Création d'un jeu vidéo

1. Première étape : personnage se déplaçant dans un labyrinthe

1.1. But du jeu

Le but de ce premier jeu est de créer un personnage se déplaçant dans un labyrinthe. Le jeu se termine lorsqu'il a atteint la sortie (matérialisée par un drapeau).

Cette étape permet d'apprendre les bases de la création d'un jeu vidéo :

- Le plateau
- Les objets
- Les sprites
- L'utilisation du clavier

1.2. Principe des éléments

Le jeu de compose de :

- 1 pièce
- 3 sprites (mur, personnage et drapeau)
- 3 objets (mur, personnage et drapeau)

Le personnage de déplace lorsqu'on appuie sur une des 4 touches de direction Il s'arrête lorsqu'on relâche les touches de déplacement et qu'il se situe complètement dans une des cases du quadrillage

1.3. Création du jeu

Création des sprites

Les sprites sont tous les éléments graphiques qui vont constituer le jeu. On retrouvera les personnages, les monstres, les objets à toucher, les éléments constitutifs du décor.

On peut les redessiner ou les charger à partir des fichiers d'exemple fournis.

Ils peuvent être fixes, ou bougeant. Dans ce cas là, ils se composent de plusieurs images qui s'affichent de façon cyclique

Nous allons créer 3 sprites de dimension 16*16 (par le menu Add -> Sprite ou en appuyant sur Ctrl Alt S)

Pour chaque sprite créé, nous allons charger une image, (à prendre en bibliothèque) au moyen de Load Sprite, éventuellement l'éditer pour le remettre à la bonne taille (au moyen de edit Sprite).



Création des objets

Les objets sont les éléments constitutifs du jeu, se déplaçant ou fixes. Chaque objet est associé à un sprite unique (mais un sprite peut être utilisé par plusieurs objets).

De plus à chaque objet sont associés des évènements, eux même reliés à des actions.

Lorsque un des évènements considérés se produit (appui sur une touche, collision par ex.), il en résulte une action représentée par une suite d'icônes.

Un objet est crée par le menu Add->Object ou par Ctrl Alt O.

Note : les objets peuvent avoir les mêmes noms que les sprites associés. Attention à ne pas les confondre.

Les briques

Les briques sont des éléments constitutifs du décor. Elles ne sont associées à aucune action

💽 Object Properties				×
Name: mur	Events:	Actions:	-Move	move main1
New Edit ✓ Visible ✓ Solid Depth: 0 ✓ Persistent				main2 control
Parent: Ino parent>			-Paths	score extra
<u>✓ Q</u> K	Add Event		- Steps	draw

Le personnage

Le personnage est l'élément principal du jeu. La plupart des actions lui sont associées.

Object Properties					x
Name: Personnage Sprite personnage Image: personnage New Edit Image: Visible Image: Solid Depth: 0 Image: Persistent Parent: <no parent=""> Mask: <same as="" sprite=""> Image: Show Information</same></no>	Events: mur (no key) (Left) (Up) (Right) (Down)	Actions:	- Move		move main1 main2 control score extra dr
<u>✓ 0</u> K	Add Event Delete Change		-Steps	:	aw

Nous avons ici 6 évènements

- Lorsque le personnage touche un mur -> il s'arrête
- Lorsqu'on n'appuie sur aucune touche -> si il est sur la grille, il s'arrête
- Lorsqu'on appuie sur la touche '<-' -> Si il est sur la grille, il part sur la gauche
 Lorsqu'on appuie sur la touche '->' -> Si il est sur la grille, il part sur la droite
- Lorsqu'on appuie sur la touche 'haut' -> Si il est sur la grille, il part vers le haut
- Lorsqu'on appuie sur la touche 'bas'

- -> Si il est sur la grille, il part vers le bas

Pour rajouter un évènement, il suffit de cliquer sur add event, puis de sélectionner le type d'évènement à rajouter.

Pour le premier évènement, nous choisirons Collision



Pour les 5 autres évènements nous choisirons keyboard, puis la touche associée (<left> <right>, <up>, <down>, ou <No key>).



Les actions à réaliser en fonction de l'évènement se choisissent à sur la partie droite de la fenêtre Object Properties.

