

Core Feature du jeu

Le système de Tuile

Sommaire :

- [Introduction](#)
- [Vocabulaire](#)
- [Tuile : Caractéristiques, interactions et comportement avec le monde](#)
 - [Caractéristiques d'une tuile](#)
 - [Différents types de tuile](#)
 - [Comportement avec le monde](#)
- [La grille : Caractéristiques et fonctionnement](#)
 - [Feedbacks de la grille](#)
- [Placement de tuile : Inputs et interactions du joueur](#)
 - [Liste des inputs](#)
 - [Contrainte de la physique](#)
 - [Modifications de tuiles](#)
 - [Autres informations importantes](#)
- [Tuile : Ressources pour implémentation](#)
- [Avatar : Interactions et réactions de l'avatar face aux tuiles](#)
 - [Réactions face à des tuiles par défaut](#)
 - [Réactions face à des tuiles modifiées](#)
- [Entités : interactions et réactions des entités face aux tuiles](#)
 - [Tuiles comme éléments bloquant](#)
 - [Tuiles comme moyen de déplacer les ennemis](#)
- [Annexe](#)

Disclaimer : Ce document est un support de travail à destination des programmeurs ayant pour but d'expliquer en profondeur **le système des tuiles et son fonctionnement**. Pour toutes informations sur d'autres features du jeu veuillez vous référer aux documents qui leurs sont attribués ou encore au [document de concept](#) pour un point de vue global.

Introduction :

[\(Revenir au sommaire\)](#)

Tout le principe du jeu repose sur le **placement de tuile et leur agencement dans un espace 3D** afin de permettre à l'avatar du joueur de naviguer sur celles-ci pour éliminer les ennemis et atteindre la sortie du niveau.

(voir : [Document de concept pour plus de détails](#))

L'espace 3D dans lequel le joueur est en capacité de poser des tuiles est **décomposé en une grille**.

(voir : [La grille : Caractéristiques et fonctionnement](#))

Les tuiles possèdent des **caractéristiques essentielles**.

Ces caractéristiques définissent ce que le joueur peut en faire et comment celles-ci interagissent avec leur monde.

(voir : [Tuiles : Caractéristiques, interactions et comportement avec le monde](#))

Le joueur place ces tuiles via différents inputs, lui permettant plusieurs actions. Le placement de tuile est soumis à des règles physiques de base.

(voir : [Placement de tuiles : inputs et interactions du joueur](#))

En fonction de l'agencement de tuile auquel le joueur souhaite confronter son avatar, celui-ci va **réagir de différentes manières**.

(voir : [Avatar : interactions et réactions de l'avatar face aux tuiles](#))

L'enjeu de cette feature réside donc dans le fait de **prendre en compte toutes les différentes situations, agencements, règles et variables entrant en jeu**.

Vocabulaire :

[\(Revenir au sommaire\)](#)

- **Joueur** : Le joueur est la personne physique qui place les tuiles et lance l'avatar.
 - **Avatar** : L'avatar est le coureur que peut lancer le joueur.
 - **Tuile** : Les tuiles sont les outils du joueur pour faire progresser son avatar dans le niveau. Ce sont avec elles que le joueur interagit.
 - **Case** : Le terme case est utilisé pour définir un endroit sur lequel le joueur peut placer, ou non, une tuile. Une case est toujours représentée par une des faces d'une tuile.
 - **Réserve** : Endroit où les tuiles du joueur sont conservées. La réserve est visible dans le coin en bas à droite de l'écran du joueur.
-

Tuiles : Caractéristiques, interactions et comportement avec le monde :

[\(Revenir au sommaire\)](#)

Caractéristiques :

- **Taille** : 1x1x1
- **Formes** : Cube / Pente
- **Nombre de face** : 6 pour les cubes / 5 pour les pentes
- **Unité de poids** : 1

Le nombre de face d'une tuile définit le nombre de case sur lequel le joueur peut placer d'autres tuiles.

*L'unité de poids d'une tuile doit pouvoir être modifiable par les Game designer depuis l'inspecteur par le biais d'une variable public (**weightUnit**).*

Exemple :

- Je place une tuile sur un emplacement disponible de la grille. La face sur laquelle j'ai posé ma tuile n'est plus disponible car elle est déjà utilisée.
- Je souhaite replacer une nouvelle tuile sur l'une des faces de la tuile posée précédemment.
- J'ai donc le choix entre les 5 faces disponibles de la tuile qui représentent en réalité 5 cases de la grille sur lesquelles je peux placer une nouvelle tuile.

