
Défi

Jeu vidéo textuel

Programmation Scratch



DÉFI – PROGRAMMATION SCRATCH

Objectifs : Poser et résoudre des problèmes en créant un programme à l'aide d'un langage de programmation par bloc

Progressions :

- EN32 : Créer de courts programmes avec des boucles et/ou des conditions à l'aide d'un langage de programmation visuel
- EN32 : Créer un programme optimisé
- EN32 : Documenter son programme

Planification

Période	Description
Périodes 1-2	Découverte de la Plateforme Scratch & Présentation du projet « Jeu vidéo textuel »
Périodes 3-4	Choix des objets (Sprite + arrière-plan personnalisé) + Affichage des éléments Choix des textes + séquençage du dialogue
Périodes 5-6	Programmation des interactions Optimisation / Finalisation
Périodes 7-8	Présentation du jeu + Discussions autour des difficultés, des améliorations possibles

1. Découverte de l'environnement Scratch

→ Par table : 1 iPad connecté à Scratch
+ 1 iPad pour afficher le support de cours ci-dessous :



Création d'un compte Scratch : pour réaliser ce défi, tu auras besoin de créer un compte Scratch. Pour cela, prends un iPad et :

1. Connecte-toi à ton compte Outlook
2. Suis les instructions de l'email « Activités Scratch – 08.12.22 »
3. N'oublie pas d'écrire tes identifiants de connexion sur ton Pass Numérique !

DÉFI – PROGRAMMATION SCRATCH

2. Présentation du défi

Un jeu collaboratif

But : créer un jeu d'énigme sur Scratch

Durée : 3x2 périodes

Histoire : le héros de notre histoire a une quête à accomplir : retrouver le trésor caché de Cugy. Pour atteindre son but, il devra passer un certain nombre d'épreuves (énigmes)

Modalités :

- Par équipe de 2 (*par table*)
- Chaque groupe crée une scène (un tableau) sur Scratch contenant :
 - Un arrière-plan personnalisé
 - Un Sprite personnalisé
 - Le personnage principal

Le personnage principal





Notre héros est un sympathique personnage bleu qui adore les aventures ! Il se balade de lieux en lieux, à la recherche d'énigmes à résoudre.
Attention : il n'aime pas perdre. Ça le met très en colère !

Exemple :



<https://scratch.mit.edu/projects/727514072/>

Partage : pour réaliser cette quête, nous allons tous partager le même espace de développement (studio). Pour cela, prends l'iPad connecté à Scratch et :

1. Clique sur le  dans la barre du haut
2. Clique sur le titre de ton projet pour :
 - a. Le renommer : `DefiJVT_[Prenom1][Prenom2]`
 - b. Indiquer les instructions d'utilisation
 - c. Partager ton projet
3. Retourne dans ton  et sélectionne « Mes Studios »
4. Clique sur le Studio de ta classe
5. Sélectionne « Parcourir les projets » pour trouver ton projet

3. Choix des objets

L'arrière-plan

1. Trouver le parfait arrière-plan pour ta scène

Va te balader dans le bâtiment avec l'iPad pour prendre la photo d'un lieu. Tu utiliseras ensuite cette photo pour l'arrière-plan de ta scène Scratch

2. Partager ta photo

Pour partager ta photo avec l'iPad connecté sur Scratch, tu peux :

- Faire un Air Drop
- L'envoyer par email grâce à Outlook
- Autre solution de ton choix

3. Importer ton arrière-plan

Une fois que la photo se trouve sur l'iPad connecté sur Scratch, il faut l'importer dans ton projet

Les Sprites

1. Trouver le héros

Tera est le héros de notre histoire. Tu dois donc commencer par le trouver dans la bibliothèque des Sprites Scratch. N'hésite pas à parcourir ses costumes pour voir toutes ses expressions !



2. Choisir ton Sprites personnalisé

Dans ta scène, Tera va rencontrer un personnage qui va lui proposer une énigme. Tu peux créer ce personnage ou le choisir dans la bibliothèque des Sprites Scratch.

3. Poser le décors

Il ne te reste plus qu'à disposer tous tes éléments pour que ton histoire prenne vie ! Tu peux rajouter des éléments de décors ou personnaliser tes Sprites (taille, couleurs, accessoires, etc.).

