

E C H O

CDSF



Par Team Nightberry

Version v1.0

10/01/2026

Par la Team Nightberry

TABLE DES MATIÈRES

- 1. Contexte et objectifs
 - 1.1. Présentation succincte du besoin initial
 - 1.2. Problématique à résoudre
 - 1.3. Objectifs principaux du projet
- 2. Périmètre du projet
 - 2.1. Ce que le projet couvre
 - 2.2. Hors périmètre
- 3. Parties prenantes et utilisateurs cibles
 - 3.1. Utilisateurs finaux
 - 3.2. Parties prenantes
- 4. Description générale du produit attendu
 - 4.1. Vision globale du fonctionnement
 - 4.2. Cas d'usage principaux
 - 4.3. Contraintes globales
- 5. Exigences fonctionnelles
 - 5.1. Liste des fonctions attendues (classées par thème)
 - 5.2. Description des fonctions (modèle de formalisation)
- 6. Exigences non fonctionnelles
 - 6.1. Performances attendues
 - 6.2. Fiabilité, sécurité, maintenabilité
 - 6.3. Ergonomie et accessibilité
 - 6.4. Contraintes réglementaires éventuelles
- 7. Hypothèses et dépendances
 - 7.1. Hypothèses de départ
 - 7.2. Dépendances
- 8. Critères d'acceptation et de validation
 - 8.1. Méthodes de vérification
 - 8.2. Indicateurs de réussite
- 9. Planning et estimation
 - 9.1. Grandes étapes
 - 9.2. Jalons importants
 - 9.3. Estimation des délais
 - 9.4. Estimation prévisionnelle des coûts

1. Contexte et objectifs

1.1 Présentation succincte du besoin initial

Notre objectif est de développer un jeu vidéo 2D de type Metroidvania, offrant une expérience d'exploration immersive dans un univers sombre et en constante évolution. Le jeu devra combiner mécanique de combat dynamiques, phase de plateformes et ambiance sonore originale renforçant l'atmosphère globale.

1.2 Problématique à résoudre

Comment proposer une expérience de jeu riche, cohérente et progressive dans laquelle l'environnement réagit aux actions du joueur et où l'exploration reste centrale ?

1.3 Objectifs principaux du projet

Le projet *Echo* poursuit plusieurs objectifs concrets et mesurables :

- Réaliser, avant la fin de l'année scolaire, un jeu d'action/plateforme fonctionnel et jouable avec des contrôles simples et efficaces, permettant au joueur de se déplacer, sauter et attaquer de manière fluide, sans bug bloquant.
- Mettre en place, au cours de l'année, un système de progression avec au moins 2 capacités déblocables (par exemple un dash ou un double saut ou autre), permettant d'accéder à de nouvelles zones du jeu, et vérifiable lors d'une démonstration finale.
- Créer un univers cohérent et immersif à travers un scénario simple, des décors évocateurs et des éléments narratifs intégrés au jeu (dialogues courts, éléments visuels), finalisés avant la présentation du projet.
- Intégrer, avant la version finale, une ambiance sonore originale comprenant des musiques et des bruitages, utilisés pour renforcer l'immersion et aider le joueur (par exemple pour signaler un danger ou un passage important).
- Développer une direction artistique en pixel art, inspirée de jeux connus (*Hollow Knight*, *Dead Cells*, *Terraria*) tout en conservant une identité personnelle
- Prévoir une coopération en ligne fonctionnelle, offrant la possibilité à plusieurs joueurs d'évoluer ensemble dans l'univers du jeu via un mode P2P.

2. Périmètre du projet

2.1 Ce que le projet couvre

Le projet *Echo* couvrira les éléments suivants :

- La conception complète d'un jeu 2D Metroidvania exploitant le moteur Python Pygame.
- Les mécaniques de déplacement et de combat (marche, saut, course, attaques).
- L'intégration de capacités évolutives (dash, double saut, attaque spéciale).
- Un level design interconnecté favorisant l'exploration non linéaire et la redécouverte de zones débloquées.
- La création originale d'une direction artistique en pixel art.
- Une bande-son atmosphérique composée majoritairement par l'équipe, mêlant effets sonores et ambiance.
- L'intégration d'un mode coopératif en ligne (P2P), permettant à plusieurs joueurs d'explorer ensemble.
- La réalisation d'un site web vitrine présentant le projet, ses mécaniques principales, et permettant le téléchargement du jeu.