Types de tuiles : _____

Il existe **3 types de tuiles**, partageant **les mêmes caractéristiques** (cela peut aussi prendre la forme d'une tuile qui dispose de 3 états, à voir ce qui est le plus pratique en programmation) :

- **Non-navigable** : Les tuiles présentes à l'origine sur un niveau ne sont pas navigables. Le joueur peut poser des tuiles navigables sur celles-ci.
- **Navigable** : **Les tuiles que pose le joueur** sont les seules sur lesquelles l'avatar peut naviguer.
- **Morte** : Les tuiles mortes sont des tuiles sur lesquelles l'avatar ne peut pas naviguer et où le joueur ne peut pas poser d'autres tuiles (comparable aux murs noirs de Portal).

Le joueur ne dispose que des tuiles **navigables** mais les Game designer et Level designer doivent pouvoir placer n'importe quelle type de tuile en moteur pour créer un niveau.

*Les Game designer doivent avoir la possibilité, en cliquant sur une tuile dans la scène de jeu, de **choisir par le biais d'une liste déroulante quelle est le type d'une tuile** (Surtout utile pour les Level designer lors de la production du layout de base des niveaux).*

En fonction du choix effectué dans le menu déroulant, le visuel et le fonctionnement de la tuile présente sont modifiés.

Comportement avec le monde : _____

Les tuiles ont un comportement unique lors des interactions avec le monde qui est toujours le même peu importe l'agencement ou la forme de la tuile :

- **A chaque fois que l'avatar ou une entité marche sur une tuile (rentre en contact avec celle-ci) elle se détruit au bout d'un certain temps.**

*Il est important de permettre aux Game designer de pouvoir modifier le temps avant que la tuile se détruise depuis **une variable public ([timeBeforeDestruction](#))** présente dans l'inspecteur. (Mettre une valeur de base de 1 seconde.)*

Il est important de noter que le fonctionnement de la physique doit être "systémique". Dans l'éventualité où le placement initial de tuile permettait à toutes les tuiles de tenir mais que ce n'est plus la cas une fois que l'avatar marche sur celle-ci et qu'elle se détruisent.

Alors, les tuiles, qui ne sont plus supposées tenir (voir : [implication de la physique sur les tuiles](#)) doivent se détruire en temps réel. Pouvant parfois entraîner une réaction en chaîne.

La grille : Caractéristiques et fonctionnement

[\(Revenir au sommaire\)](#)

La grille est présente et visible sur l'entièreté d'un niveau, elle est **l'endroit sur lequel le joueur peut placer ses tuiles.**

Celle-ci est elle-même **composée de tuiles 3D** qui peuvent être de différents types (cliquez ici pour voir [les types de tuiles](#)).

En fonction du type de tuile qui compose la grille, le joueur peut ou non, placer d'autres tuiles sur les cases de celle-ci.

Les cases de la grille sont les faces d'une tuile.

Autrement dit, **chacune des 6 faces d'une tuile représente une case de la grille** sur laquelle le joueur va pouvoir ou non, en fonction du type de tuile, venir placer de nouvelles tuiles.

Cela signifie que lorsque le joueur place une nouvelle tuile, alors il **ajoute au maximum 5 nouvelles cases** à la grille (peu importe le type de case) sur laquelle il peut encore par la suite placer des nouvelles tuiles, et ainsi de suite.

C'est cette grille qui permet au joueur de savoir s'il peut ou non poser des tuiles en lui indiquant par le biais de retours visuels.

Feedbacks de la grille : _____

- Si le joueur place son curseur sur **une case de la grille pouvant recevoir une tuile** alors **cette case se colore en vert**.
- Si le joueur place son curseur sur **une case de la grille ne pouvant pas recevoir de tuiles** (cases uniquement présentes sur les [tuiles mortes](#)) alors **cette case se colore en rouge**.
- Si le joueur essaie quand même de **placer une tuile sur une case rouge** alors, la caméra effectue une légère vibration et la case clignote en rouge.
- Lorsque le joueur **survole la grille avec le curseur de sa souris** un sfx se lance pour chaque case de la grille survolée.

Tous ces paramètres de sons, ainsi que les couleurs que prennent les cases de la grille **doivent pouvoir être modifiés par les Game designer** depuis des paramètres accessibles dans l'inspecteur.

Placement de Tuile : Inputs et interactions du joueur

[\(Revenir au sommaire\)](#)

Le joueur à chaque lancement d'une partie dispose **d'une quantité limitée (représenté par un nombre entier) de tuiles** défini en fonction du niveau sur lequel il se trouve.

Le joueur peut donc placer ses tuiles dans le niveau dans la limite de la quantité dont il dispose.