Le dialogue

1. Ne perds pas ton temps avec le dialogue !

L'important dans ce défi, c'est l'interaction entre les personnages et l'utilisateur du programme (par l'intermédiaire du clavier). Tu peux donc partir sur un dialogue très simple, que tu affineras à la fin du projet s'il te reste du temps.

Exemple de dialogue :

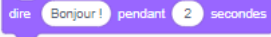



- Bonjour. Je vais te poser une énigme
- Attends que je termine de parler pour répondre
- Selon toi, est-ce que les araignées sont des insectes ?

DÉFI – PROGRAMMATION SCRATCH

4. La programmation

1. Le dialogue entre les personnages

Tu peux faire interagir les personnages entre eux. Pour cela, il existe plusieurs solutions :

- Jouer sur le timing en utilisant  et 
- Utiliser un déclencheur de type événement  et 

2. L'interaction avec le clavier

Tu dois trouver un moyen de faire interagir le joueur dans l'histoire. Par exemple :

- Lorsque le joueur appuie sur la touche « o », cela signifie « oui »
- Lorsque le joueur appuie sur la touche « n », cela signifie « non »

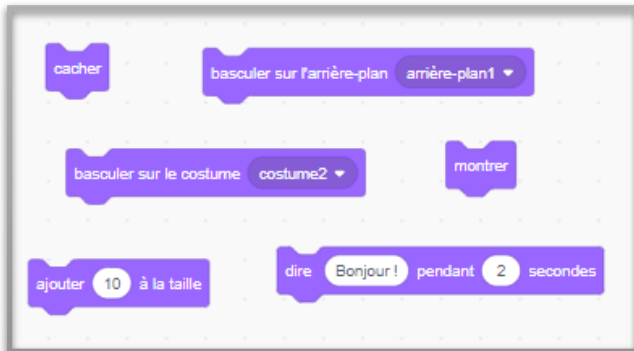
3. Faire vivre l'histoire !

Tu peux jouer avec les costumes, les déplacements, les tailles... pour rendre ton histoire plus vivante

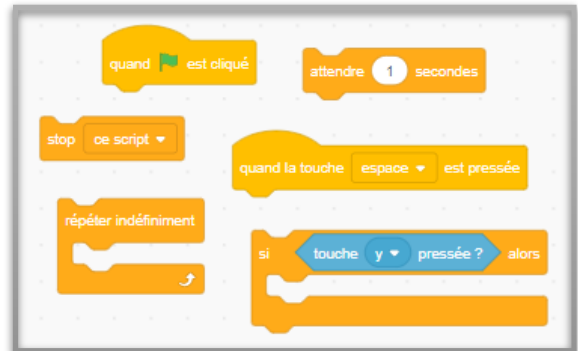
Astuce n°1

Boîte à outils

Tu auras besoin de ces éléments. Tu n'es pas obligé de tous les utiliser.



A utiliser pour qu'il se passe quelque chose



A utiliser pour gérer les choix

Astuce n°2

Remise à zéro

Une chose importante à laquelle tu dois penser est la « remise à l'état initial » de tous les éléments qui changent durant l'exécution de ton programme (attention de bien le faire pour chacun de tes Sprites !)



DÉFI – PROGRAMMATION SCRATCH

Auto-évaluation

Astuce n°3

Optimiser

Ton programme marche ? Bravo !

Prends le temps de le parcourir encore une fois pour t'assurer que toutes les lignes sont utiles, qu'il n'y a pas des répétitions en trop, que tu ne peux pas fusionner des blocs ensemble ou encore simplifier tes conditions...

Astuce n°4

Nettoyer

Avant de rendre ton projet et de t'auto-évaluer, il reste une chose importante à faire : le nettoyage de ton code ! Un bon programmeur ne laisse pas de bouts de code inutiles traîner.

Astuce n°5

Commenter

Tu viens de finir ton programme ? Tu peux être fière de toi !

Pour l'instant, tu arrives facilement à te relire car tu as encore ton raisonnement en tête. Mais dans 1 mois, quand tu voudras le rouvrir pour te remémorer comment tu avais fait, ton code te paraîtra peut-être plus difficile à déchiffrer. Commente-le dès maintenant pour en faciliter la relecture plus tard !

De plus, les programmeurs sont une communauté de personnes qui apprennent les uns des autres, qui s'entraident et coconstruisent. Il est donc très fréquent qu'un programmeur ouvre le programme d'un autre pour voir comment il a créé son code, comment il a contourné une difficulté, etc. En mettant des commentaires dans ton programme, tu lui permets de parcourir plus facilement ton code pour apprendre de ton expérience.