2.2 Hors périmètre

Certaines fonctionnalités ne font pas partie du périmètre de développement :

- Mode multijoueur compétitif (battle, versus, etc.).
- Narration sous forme de dialogues parlés ou doublages audio.
- Cinématiques animées.
- Tout support autre que PC (les portages potentiels seront envisagés ultérieurement).

3. Parties prenantes et utilisateurs cibles

3.1 Utilisateurs finaux

Le public visé par *Echo* est composé des Joueurs amateurs de jeux d'aventure/plateforme, attachés à la progression, la difficulté maîtrisée et la narration environnementale. Il s'adresse aussi aux fans de Metroidvania et de jeux indépendants cherchant des expériences émotionnelles et immersives ainsi qu'aux joueurs adeptes d'ambiances sombres et narratives où chaque détail visuel et sonore renforce la cohérence du monde.

3.2 Parties prenantes

Les principaux acteurs impliqués dans le projet *Echo* sont l'équipe de développement (étudiants EPITA), responsable de la conception, du game design et du développement technique sous Pygame, les encadrants pédagogiques, des jalons de soutenance et de la qualité globale du projet. Nous pouvons aussi inclure les partenaires externes éventuels, pouvant contribuer aux domaines du son, du graphisme ou de la composition musicale (IA, proche, ...).

4. Description générale du produit attendu

4.1 Vision globale du fonctionnement

Echo est un jeu d'exploration en 2D, où le joueur incarne Kaelen, un ancien soldat évoluant dans un monde plongé dans une nuit perpétuelle. À mesure qu'il progresse, le joueur découvre la véritable cause de l'obscurité : un voile de nanites déployé par la corporation Solaris pour maintenir sa domination mondiale.

Solaris est une méga entreprise qui vend de la lumière à la population à travers des cristaux

Le gameplay s'articule autour de trois piliers :

1. Exploration et observation, avec un level design ouvert et interconnecté.
2. Combats dynamiques, contre des créatures appelées *Nocturnes*, sensibles à la lumière et au son.
3. Progression et découverte, à travers des capacités débloquées via l'exploration (double saut, dash, etc.) et un équipement évolutif.

Le monde se divise en zones distinctes mais interconnectées, chacune possédant sa propre identité visuelle, ses ambiances sonores et ses secrets. L'environnement réagit à la présence du joueur : la lumière influence les ennemis, et le son révèle ou attire le danger.

4.2 Cas d'usage principaux

CU1 : Exploration du monde

- Le joueur effectue l'exploration de différentes zones du monde du jeu.
- Le système affiche de nouveaux environnements et limite l'accès à certaines zones tant que les capacités ou objets requis ne sont pas obtenus.

- Résultat attendu : le joueur découvre progressivement le monde et identifie des zones accessibles ou verrouillées.

CU2 : Déblocage de capacités et de passages

- Le joueur effectue l'acquisition de capacités ou d'objets clés au cours de son aventure.
- Le système débloque de nouveaux passages ou mécaniques de déplacement.
- Résultat attendu : le joueur peut accéder à de nouvelles zones et progresser dans le jeu.

CU3 : Combat contre les ennemis

- Le joueur effectue des combats contre des ennemis rencontrés lors de l'exploration.
- Le système gère le comportement des ennemis (déplacement, attaque, repli) à l'aide d'une intelligence artificielle simple.
- Résultat attendu : le joueur doit adapter sa stratégie pour vaincre les ennemis et continuer sa progression.

4.3 Contraintes globales

Le jeu sera conçu pour fonctionner exclusivement sur PC, sous environnement Windows, avec un objectif de fluidité à 60 FPS constants.

Le style graphique adopté sera du pixel art de précision, inspiré de *Dead Cells* et *Terraria*, avec des contrastes forts entre lumière et obscurité.

L'ambiance sonore reposera sur un mixage de créations originales et de sons environnementaux réels, amplifiant la tension et la cohérence du monde.

Le développement doit respecter un calendrier académique structuré sur l'année scolaire 2025–2026, avec trois soutenances principales.

5. Exigences fonctionnelles

5.1 Liste des fonctions attendues (classées par thème)

Déplacements

Le joueur doit pouvoir se déplacer de façon fluide et intuitive dans un environnement 2D. Les actions de déplacement incluent la marche, la course et le saut.

Ces éléments constituent la base du gameplay, indispensables pour l'exploration et la survie dans les zones obscures.

(Obligatoire)

Combat

Les combats se dérouleront en temps réel.

Le joueur dispose d'une attaque de base, évolutive via les objets et capacités obtenues lors de la progression.