Cette quantité de tuiles est constamment visible le coin en bas à droite de l'écran sous la forme d'une UI affichant le nombre de tuiles restantes.

(reprendre l'affichage utilisé dans [Dorf Romantik](#)).

De plus, ce placement de tuile ne peut s'effectuer uniquement en dehors de la phase de course. **Il est impossible de placer des tuiles lors de la phase de course de l'avatar.**

*Il est nécessaire que les Game Designer puissent avoir accès dans l'inspecteur à la variable leur permettant de choisir le nombre de tuiles (**tilesQuantity**) dont dispose le joueur sur le niveau actuel.*

Si le joueur n'a plus de tuile alors il ne peut plus en poser.

En d'autres termes si **tilesQuantity** ≤ 0 alors, le joueur ne peut plus poser de tuile.

Le joueur dispose de **plusieurs inputs et interactions possibles** avec les tuiles lors du placement.

En voici la liste des commandes / inputs basiques :

- **Pour placer une tuile**, le joueur doit placer son curseur sur une des cases disponible de la grille, puis effectuer **un clic gauche** dessus (voir : [La grille : Caractéristiques et fonctionnement](#)).
- **Pour détruire une tuile**, le joueur doit placer son curseur sur une des cases disponible de la grille, puis effectuer **un clic droit**.

- **Pour placer ou détruire simultanément plusieurs tuiles**, le joueur peut maintenir la touche Ctrl de son clavier, puis faire glisser sa souris en maintenant le clic correspondant à l'action souhaitée depuis une case disponible de la grille.
Si une case de la grille comprise dans la sélection multiple n'est pas disponible alors, elle n'est pas prise en compte et ne subit pas de modification. Ce système reprend le fonctionnement de [CubeGrid dans Unreal Engine](#).

Les 2 premiers inputs reprennent exactement le fonctionnement de placement et destruction de tuiles proposé par **TownScaper** donc, en cas de doutes ou de précisions manquantes [cliquez ici](#) afin de visualiser le fonctionnement.

Placer une ou plusieurs tuiles à pour effet de consommer des ressources dans la réserve du joueur. On doit alors soustraire le nombre de tuiles placées par le joueur à sa quantité limite de tuiles dont il dispose (**tilesQuantity** - nombre de tuiles placées).
Au contraire, si le joueur **détruit une ou plusieurs tuiles, alors celles-ci retournent dans sa réserve** et celle-ci se voit incrémentée du nombre de tuiles détruites (**tilesQuantity** + nombre de tuiles détruites).

Cette soustraction et incrémentation doit se faire dynamiquement et doit instantanément être visible pour le joueur.

Des feedbacks visuels et sonores doivent être présents pour informer le joueur des ses actions :

- **Lorsque le joueur dépense ou reçoit des tuiles**, le nombre présent sur l'UI doit osciller et avoir un effet élastique (un scale à 1.25x puis retour à 1x) à chaque tuile posée.
- **Lorsque le joueur n'a plus de tuiles / vient de placer sa dernière tuile**, le nombre afficher sur l'UI du joueur doit scale-up puis scale-down tout en clignotant en rouge.

 **Pour plus de détails techniques** sur le fonctionnement exact du placement des tuiles, vous pouvez vous référer au [behaviour Tree](#) qui lui est associé.

Disclaimer : Le Behaviour Tree et ce document vont de paire, ils se complètent l'un l'autre, alors il est important de prendre connaissance des 2 documents avant de programmer la feature.

Contraintes physiques : _____

De part leur **caractéristique de poids**, lors du placement de tuile, **le joueur est contraint par quelques règles physiques** :

- **Les tuiles ne peuvent pas voler**. Il est donc impossible de placer une tuile seule dans les airs.

Exemple :

- Si une tuile se retrouve dans les airs (en train de voler) lors de la course, alors elle est détruite.

- **Une tuile peut supporter jusqu'à une certaine valeur maximum d'unités de poids supplémentaire avant de s'effondrer.** Si la valeur maximum d'unité de poids qu'une tuile peut supporter est dépassée alors, cette tuile est détruite et ajoutée à la réserve.

Exemple (avec une valeur de 3 unités de poids maximum) :

- Il est donc possible d'empiler 4 tuiles maximum. La 5ème tuile fait se briser la première.
- Il est impossible de créer un pont dans le vide de plus de 4 tuiles de longueur depuis une tuile qui repose sur le sol. La tuile qui repose sur le sol va supporter la création de 3 tuiles supplémentaires mais pas plus. La 4ème ne peut pas être construite.