Les déplacements se situent sur l'onglet 'move', les tests sur l'onglet 'control'



* Collision contre le mur :

L'action associée est « start moving in a direction » comme suit

			/ 2 / M IN
		- Start moving in a direction	
Events: → mur we key>	Actions: Start moving in a direction	Applies to Self Other Object: Directions: Directions: Other Object Obj	
Add Event		Speed: 4	
Delete Change		<u> </u>	🗙 Cancel

* Appui sur la touche gauche :

L'action associée est si on est sur la grille alors « start moving in a direction » comme suit

		Start moving in a direction
Events: mur cno key> CLeft> CUp> CRight> COwn>	Actions: If instance is aligned with grid Start moving in a direction	Applies to Self Other Object: Directions: Speed: 4
Add Event		
Delete Change		Cancel

Pour les 3 autres touches, l'algorithme est le même en changeant simplement la direction.

* Appui sur aucune touche :

L'action associée est « si on est sur la grille » alors « start moving in a direction » comme suit.

		Start moving in a direction
Events: mur mur mar <no key=""> mar <left> mar <lef< th=""><th>Actions: If instance is aligned with grid Start moving in a direction</th><th>Applies to Self Other Object: Directions: C 0 4 C 0</th></lef<></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></left></no>	Actions: If instance is aligned with grid Start moving in a direction	Applies to Self Other Object: Directions: C 0 4 C 0
		Speed: 0
Add Event Delete Change		

Le test de grille est obligatoire afin que le personnage ne s'arrête que lorsqu'il est sur une case complète. Cela simplifie le maniement du jeu.

Le drapeau

Le personnage est le point d'arrivée du labyrinthe. Il n'a qu'une seule action associée : si il touche le personnage alors on change de niveau de jeu.

Le comportement exact est le suivant :

- si un niveau supérieur existe (si une nouvelle pièce existe)
 - o alors on va dans la nouvelle pièce
- sinon

_

o la partie se termine

Object Properties			
Object Properties Name: Drapeau Sprite Image: Constraint of the second secon	Events: Personnage	Actions: Go to next room tos Else End the game	- Objects move main1 - Objects move main1 - Sprite - Sprite - Sounds - Sounds
<u>✓ 0</u> K	Add Event Delete Change		

Création de la pièce de jeu

La pièce de jeu est la pièce où se situe l'action du jeu. Elle se compose d'une image de fond et de plusieurs objets fixes en mouvement.

Une nouvelle pièce se crée au moyen du menu Add -> Room ou Ctrl+Alt+R

On choisira une taille de carreaux de 16 * 16 (Snap X et Snap Y).

On viendra ensuite créer le labyrinthe en rajoutant des briques, le personnage à son point de départ et le drapeau à l'arrivée.

Pour cela on clique d'abord dans la zone object et on choisit le type d'objet à rajouter, puis on rajoute les différents objets sur le plateau de jeu en cliquant sur le bouton gauche de la souris. On peut aussi supprimer un objet en cliquant sur le bouton droit de la souris.

Quand le labyrinthe est terminé, on réduira la taille de la pièce à la taille du labyrinthe dans le menu settings

backgro	ounds	<u>v</u> iews
objects	<u>s</u> ettings	<u>t</u> iles
<u>N</u> ame:	Piece 1	
Caption fo	or the room:	
Width:	288	
Height 2	288	
Speed:	30	
□ P <u>s</u>	rsistent	
E	Creation co	de

🖬 Room Properties		_ 🗆 🗙
✓ 19 🗋 ⇒ 😫 😫 👌	🔓 Snap⊻: 16 Snap⊻: 16 🔠 🔆 🔎 ▾	
<u>b</u> ackgrounds <u>views</u> <u>o</u> bjects <u>s</u> ettings <u>tiles</u>		
Object to add with left mouse:		
$ \begin{array}{c c} \hline mut \\ \hline \\ Laft mouse button = add \\ + < Alt> = no map \\ + < Shift> = multiple \\ + < Cri> = move \\ Right mouse button = delete \\ + < Shift> = delete all \\ + < Cri> = popup menu \\ \end{array} $		
Delete underlying	x: 192 y: 384	

1.4. Test du jeu



Le jeu se lance en cliquant sur On obtient le plateau de jeu suivant



La partie se termine lorsque le personnage a atteint le drapeau ou lorsqu'on appuie sur la touche 'escape'

2. Deuxième étape : personnage se déplaçant dans un labyrinthe, en mangeant des diamants

2.1. But du jeu

Le but de ce premier jeu est de créer un personnage qui se déplace dans un labyrinthe. Il mange des diamants. Le jeu s'arrête quand il a mangé tous les diamants.

