est l'occasion de créer de bonnes discussions sur le sujet !
- Ce n'est pas grand chose et ça fait toujours plaisir, n'hésitez pas à envoyer un courriel de remerciements aux participants ! Si vous ne l'avez pas fait le jour de l'événement, c'est la dernière occasion de leur donner un petit questionnaire pour connaître leur avis sur l'atelier !
- Enfin, vous pouvez orienter les participants pour qu'ils puissent continuer leur apprentissage chez eux ! N'hésitez pas à leur concocter un petit guide pour qu'ils puissent refaire l'atelier chez eux sur leur ordinateur. En complément, vous pouvez leur donner les liens de plusieurs tutoriels pour débutants sur le même thème que l'atelier.

---

Vous avez pu le constater, beaucoup de pistes d'amélioration sont proposées après ce premier atelier et nous ne comptons pas nous en arrêter là. N'hésitez pas à en débattre ou en proposer de nouvelles dans les commentaires, car ces événements sont avant tout communautaires et nous sommes à l'écoute des conseils !

Suite au succès de ce premier atelier, nous réfléchissons déjà à l'organisation de prochains ateliers en région parisienne. Vous souhaitez rejoindre notre équipe de valeureux organisateurs ? Organiser un événement près de chez vous ? Manifestez vous sur [le sujet dédié](#) !

Nous remercions la communauté pour son soutien durant l'organisation de cet événement ! Un grand merci aussi à @informaticienzero pour la validation de cet article.

## Contenu masqué

### Contenu masqué n°1 : Code source du jeu Snake

[Voir le fichier sur Github](#)

```
1 #####
2 ###      Corrigé complet Snake      ###
3 ###      (c) Zeste de Savoir (c)     ###
4 ###      Licence GPL                 ###
5 ###      Auteur : TAlone             ###
6 #####
7
8 # Import de la bibliothèque pygame (gestion des graphismes)
9 import pygame
10 # Import de la bibliothèque du coding gouter (fonctions
    d'abstraction)
```

```
11 import cg
12
13 # Tout le code à écrire sera là-dedans
14 def main():
15     # Création d'une instance de jeu
16     jeu = cg.Game()
17     # Création d'un serpent à l'écran
18     snake = cg.Snake()
19
20     # Déclaration des sprites
21     jeu.addAsset("Cactus", "assets/cactus.png")
22     # Ajout du corps
23     jeu.addAsset("Corps", "assets/body.png")
24     # Ajout queue et tête
25     jeu.addAsset("Queue", "assets/tail.png")
26     jeu.addAsset("Tête", "assets/head.png")
27     # Ajout de la pomme
28     jeu.addAsset("Pomme", "assets/apple.png")
29
30     # Ajout de l'asset spécial qui sera automatiquement mis en
31     # fond
32     jeu.addAsset("background", "assets/background.png")
33
34     # Variable controlant l'ouverture de la fenêtre
35     ouvert = True
36     # Par défaut, le serpent ne doit pas bouger
37     direction = snake.DIRECTIONS.STOP
38
39     # Fonction pour faire apparaitre la pomme
40     def spawnPomme():
41         # Stockage de la position de la pomme
42         pomme = jeu.randomPosition()
43
44         # Tant que la pomme est hors-zone, on la change de
45         # place
46         while jeu.isSide(pomme):
47             pomme = jeu.randomPosition()
48
49             # Ne pas mettre la pomme sur le serpent
50             for part in snake.partsIterator():
51                 if jeu.collission(pomme,
52                                     part.position):
53                     # Tant que la pomme est sur
54                     # le joueur, on la change
55                     # de place
56                     pomme =
57                         jeu.randomPosition()
58
59             # Retourne la position trouvée
60             return pomme
```

```
55
56     # Ajout de la pomme
57     pomme = spawnPomme()
58
59     # Boucle principale du jeu
60     while ouvert:
61         # Récupération des événements de la file d'attente
62         for event in pygame.event.get():
63             # Si l'utilisateur veut quitter...
64             if event.type == pygame.QUIT:
65                 # ...sortie de la boucle
66                 # principale, et fermeture du
67                 # jeu.
68                 ouvert = False
69             # Lorsqu'une touche est appuyée...
70             if event.type == pygame.KEYDOWN:
71                 # ...on vérifie la direction
72                 # droite...
73                 if event.key == pygame.K_RIGHT:
74                     direction =
75                         snake.DIRECTIONS.RIGHT
76                 # ...et les autres directions.
77                 elif event.key == pygame.K_UP:
78                     direction =
79                         snake.DIRECTIONS.TOP
80                 elif event.key == pygame.K_DOWN:
81                     direction =
82                         snake.DIRECTIONS.BOTTOM
83                 elif event.key == pygame.K_LEFT:
84                     direction =
85                         snake.DIRECTIONS.LEFT
86
87             # Effacement de l'écran, et remplissage avec les
88             # tiles de fond
89             jeu.eraseScreen()
90
91             # Dessin de tous les cactus
92             for tile in jeu.screenIterator():
93                 if jeu.isSide(tile):
94                     jeu.draw(jeu.asset("Cactus"), tile)
95
96             # Dessin du serpent à l'écran
97             for part in snake.partsIterator():
98                 # Choix du sprite en fonction de la partie
99                 # à dessiner et rotation
100                if part.type == snake.PARTS.HEAD:
101                    # Rotation de la tête
102                    sprite = jeu.asset("Tête",
103                                        part.rotation)
104                elif part.type == snake.PARTS.TAIL:
```

```
95         sprite = jeu.asset("Queue",
96                               part.rotation)
97     else:
98         sprite = jeu.asset("Corps",
99                               part.rotation)
100
101     # Dessin du sprite à l'écran
102     jeu.draw(sprite, part.position)
103
104     # Dessin de la pomme
105     jeu.draw(jeu.asset("Pomme"), pomme)
106
107     # Ne pas oublier de terminer la phase de dessin
108     jeu.end()
109
110     # Déplacement du serpent
111     snake.move(direction)
112
113     # Si il y a collision entre la pomme et la tête du
114     # serpent
115     if jeu.collision(snake.getHeadPosition(), pomme):
116         # Le serpent grandit
117         snake.grow()
118         # La pomme change de position
119         pomme = spawnPomme()
120
121     # Vérifie que le serpent soit toujours à l'écran
122     for tile in jeu.screenIterator():
123         if jeu.isSide(tile) and
124             jeu.collision(snake.getHeadPosition(),
125                           tile):
126             # Ferme le jeu si collision avec un
127             # mur
128             ouvert = False
129
130     # Vérifie que le serpent ne se mord pas
131     for part in snake.partsIterator():
132         # Attention : la tête est incluse, il faut
133         # donc la retirer
134         if jeu.collision(snake.getHeadPosition(),
135                           part.position) and part.type !=
136             snake.PARTS.HEAD:
137             # Ferme le jeu si le serpent se
138             # rentre dedans
139             ouvert = False
140
141 # Lancement de main seulement si le fichier n'est pas importé comme
142 # module
143 if __name__ == "__main__":
144     main()
```

*Contenu masqué*

[Retourner au texte.](#)

# Liste des abréviations

**IRL** In Real Life ou « dans la vie réelle » par opposition à la vie numérique.. 1