La variable (**maxWeightUnits**) qui définit la valeur maximum d'unité de poids supplémentaire (sous la forme d'un nombre entier) avant qu'une tuile ne s'effondre doit être modifiable par les Game Designer depuis l'inspecteur.

🚩 Pour des explications plus techniques, veuillez consulter [le Behavior tree : "Fonctionnement de la physique"](#).

Pour des explications plus visuelles, veuillez consulter [les schémas de fonctionnement de la physique ici](#).

Des feedbacks visuels doivent permettre au joueur de **connaître en continu le poids que supporte une tuile** :

- Pour se faire, la couleur d'une tuile doit disposer de x dégradés de couleurs correspondant à la valeur contenue dans la variable **maxWeightUnits**. Au plus la tuile supporte de poids, au plus le dégradé de couleur est foncé.
- **Lorsque le joueur place son curseur sur une tuile qu'il a posée**, un UI en world apparaît en fondu au bout de 0.5 secondes, lui indiquant **l'unité de poids actuel que supporte la tuile / maxWeightUnits**.

Modification de tuiles : _____

Une fois une tuile placée, **le joueur peut la modifier** pour que celle-ci passe **d'une forme de cube à une forme de pente**.

Pour se faire le joueur dispose de quelques interactions simples.

Voici la liste des commandes qui permettent de modifier une tuile :

- **Pour transformer une tuile en forme de cube vers une tuile en forme de pente**, le joueur doit effectuer **un clic molette** sur celle-ci.
- **Pour transformer simultanément plusieurs tuiles**, le fonctionnement est le même que lors du placement ou de la destruction simultanée de plusieurs tuiles ([à retrouver ici](#)) mais en pressant cette fois le clic molette.

Lors de la transformation d'une tuile celle-ci passe par plusieurs étapes :

- 1) Le joueur clique une fois sur une tuile en forme de cube alors **celle-ci prend la forme d'une pente** qui est dirigée vers le nord de l'écran.
- 2) A chaque clique molette du joueur sur cette tuile, **elle effectue une rotation de 90°** dans le sens des aiguilles d'une montre.
- 3) Au bout de 4 clics après la première transformation de la tuile. Donc, lorsque celle-ci a effectué une rotation complète. **La tuile se transforme à nouveau en une tuile en forme de cube.**

🚩 Afin de profiter d'une **explication plus détaillée et technique** sur le fonctionnement de la modification de tuile, rendez-vous sur le [behaviour tree : "Modification de tuile"](#)).

Informations importantes : _____

Afin d'éviter au joueur de toujours devoir effectuer cette manipulation lors de la transformation, **les tuiles placées par le joueur qui sont adjacentes à une pente et dans le sens de la pente, prennent automatiquement les propriétés de la pente adjacente.**

Par exemple :

- Je viens de placer une tuile, je l'ai transformée en une pente et je l'ai mise dans l'orientation souhaitée.
- Alors, la tuile que je vais par la suite poser sur une case adjacente et dans le sens de la descente ou de la montée de celle-ci aura automatiquement les mêmes propriétés.

Cela permet d'effectuer facilement et rapidement des pentes en limitant le nombre d'inputs pour le joueur.

Dans le cas où le joueur place une tuile entre 2 autres tuiles dont une en forme de cube et une en forme de pente, celle-ci prend **par défaut la forme de cube.**

🚩 (Pour de plus amples informations et explications techniques à propos de cette modification automatique de tuile, **rendez-vous sur le [behaviour tree du placement de tuile](#)**)

Module Particules : _____

Les Game designer doivent pouvoir **ajouter et modifier les particules émises par les tuiles lors de leur placement** directement depuis l'inspecteur en un simple drag & drop du système de particules.

Voici une liste (*non exhaustive*) des paramètres à prendre en compte :

- **Actions / Triggers :**
 - Placement d'une tuile

- Destruction d'une tuile (ce vfx est également lancé lorsque l'avatar détruit une tuile en marchant dessus, voir : [comportement des tuiles pour plus de détails](#))
- Modification d'une tuile

Module sons : _____

Les Game designer doivent pouvoir **attribuer des sons en fonction de différents paramètres / événements** en un simple drag & drop du SFX dans l'inspecteur.

Voici une liste (non exhaustive) des paramètres en questions :

- **Action(s) / Trigger(s)**
 - Placement d'une tuile
 - Destruction d'une tuile (ce sfx est également lancé lorsque l'avatar détruit une tuile en marchant dessus, voir : [comportement des tuiles pour plus de détails](#))
 - Modification d'une tuile
- Looping on / off

Placement de Tuile : Ressources pour implémentation

[\(Revenir au sommaire\)](#)

- [TownScaper](#) : Pour le placement de tuile en 3D sur les 6 faces d'une tuile.
- [Dorf Romantik](#) : Important pour l'affichage des tuiles disponibles ainsi que de la quantité de tuiles restante.