Nous apprenons à utiliser dans cette étape :

- Les sons
- Les objets invisibles de contrôle du jeu

2.2. Principe des éléments

Le jeu de compose de :

- 1 pièce
- 3 sprites (mur, personnage et diamant)
- 4 objets (mur, personnage, diamant et compteur de diamant)
- 1 son (beep)

Le personnage de déplace de la même façon que dans le premier jeu.

2.3. Création du jeu

Création des sprites

Nous utilisons les sprites précédents auxquels nous ajoutons le diamant.

Sprite Properties		
Name: Diamant	Precise <u>collision</u> checking	11
etiro2 beal 🕾	Smooth edges	
	Preload texture	
Width: 16 Height: 16	Origin	
Number or subimages: 1	X O Y O	
	Center	
🍪 <u>E</u> dit Sprite	Bounding Box	
Transporent	Automatic O Full image	
I I I I I I I I I I I I I I I I I I I	<u>O</u> <u>M</u> anual	
	Left Bight 15	
✓ <u>0</u> K	<u>T</u> op 0 <u>B</u> ottom 15	

Création des sons

Les objets sont utilisés en fonction des évènements se produisant pendant le jeu. Ils sont associés en général à un fichier wave qui contient déjà le son.

Un objet est crée par le menu arrête>Sound ou par Ctrl Alt U. Nous utiliserons ici un seul son « beep » associé au fichier beep1.sound



Création des objets

Le personnage et les briques du mur sont identiques à ceux de la phase précédente. Nous allons créer le diamant

Le diamant

Les diamants sont dispersés dans le parcourt. Ils sont détruits et rapportent des points dès que le personnage les touche

Object Properties			_ 0	×
Name: Diamant Sprite Diamant New Edit ✓ Visible Solid Depth: 0 Persistent Parent: Mask: <same as="" sprite=""> ● Show Information</same>	Events: Personnage Add Event Delete Change	Actions: Destroy the instance Play sound beep Set the score relative to 10 Draw the value of score	- Move - Move	K move main1 main2 control score extra draw
		1		

Nous avons ici un seul évènement

- Lorsque le personnage touche un diamant
 - Le diamant est détruit
 - o On émet un beep

-

- On rajoute 10 au score
- o On réaffiche la valeur du score

Le compteur de diamants

Le compteur de diamants est un objet factice. Il sert à vérifier à chaque étape le nombre de diamants restant en jeu. Lorsqu'il n'en reste plus, le niveau est terminé

Object Properties				×
Name: comptdiam Sprite (no sprite) New New Visible Solid Depth: 0 Persistent Barent: <no parent=""> Mask: (same as sprite) Show Information</no>	Events: Step Add Event Delete Change	Actions: Start of a block Set the score relative to 50 Final for the score relative to 50 Final fo	-Score	move main1 main2 control score extra draw

Nous avons ici un seul évènement

- A chaque pas
 - Si le nombre de diamants restant est égal à 0
 - On rajoute 50 au score
 - Si la prochaine pièce existe
 - On y va
 - Sinon
 - La partie se termine

Création de la pièce de jeu

Ici la pièce de jeu comprendra 1 personnage, des briques représentant un parcours, des diamants disséminés, et un compteur de diamants (invisible)

🖻 Room Properties				
✓ 🖌 🗋 🔿 😫 😫 🔮) 💪 Snap 🖄 🕯	6 Snap <u>Y</u> : 16 🔠	≫ 🔎 •	
backgrounds views splicits settings jiles Diject to add with left mouse: Compliant is and + 4.4br = no ange + 4.4br = no ange - 4.4br = no a				×
	X: V:	object:	id:	

2.4. Test du jeu



La partie se termine lorsque le personnage a atteint le drapeau ou lorsqu'on appuie sur la touche 'escape'

3. Troisième étape : On rajoute un monstre qui se déplace dans la labyrinthe

3.1. But du jeu

Le but de ce premier jeu est de créer un personnage qui se déplace dans un labyrinthe. Il mange des diamants et doit atteindre un drapeau. Le drapeau est caché par une brique qui disparaît quand tous les diamants ont été mangés. Le jeu s'arrête quand le personnage a atteint le drapeau.

Nous apprenons ici à créer :

- Des monstres qui se déplacent
- Des portes qui disparaissent au cours du jeu
- La gestion des vies

3.2. Principe des éléments

Le jeu de compose de :

- 1 pièce
- 5 sprites (mur, porte, personnage, diamant, drapeau et monstre)
- 6 objets (mur, personnage, diamant drapeau, monstre, compteur de diamant, Contrôleur de vies et de score, contrôleur de démarrage)
- 3 sons (beep, touché, et brique rouge détruite)

Le personnage de déplace de la même façon que dans le deuxième jeu. Lorsque tous les diamants ont été mangés, la porte s'ouvre, donnant accès au drapeau pour finir la partie.