Objectifs - Je suis capable de...	SUR SCRATCH		
	Critères de Suffisance (note : 4/6)	Critères d'Aisance (note : 5/6)	Critères de Maîtrise (note : 6/6)
1. Créer un Sprite			
2. L'afficher correctement			
3. Lui faire dire du texte			
4. Choisir un Arrière-plan adéquat			
5. Le télécharger dans Scratch			
6. L'afficher correctement			
7. Trouver le personnage principal			
8. L'afficher correctement			
9. Lui faire dire du texte			
10. Faire interagir le héros en lui proposant un choix			
11. Proposer des événements différents selon le choix			
12. Faire évoluer le costume du Sprite			
13. Déplacer le Sprite			
14. Forcer le joueur à ne pouvoir parler qu'à la fin de la question			
15. Utiliser des concepts de codage avancés (boucle, condition)			
16. Eviter les répétitions dans le code			
17. Eviter les bugs dans le code			
18. Eviter les codes inutiles			

FICHE MÉMO

Les concepts

Instructions : ordre que je donne à une machine en utilisant un langage appelé : langage de programmation

Langage de programmation : langage compréhensible par l'homme et la machine

Algorithme : description d'une suite d'étapes permettant de résoudre un problème

Programme : algorithme exprimé dans un langage de programmation

Les commandes utiles en Scratch

Je déplace mon Sprite

avancer de 10 pas

aller à x: 0 y: 0

glisser en 1 secondes à x: 0 y: 0

ajouter 10 à x

ajouter 10 à y

Je joue avec mon Sprite

dire Bonjour ! pendant 2 secondes

penser à Hmm... pendant 2 secondes

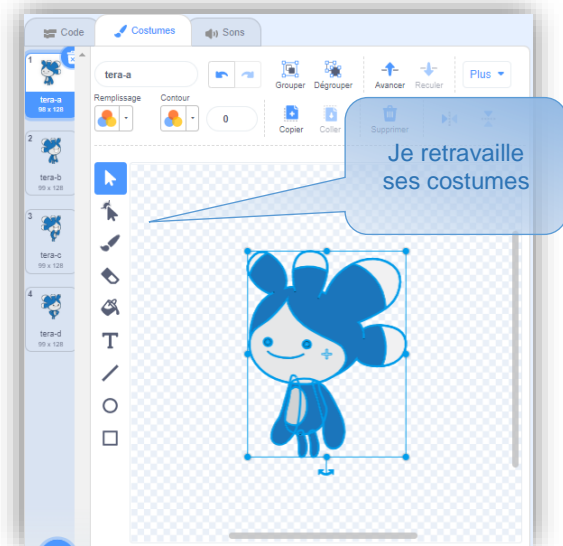
basculer sur le costume costume2

ajouter 10 à la taille

mettre la taille à 100 % de la taille initiale

montrer

cacher

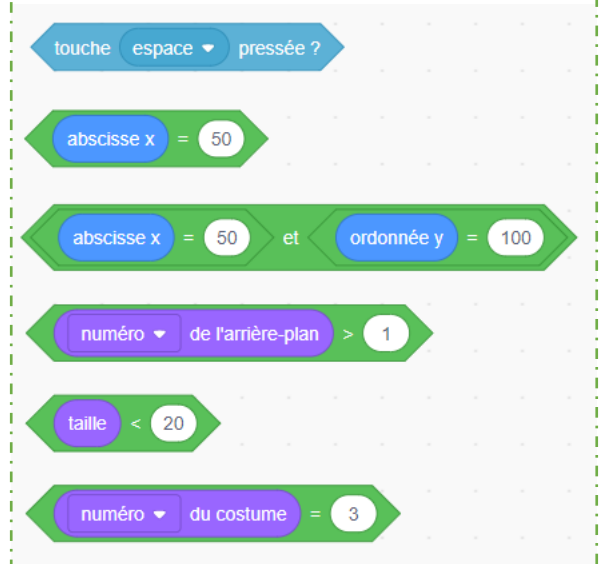


Les commandes utiles en Scratch

Je contrôle les événements



Je capte un événement

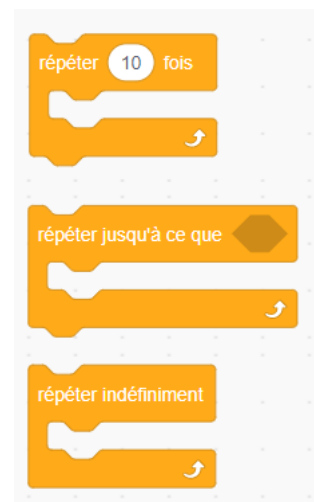


Je teste une condition



Mon programme teste ma condition puis passe à la prochaine instruction

Je répète jusqu'à une condition



Mon programme exécute mes instructions jusqu'à un événement de sortie de la boucle