Au travers de ces liens vous trouverez plusieurs exemples sur le fonctionnement essentiel des tuiles et sur la manière de les placer.

Avatar : Interactions et réactions de l'avatar face aux tuiles

[\(Revenir au sommaire\)](#)

L'avatar du joueur est l'entité qui va réaliser la phase de course après que le joueur ait terminé de placer et agencer ses tuiles.

(Pour plus de détails sur l'avatar rendez-vous dans [le document de concept ici](#))

L'agencement des tuiles régit le comportement de l'avatar lors de sa phase de course.

Plusieurs réactions de l'avatar sont possibles en fonction des différents patterns d'agencement qu'il rencontre.

Ci-dessous, **la liste exhaustive de toutes les réactions de l'avatar en fonction des différents agencement.**

Les schémas des réactions sont à retrouver [ici](#) et les explications ci-dessous :

Réactions de l'avatar sur des tuiles en forme de cube :

- Si les tuiles sont toutes à la même hauteur, l'avatar court simplement dessus en prenant progressivement de la vitesse.
- Si l'avatar court sur des tuiles et rencontre un empilement d'une tuile alors il va sauter au dessus en effectuant un bunny hop.
- Si l'avatar court sur des tuiles et rencontre un empilement de 2 tuiles alors il va escalader par dessus en effectuant un passe murail.
- Si l'avatar court sur des tuiles et rencontre un empilement de 3 tuiles alors il va s'écraser dans le mur qu'elles représentent.
- Si l'avatar court sur des tuiles et rencontre un empilement de 2 tuiles comportant un trou à leur base alors il va glisser par dessous.
- Si l'avatar court sur des tuiles et rencontre un empilement de 2 tuiles situées sur un de ses côtés alors que devant lui les tuiles ne sont plus navigables, alors il va marcher sur le mur.

Réactions de l'avatar sur des tuiles en pente :

- Si l'avatar court sur une tuile qui est une pente positive alors il va sauter sur une distance de 2 tuiles dans la direction de la pente.
- Si l'avatar court sur des tuiles qui sont des pentes positives sur une certaine distance alors il avance sur celles-ci mais sa vitesse réduit. Une fois que sa vitesse atteint 0 alors il glisse dans le sens opposé.
- Si l'avatar court sur des tuiles qui sont des pentes négatives alors il glisse dessus jusqu'à être en contact avec un tuile qui n'est pas une pente.

🚩 Pour des explications plus techniques sur le fonctionnement de l'avatar vis-à-vis des tuiles, veuillez vous référer au [Behaviour Tree : "Réactions de l'avatar face aux tuiles"](#).

Pour des explications plus visuelles, veuillez vous référer aux [schémas de réactions juste ici](#).

Ennemis : Interactions et réactions des ennemis face aux tuiles

[\(Revenir au sommaire\)](#)

Les ennemis sont les éléments bloquants et dangereux pour le joueur lors de la phase de course. Les ennemis peuvent détecter l'avatar du joueur si celui-ci est dans leur champ de vision.

(Pour plus de détails concernant les ennemis et leur fonctionnement rendez-vous [ici](#))

Les tuiles agissent comme des murs qui bloquent la vision pour les ennemis durant la phase de course.

- Si le joueur place des tuiles de telle sorte à ce que la vision des ennemis soit bloquée, alors ils ne pourront pas détecter son avatar pendant la phase de course.

Durant la phase de placement de tuile, **le joueur peut également se servir des tuiles pour déplacer les ennemis.**

Les tuiles sur lesquelles se trouvent les ennemis sont des tuiles **non-navigables** alors le joueur est en capacité de placer des tuiles dessus.

- **Si le joueur place des tuiles en forme de cube sous un ennemi**, cela a pour effet de l'élever de 1 de hauteur.
 - **Si le joueur détruit une tuile sous un ennemi**, cela a pour effet de le faire descendre de 1 de hauteur.
 - **Si le joueur place des tuiles en forme de pente sous un ennemi**, cela a pour effet de faire glisser celui-ci jusqu'à la prochaine tuile en forme de cube sur laquelle il peut rester.
-

Annexe

Liens vers les documents de support pour la production de la feature !

Drive comportant les documents de support de la feature : [Cliquez ici](#)

Document de concept : [Cliquez ici](#)

Schémas de fonctionnement : [Cliquez ici](#)

Behavior documents : [Cliquez ici](#)

Listes des variables : [Cliquez ici](#)