Les ennemis, représentés principalement par les "Nocturnes", disposeront d'une IA simple mais cohérente : déplacement aléatoire ("Errer"), poursuite du joueur lors de la détection ("Attaquer"), et potentiellement fuite après perte de points de vie ("S'échapper").

(Obligatoire)

Capacités

Les compétences spéciales telles que le double saut, le dash ou les attaques spéciales seront acquises par exploration.

Chaque nouvelle capacité débloquera des zones auparavant inaccessibles, favorisant l'exploration non linéaire.

Ces capacités s'intégreront naturellement dans le lore : elles représentent, symboliquement, la résilience et les souvenirs de Kaelen et de son fils Jonas.

(Souhaitable)

Objets

Les objets sont des éléments interactifs essentiels à la progression : équipements ou artéfacts.

Ils permettent d'améliorer les statistiques du joueur, d'ouvrir des zones spécifiques ou d'enrichir la narration.

Certains objets deviennent également des "déclencheurs environnementaux", nécessaires à la résolution de puzzles comme l'activation de leviers.

(*Obligatoire*)

Level Design

Le monde d'*Echo* est constitué de zones interconnectées, dans la tradition du Metroidvania.

Le joueur devra explorer, débloquer des raccourcis, et découvrir des secrets dissimulés dans des zones sombres nécessitant parfois des sources de lumière artificielle pour progresser.

(*Obligatoire*)

Coop en ligne

Un mode de coopération en ligne P2P permettra à plusieurs joueurs d'évoluer simultanément dans le même univers.

Les rôles seront identiques, et la progression sera synchronisée : lorsqu'un joueur quitte la session, ses données (objets, capacités, progression) seront conservées.

Un joueur fera office d'hôte du serveur, les autres se connectant en pair-à-pair.

(*Souhaitable*)

Interface

Le jeu comportera un menu principal permettant de lancer une partie, charger une sauvegarde et accéder aux paramètres (volume, luminosité, configuration des touches).

Une interface en jeu affichera les points de vie, les objets collectés, et les capacités actives.

(*Obligatoire*)

Boss

Au moins un combat contre un boss majeur sera intégré.

Ce combat servira de point culminant narratif, mêlant challenge et émotion.

Le premier boss prévu est le Capitaine Vex, l'assassin de Jonas.

5.2 Description des fonctions (modèle de formalisation)

La description des fonctionnalités du jeu s'appuie autour de trois éléments clés : les conditions d'entrée, le déroulement et les conditions de sortie.

Par exemple, pour l'action de déplacement en course, l'enchaînement débute lorsque le joueur active la touche dédiée, à condition que l'environnement le permette ; le personnage accélère progressivement, accompagné d'une animation fluide et d'un renforcement sonore des pas, jusqu'à ce que la course prenne fin, soit par relâchement de la touche, soit par un obstacle, entraînant un retour à la vitesse normale.

De la même manière, une action de combat basique s'initie lorsque le joueur déclenche une attaque en présence d'un ennemi proche ; l'attaque est exécutée, infligeant des dégâts calculés selon les statistiques des deux entités, tandis que des animations et effets sonores dédiés se déclenchent, avant que la séquence ne s'achève par la réaction de l'ennemi (perte de points de vie ou élimination) et l'application d'un léger délai avant la prochaine action.

La collecte d'un objet, quant à elle, commence dès que le joueur entre en contact avec un élément interactif visible ; l'objet disparaît, un effet sonore confirme la prise, et il est automatiquement intégré à l'inventaire, lequel est mis à jour en conséquence.

Ce modèle de présentation sera appliqué à l'ensemble des mécaniques principales afin de garantir une cohérence globale et une traçabilité optimale tout au long du développement.

6. Exigences non fonctionnelles

6.1 Performances attendues

Le jeu devra garantir une expérience fluide et sans latence, permettant une immersion totale du joueur. Pour atteindre cet objectif, plusieurs exigences techniques sont définies : des temps de chargement courts, idéalement inférieurs à trois secondes, un frame rate stable de 60 FPS sur une configuration PC moyenne ainsi qu'une réactivité immédiate des contrôles, notamment pour les actions basiques.

6.2 Fiabilité, sécurité, maintenabilité

Le développement reposera sur une architecture de code claire et structurée afin de faciliter l'intégration de futures fonctionnalités. Nous avons mis en place un système de sauvegarde permettant de conserver la progression du joueur, ainsi qu'une stabilité globale du jeu, en évitant les crashs. L'utilisation de Visual Studio Code pour le développement et d'un répertoire Git pour le suivi des versions permettra d'assurer un code propre, organisé et sécurisé tout au long du projet.