BILAN

Auto-évaluation

Nom du projet Scratch : _____

Objectifs - Je suis capable de...	SUR SCRATCH		
	Critères de Suffisance (note : 4/6)	Critères d'Aisance (note : 5/6)	Critères de Maîtrise (note : 6/6)
1. Créer un Sprite			
2. L'afficher correctement			
3. Lui faire dire du texte			
4. Choisir un Arrière-plan adéquat			
5. Le télécharger dans Scratch			
6. L'afficher correctement			
7. Trouver le personnage principal			
8. L'afficher correctement			
9. Lui faire dire du texte			
10. Faire interagir le héros en lui proposant un choix			
11. Proposer des événements différents selon le choix			
12. Faire évoluer le costume du Sprite			
13. Déplacer le Sprite			
14. Forcer le joueur à ne pouvoir parler qu'à la fin de la question			
15. Utiliser des concepts de codage avancés (boucle, condition)			
16. Eviter les répétitions dans le code			
17. Eviter les bugs dans le code			
18. Eviter les codes inutiles			

→ Proposition d'un critère supplémentaire :

Note auto-évaluée par le binôme :

Note finale* :

* = Note de l'enseignante arrondie en fonction de l'auto-évaluation :

- Si l'auto-évaluation est bonne (à 0,5 point près) = note finale arrondie au demi-point supérieur
- Sinon = note finale arrondie au demi-point inférieur

Analyse du résultat

Difficultés rencontrées

Solutions trouvées

1. **L'arrière-plan** (le choisir, l'importer dans Scratch, ...)

--	--

2. **Les Sprites** (le trouver / choisir, modifier le costume, le faire bouger, ...)

--	--

3. **Le Dialogue** (choix des blocs, séquençage, ...)

--	--

4. **L'énigme** (affichage de l'énigme, prise en compte de la réponse avec le clavier, ...)

--	--

5. **Autres difficultés**

--	--

Améliorations possibles

1.

2.

3.

4.

LOGIQUE COMPUTATIONNELLE

Résolution de problèmes

Durant le projet, il était nécessaire de faire interagir le joueur à travers le clavier. Il devait ainsi pouvoir choisir entre 3 propositions de réponse à une énigme. Pour faire cela, plusieurs solutions étaient possibles. Nous allons décortiquer ensemble deux propositions de solution.

Solution n°1

La solution de Nora & Eva

Parcours ce programme pour comprendre comment il fonctionne.

```
quand est cliqué
  dire Eh! pendant 2 secondes
  dire Je vais te poser une question, si tu réponds faux je te manges! pendant 4 secondes
  attendre 4 secondes
  dire C'est quand la fondation de Rome? pendant 2 secondes
  dire a. -105 Av. JC b. -400 Av. JC c. 753 Av. JC pendant 8 secondes
  attendre jusqu'à ce que touche n'importe laquelle ▼ pressée?
  si touche c ▼ pressée? alors
    dire BRAVOOO!!!!
  sinon
    dire C'EST FAUX!!!!
    attendre 1 secondes
    dire JE VAIS TE MANGERRRRRR!!
```

→ Qu'est-ce que tu trouves malin dans ce programme :

Malheureusement, ce programme ne fonctionne pas toujours comme prévu avec le clavier. Pourquoi ?