3.3. Création du jeu

Création des sprites

Nous utilisons les sprites précédents auxquels nous ajoutons :

- La porte
- Le fantôme
- La vie

Construction Const			
Name: Porte	✓ Precise <u>c</u> ollision checking		
🔁 Load Sprite	Smooth edges		
	Preload texture		
Width: 16 Height: 16	Origin		
Number of subimages: 1	X O Y O		
	Center		
👶 <u>E</u> dit Sprite	Bounding Box		
Transaction	Automatic C Eull image		
I	<u> </u>		
	Left 0 <u>R</u> ight 15		
<u>✓ </u> <u>□</u> K	<u>I</u> op 0 <u>B</u> ottom 15		

Sprite Properties		_ 🗆 🗡
Name: Fantome	Precise <u>c</u> ollision checking	
Market Load Sprite Width: 16 Height: 16 Number of subimages: 1	© Smooth edges ✓ Preload texture Origin × 0 Y 0 Center	
<u>≩</u> <u>E</u> dit Sprite I ransparent <u>V</u> <u>D</u> K	Bounding Box Image Automatic Image Image	
Contemporaties		<u>_</u> _×
Name: Porte	Precise <u>c</u> ollision checking	
Coad Sprite Width: 16 Height: 16 Number of subimages: 1	© <u>S</u> mooth edges ✓ <u>P</u> reload texture Origin <u>×</u> 0 <u>×</u> 0 <u>Center</u>	
Edit Sprite	Bounding Box © Automatic <u>F</u> ull image Manual Left 0 Right Top 0 Bottom	

Création des sons

Nous allons rajouter 2 sons :

- Touché, qui indique que le personnage a été touché
- Porte ouverte qui indique que tous les diamants ont été mangés et que la porte disparaît



Création des objets

Le personnage, les briques du mur, le drapeau, le diamant et son compteur sont identiques à ceux de la phase précédente. Nous allons créer les fantômes, le contrôle de la pièce, le démarrage, et la porte

Le fantôme horizontal

Le fantôme vertical comme son nom l'indique se déplace à la verticale. Il fait demi tour lorsqu'il rencontre un mur

	Object Properties			_ 0	×
	Name: Fantome Horiz Spite	Events: ♀ Create ► ■ mur ► ♀ ♀ Personnage ► ■ porte	Actions:	- Move	move main1 main2 control score extra dr.
:	<u>✓ 0</u> K	Add Event Delete Change		Jeps	WE

Nous avons ici 4 évènements

- A la création
 - o Il commence à bouger vers la droite
- Quand il rencontre un mur
 - o Il fait demi tour



- Quand il rencontre le personnage
 - o On émet un son ; on attend 1 seconde
 - o On enlève une vie
 - o On redémarre la pièce courante



- Quand il rencontre une porte : identique au mur

Le fantôme vertical

Il est identique au fantôme horizontal si ce n'est qu'il commence à bouger vers le haut

Le contrôleur de pièce

C'est un objet fantôme. Il n'apparaît pas vraiment pendant le jeu. Il sert à afficher le score et les vies. Lorsqu'il ne reste plus de vie, c'est lui qui affiche les meilleurs scores et démarre une nouvelle partie.

🖸 Object Properties			
Name: ContrôlePiece Sprite Kno sprite> New ✓ Visible Solid Depth: 0 Persistent Parent: Mask: ③ Show Information ✓ DK	Events: No More Lives Draw Add Event Delete Change	Actions: Set font to Fonte de score Draw the value of score Draw a text Draw the lives as image	Move Move maint main

Nous avons ici 2 évènements

- A l'affichage
 - Définit la fonte du score et affiche le score
 - o Affiche « nombre de vies »
 - o Affiche les vies restantes
- Lorsqu'il ne reste plus de vies
 - o On affiche la table des meilleurs scores
 - o On redémarre la partie



Le contrôleur de démarrage

Ce contrôleur est aussi un objet fantôme. On le place dans la première pièce. Il met le score à zéro et le nombre de vies restantes à 3

Nous avons ici un seul évènement à la création

Les portes

Les portes sont là pour protéger le drapeau. Elles disparaissent lorsque tous les diamants ont été mangés.