6.3 Ergonomie et accessibilité

L'interface du jeu se veut simple, fonctionnelle et lisible.

Tous les éléments à l'écran (barres de vie, items, menus, messages, feedbacks visuels) seront lisibles et contrastés, adaptés au style pixel art.

Des indications visuelles (icônes, effets de lumière ou vibrations visuelles) signaleront les points d'interaction, les dégâts subis ou les capacités disponibles.

Les paramètres d'accessibilité incluront le réglage du volume sonore (musiques et effets), la configuration des touches.

6.4 Contraintes réglementaires éventuelles

Le projet étant à vocation pédagogique, il n'y a pas de vraies contraintes légales. Cependant, tout le contenu du jeu est libre de droits (sons, textures, ...) et les données utilisateurs sont respectées (aucune collecte ou envoi de données personnelles dans le mode réseau).

7. Hypothèses et dépendances

7.1 Hypothèses de départ

Le développement du projet *Echo* repose sur plusieurs hypothèses établies dès son lancement :

Le jeu est destiné exclusivement au PC (Windows), sans portage console ou mobile prévu pour l'instant mais potentiellement une adaptation Linux. Le calendrier académique 2025–2026 offre une progression logique et suffisamment étalée pour permettre la conception, l'intégration et les phases de test des fonctionnalités essentielles.

Enfin, les membres de l'équipe collaborent au sein d'une base de code commune, hébergée sur un dépôt Git partage pour une meilleure coordination. L'ensemble de ces hypothèses assure le bon avancement du projet dans un cadre scolaire et la cohérence entre les livrables intermédiaires et le produit final.

7.2 Dépendances

Le projet repose sur plusieurs dépendances techniques et fonctionnelles :

- Moteur de jeu et librairies externes :

Utilisation de Pygame, bibliothèque Python destinée au développement de jeux 2D.

Certaines fonctionnalités additionnelles pourront s'appuyer sur des bibliothèques complémentaires (gestion sonore, effets visuels).

- Fonctionnalités réseau :

L'implémentation du mode coopératif P2P dépendra de la stabilité des modules Python utilisés.

- Outils de création graphique et sonore :

Les logiciels de pixel art (par ex. Aseprite, pixilart.com) et de composition sonore (Audacity, LMMS ou outils équivalents) devront utiliser et partagés entre l'équipe.

- Infrastructure web :

Le site web du projet est hébergé sur GitHub Pages, et utilise une base de données stockée par un service externe.

Tout dysfonctionnement de ces outils représente un risque pour la qualité ou la cohérence du produit final, et devra être anticipé.

8. Critères d'acceptation et de validation

8.1 Méthodes de vérification

L'évaluation de la conformité du jeu aux spécifications se fera selon les étapes de validation suivantes :

- Démonstrations intermédiaires :

Présentations orales et testables du prototype d'Echo à chaque jalon de soutenance.

Ces démonstrations permettront de vérifier la stabilité du gameplay, la bonne intégration des mécaniques de base et la cohérence de la direction artistique.

- Tests utilisateurs internes :

Sessions de test menées au sein de l'équipe pour observer le ressenti des joueurs, détecter les anomalies et ajuster la difficulté, la fluidité et la visibilité des éléments interactifs.

- Prototype validant :

Des versions jouables à étapes clés (déplacements, combat, IA, zones interconnectées) permettront de valider progressivement les blocs fonctionnels avant leur intégration définitive.

8.2 Indicateurs de réussite

Le projet sera considéré comme réussi si les critères suivants sont atteints :

- Le jeu est entièrement jouable du début à la fin, avec un scénario et un level design cohérents.
- Le gameplay est fluide et équilibré, sans latence ni ralentissements perceptibles.
- Le level design respecte la logique d'exploration évolutive propre au style Metroidvania.
- La direction artistique et sonore s'accorde parfaitement avec le ton narratif sombre et immersif de l'univers.
- Aucun bug bloquant n'empêche la progression ou la sauvegarde.
- Les participants à la soutenance peuvent jouer et comprendre l'univers sans explication extérieure, grâce à la narration environnementale et au soin apporté aux détails visuels et sonores.

9. Planning et estimation

9.1 Grandes étapes

Le développement du projet *Echo* suivra une progression structurée en plusieurs phases :

1. Prototype de gameplay :

Mise en place des déplacements, mécaniques de base (marche, saut, attaque simple), première intégration graphique.