→ Explique le problème de ce programme :

→ Explique la solution que tu proposerais pour résoudre ce problème :

La solution de Lula & Joy

Parcours ce programme pour comprendre comment il fonctionne.

```
quand cliqué
attendre 2 secondes
dire Bonjour ! pendant 1 secondes
attendre 4 secondes
dire Si tu veux passer tu dois répondre à une énigme pendant 3 secondes
attendre 3 secondes
dire Quelle est la superficie de Cugy ? pendant 3 secondes
attendre 2 secondes
dire A: 2,9 km2 / B: 2,88 km2 / C: 3,08 km2 pendant 0.5 secondes
attendre jusqu'à ce que
  touche a pressée ? ou touche b pressée ? ou touche c pressée ?
attendre 2 secondes
si touche a pressée ? alors
  dire Bonne réponse tu peux passer ! pendant 4 secondes
si touche b pressée ? ou touche c pressée ? alors
  attendre 2 secondes
  dire Mauvaise réponse ! pendant 2 secondes
```

→ Qu'est-ce que tu trouves malin dans ce programme :

Malheureusement, ce programme ne fonctionne pas toujours comme prévu avec le clavier. Pourquoi ?

→ Explique le problème de ce programme :

→ Propose la solution la plus maline pour que l'interaction avec le clavier fonctionne parfaitement :

(aide-toi des 2 programmes vus sur cette fiche)

```
dire Quelle est la superficie de Cugy ? pendant 3 secondes
attendre 2 secondes
dire A: 2,9 km2 / B: 2,88 km2 / C: 3,08 km2 pendant 0.5 secondes
```

Réflexions d'après-projet concernant l'évaluation

Retravail des critères d'évaluation :

1. **Respect de la consigne** : Les élèves ont-ils respecté la consigne donnée ? Ont-ils créé deux Sprites qui dialoguent et posent une énigme avec 3 propositions de réponse différentes ? (Note sur 1)
2. **Qualité du code** : Les élèves ont-ils utilisé des blocs de code pertinents pour résoudre le problème posé ? Leur code est-il lisible et bien organisé ? (Note sur 1)
3. **Interactivité** : Le projet est-il interactif ? L'utilisateur peut-il taper la réponse au clavier ? Les réactions des Sprites sont-elles bien synchronisées avec les actions de l'utilisateur ? (Note sur 1)
4. **Originalité** : Les élèves ont-ils fait preuve d'originalité dans la création de leur projet ? Ont-ils ajouté des éléments visuels ou sonores pour rendre leur projet plus intéressant ? (Note sur 1)
5. **Difficulté** : La difficulté de l'énigme posée est-elle adaptée aux capacités des utilisateurs ? L'énigme est-elle suffisamment complexe pour être intéressante, mais pas trop difficile pour être résolue par un élève de 12 ans ? (Note sur 1)
6. **Résolution de problème** : Les élèves ont-ils résolu le problème posé dans la consigne de manière efficace ? Leur projet est-il fonctionnel ? (Note sur 1)

? Evaluation graduelle ?

Critères de suffisance :

1. **Respect de la consigne** : Les élèves ont-ils respecté les consignes données ?
 - a. Ont-ils créé deux Sprites qui dialoguent
 - b. Ont-ils posé une énigme avec 3 propositions de réponse différentes
2. **Qualité du code** : Les élèves ont-ils utilisé des blocs de code pertinents pour résoudre le problème posé ?
3. **Interactivité** : Le projet est-il interactif ? L'utilisateur peut-il taper la réponse au clavier ?
4. **Fonctionnalité** : Le projet fonctionne-t-il sans problème majeur ?

Critères d'aisance :

1. **Attractivité** : Les élèves ont-ils ajouté des éléments visuels pour rendre leur projet plus intéressant ?
 - a. Déplacement des Sprites
 - b. Changement de costume
2. **Respect du thème** : L'énigme posée est-elle en rapport avec le thème donné ?
3. **Organisation du code** : Le code est-il lisible et bien organisé ?
4. **Esthétique** : Le projet est-il visuellement attrayant et agréable à regarder ?

Critères de maîtrise :

1. **Utilisation avancée des blocs** : Les élèves ont-ils utilisé des blocs de code plus avancés pour créer des fonctionnalités supplémentaires, allant au-delà de la consigne donnée ?
2. **Optimisation** : Le code est-il optimisé pour fonctionner de manière fluide et sans problème ?

→ évaluer les projets Scratch des élèves à différents niveaux de compétence

→ Les critères "de suffisance" évaluent la capacité des élèves à répondre aux exigences de base de la consigne, tandis que les critères "d'aisance" évaluent leur capacité à ajouter des éléments supplémentaires et à améliorer la qualité de leur projet. Les critères "de maîtrise" évaluent leur capacité à utiliser des fonctionnalités avancées et à créer des projets innovants et complexes. Cela permet de donner une évaluation plus juste et précise de leurs compétences en programmation Scratch.