Nous avons ici un seul évènement

- A chaque pas
 - Si le nombre de diamants restants est égal à 0
 - La porte se détruit elle-même
 - On produit un son correspondant à l'ouverture de porte

Création de la pièce de jeu

Ici la pièce de jeu comprendra 1 personnage, des briques représentant un parcours, des diamants disséminés, un drapeau protégé par une ou plusieurs portes.

En dessous du parcours, on mettra un contrôleur de pièce et le contrôleur de démarrage

C'est sur cette partie là que les jeunes peuvent être imaginatifs. On leur laissera un maximum de temps pour créer leur pièce



4. Quatrième étape : On rajoute des blocs de pierre qui bloquent le personnage, et on crée des fantômes qui changent de direction

4.1. But du jeu

Le jeu est identique aux précédents si ce n'est que l'on rajoute des blocs de pierre que l'on ne peut déplacer qu'à l'unité (on ne peut déplacer 2 blocs à la fois). On crée aussi des fantômes qui changent de direction tous seuls

On apprend à créer ici :

- Des blocs que l'on peut déplacer
- Des fantômes qui changent de direction

4.2. Principe des éléments

Le jeu de compose de tous les éléments précédents plus :

- des blocs de pierre
- des fantômes changeant de direction

Le personnage de déplace de la même façon que dans le troisième jeu. Lorsque tous les diamants ont été mangés, la porte s'ouvre, donnant accès au drapeau pour finir la partie. On a en plus des blocs de pierre qui peuvent bloquer les fantômes

4.3. Création du jeu

Création des sprites

Nous utilisons les sprites précédents auxquels nous ajoutons :

- Le bloc de pierre

C Sprite Properties			
Name: Bloc_pierre	✓ Precise <u>c</u> ollision checking	6	
<mark>⊘</mark> Load Sprite Width: 16 Height: 16 Number of subimages: 1	Smooth edges		
	Preload texture		
	Origin		
	X 0 Y 0		
	Center		
👶 <u>E</u> dit Sprite	Bounding Box		
Transa areat	Automatic C Full image		
	Manual		
<u>✓ 0</u> K	Lent 0 <u>Bight 15</u> Lop 0 <u>B</u> ottom 15		

Création des sons

Nous allons rajouter 1 son qui indique qu'un bloc de pierre a été déplace.

Création des objets

Nous allons créer les fantômes changeant, et les blocs de pierre

Le fantôme changeant de direction

Le fantôme vertical comme son nom l'indique se déplace soit à la verticale soit à l'horizontale. Il fait demi tour lorsqu'il rencontre un mur, ou on objet.

De plus, il change de direction de façon aléatoire

Object Properties			<u>_ 🗆 ×</u>
Name: Fantome Tous sens Sprite Fantome New Edit Visible Solid Depth: 0 Persistent Barent: (no parent) Mask: (same as sprite) Show Information	Events: Create Step Create Personnage	Actions: If an expression is true Start moving in a direction Else Start moving in a direction	Move main1 main2 control score extra draw

Nous avons ici 7 évènements

- A la création
 - o Il commence à bouger vers la droite
- Quand il rencontre un mur, un diamant, une porte ou un bloc de pierre
 - o Il fait demi tour



- Quand il rencontre le personnage
 - o On émet un son ; on attend 1 seconde
 - o On enlève une vie
 - o On redémarre la pièce courante



Le bloc de pierre

🖸 Object Properties			
Name: BlocPierre Sprite Bloc_pierre New Edit ✓ Visible Solid Depth: 0 Persistent Bask: <same as="" sprite=""> Image: Show Information</same>	Events: Personnage Add Event	Actions: If a position is collision free Start of a block Play sound pousser Jump to a given position Control for the block	Move main move main main main main main main main main
If a position is collision Applies to ○ C Self ○ Other ○ Object: x: 400 y; 4*0t objects: All V F V DK	Inter Inspect	Jump to a given position Applies to G Self Other Object: x 4*other hspeed y 4*other vspeed V Relative	× Cancel

Nous avons ici 1 évènements

- A l'affichage
 - Définit la fonte du score et affiche le score
 - o Affiche « nombre de vies »
 - o Affiche les vies restantes

Création de la pièce de jeu

Ici la pièce de jeu comprendra 1 personnage, des briques représentant un parcours, des diamants disséminés, un drapeau protégé par une ou plusieurs portes.

En dessous du parcours, on mettra un contrôleur de pièce et le contrôleur de démarrage

C'est sur cette partie là que les jeunes peuvent être imaginatifs. On leur laissera un maximum de temps pour créer leur pièce

