

Aide-mémoire – GDevelop

Création d'un projet

- Toujours cocher *Optimiser pour le pixel art*
- Propriétés du projet : définir l'auteur et l'éditeur
- Nommer sa scène **NiveauTest** ou **Niveau1**
- Afficher la grille et le masque
 - Configurer la grille selon les unités du jeu (16×16 par défaut)
- Coder le débogage dans une feuille d'événements externe séparée nommée **Debug**

EcriturePascale

- Chaque nom commence par une majuscule, et chaque nouveau mot aussi (ex. : JoueurPrincipal)
- Aucun accent, aucune espace, aucun trait d'union

Exceptions notables :

- Nom de l'éditeur dans les propriétés du projet
- Texte affiché dans le jeu

Concepts clés

- **Propriété** : toutes les propriétés peuvent être modifiées par les événements.
- **Objet vs instance** : l'objet est le modèle (liste d'objets); l'instance est une copie placée dans la scène. Modifier l'objet change toutes ses instances; modifier une instance ne change qu'elle.
- **Comportement** : fonctionnalité prête à l'emploi ajoutée à un objet (ex. : Personnage de plateforme, Plateforme). Évite de coder soi-même le mouvement ou les collisions.
- **Variable** : valeur nommée qu'on peut lire et modifier. Trois portées :
 - *Globale* : partout dans le jeu
 - *De scène* : dans la scène seulement
 - *D'objet* : propre à chaque instance
- **Calque** : couche d'affichage qui a sa propre caméra.
 - **Utilité** : séparer ce qui suit la caméra (niveau, personnages) de ce qui reste fixe à l'écran (interface, pointage, vies), et contrôler l'ordre d'affichage (arrière-plan, parallaxe, avant-plan).
 - **Limiter leur nombre** : chaque calque a sa propre caméra à synchroniser, et chaque calque de plus augmente le risque de placer un objet sur le mauvais calque. Viser le minimum utile (ex. : ArrierePlan, Niveau, Interface).
- **TimeDelta()** : temps écoulé depuis la dernière image, en secondes. On multiplie un déplacement codé à la main par TimeDelta() pour que la vitesse soit la même peu importe le nombre d'images par seconde.

Évènement

- Les événements sont évalués à chaque image rendue.
- On n'est pas obligé de mettre une condition : une action sans condition s'exécute à chaque image.
- On peut combiner plusieurs conditions (ET, OU) et mettre plusieurs actions.
- Dans la boîte d'expression, on peut utiliser des variables.
- **Sélection d'objets** : les conditions filtrent les instances; les actions s'appliquent seulement aux instances sélectionnées. Sans condition sur l'objet, l'action s'applique à *toutes* ses instances.
- **Sous-événement** : s'exécute seulement si les conditions de l'évènement parent sont vraies.
- **Déclencher une fois** : empêche une action de se répéter à chaque image tant que les autres conditions restent vraies.

Points de l'image

- **Centre** : rotation
- **Origine** : position (translation) et échelle

Exercices – Symboles

- **Icône robot** : on doit coder
- **Icône étoile** : nouvelle notion
- **Icône crayon** : _____
- **Mots en gras** : mots recommandés à chercher pour trouver une condition ou une action