2. Intégration des zones et ennemis :

Conception des premières cartes et ajout des IA basiques (errance, poursuite, attaque).

3. Ajout des capacités :

Développement des mécaniques de dash, double saut, et capacités spéciales débloquées via exploration.

4. Implémentation du boss et narration :

Création du combat contre le boss principal (Vex), intégration du lore et des interactions visuelles/narratives.

5. Optimisation et finitions :

Corrections de bugs, équilibrage des zones, amélioration du rendu sonore et graphique, et polish final avant livraison.

9.2 Jalons importants

Les étapes clés du projet sont alignées sur les soutenances académiques :

- Soutenance 1 : du 12 au 16 janvier 2026 – Présentation du prototype de base (déplacement, combat, interface).
- Soutenance 2 : du 16 au 20 mars 2026 – Présentation d'une version intégrée avec progression, ennemis, et éléments de lore intégrés visuellement.
- Soutenance finale : du 25 au 29 mai 2026 – Démonstration du jeu complet, jouable et optimisé, accompagné du site vitrine.

9.3 Estimation des délais

Le développement se déroulera sur environ huit mois, avec une répartition du travail accompagnant le rythme des soutenances :

- Septembre – Décembre 2025 : conception, prototype et premiers tests fonctionnels.
- Janvier – Mars 2026 : intégration globale des zones, premières mécaniques et ennemis.
- Avril – Mai 2026 : finalisation, équilibrage, et tests de validation.

Chaque membre de l'équipe disposera d'un tableau de bord Notion répertoriant les tâches essentielles, leur état d'avancement et leur validation par les pairs.

9.4 Estimation prévisionnelle des coûts

Aucune dépense financière directe n'est prévue.

Le projet repose sur le temps de travail des membres de l'équipe et l'utilisation d'outils gratuits ou mis à disposition par l'école.

- Temps-homme : uniquement interne à l'équipe, réparti selon les compétences (développement, graphisme, son).
- Ressources logicielles : outils gratuits (Pygame, Aseprite, Audacity, Notion, Git).
- Matériel : postes personnels des membres de l'équipe.

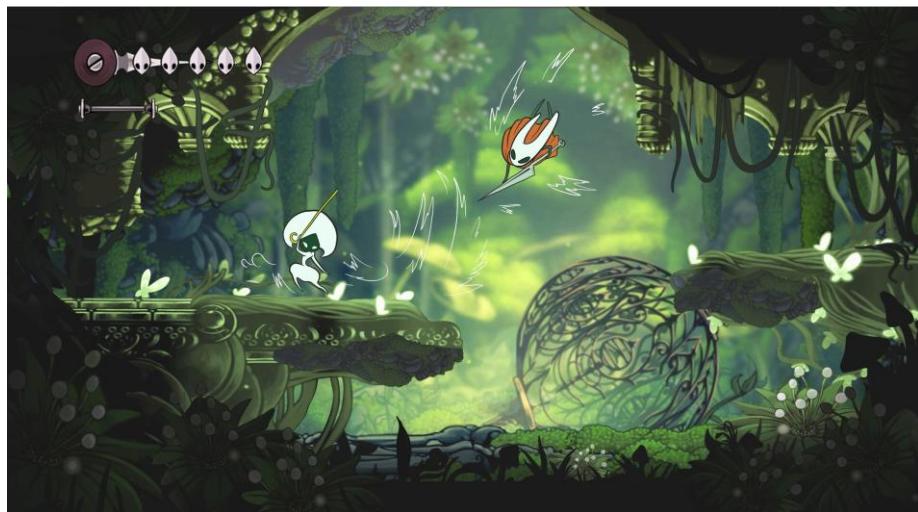
Aucune dépense d'hébergement ou de licence payante n'est requise pour la version finale.

Annexes

Références et inspirations

Le jeu s'inspire du travail artistique et ludique de plusieurs références du genre :

- *Hollow Knight* pour l'aspect narratif environnemental et l'ambiance visuelle.



- *Dead Cells* pour la précision du pixel art et la fluidité des combats.



- Et d'autres comme Terraria



CDSF

Version v1.0

10/01/2026

AUTEUR PRINCIPAL

Team Nightberry

CONTRIBUTEURS

- Amaury Giraud-Laforet
- Eric Sahakian
- Florian Croiset
- Gaspard Sapin
- Jules Cohen

DESCRIPTION

Cahier des Spécifications Fonctionnelles

*Cahier des Spécifications Fonctionnelles du groupe LYN_L2.1_G2
pour la soutenance du 13/01.*