



Projet ISN : création jeu vidéo

« LABYRINTHES ALEATOIRES »

Table des matières

Introduction.....	2
Présentation de l'Equipe créatrice du projet ISN.....	2
1. Choix du projet.....	2
2. Progression.....	3-8
2.1. Premier groupe	3-6
2.2. Second groupe	7-8
3. Bilan : difficultés rencontrés et perspectives possibles.....	8-10
3.1. Difficultés rencontrées.....	8-9
3.2. Perspectives.	9-10
Conclusion.....	10
Annexe.....	11

Introduction.

L'ISN (Informatique et Sciences du Numérique) consiste tout au long de l'année en l'acquisition de diverses compétences qui amélioreront d'une part l'étendue de notre savoir informatique et, d'autre part, le travail en groupe puisque l'élaboration de deux projets (l'un en milieu d'année et l'autre en fin d'année qui sera évalué pour le BAC) a été réalisé. Mon envie de créer mon propre jeu vidéo a été spontanée étant donné que je compte me diriger vers ce domaine pour exercer mon futur métier. Ainsi, cette matière se révèle comme les prémices de mon initiation à la programmation informatique.

Présentation de l'Equipe créatrice du projet ISN.

- LE PAPE Sandy.
- CORAY Yoann.
- PICARDA Tiphaine.
- BERROU Katel.
- DEHAENNE Isabelle.

1. Choix du projet.

La 1ère séance fut consacrée à la réflexion du projet et surtout, à son type (jeu, représentation des nombres, codage numérique, algorithme de tri, ...). Nous voulions, tout comme les mini-projets, réaliser un jeu mais qui, cette fois-ci, serait plus élaboré. Durant la séance, plusieurs idées furent proposées comme le Tetris, le Casse-briques, un jeu d'aventure type Mario et ... un labyrinthe. Après un débat houleux ayant mêlé souvenirs d'enfance et désirs présents, nous avons décidé d'adopter, malgré quelques réticences de la majorité du groupe qui aurait préféré réaliser un casse-briques (que Yoann a réalisé de son côté hors séances ISN), le labyrinthe. Le labyrinthe fut quelque chose de rapide à visualiser mais le concevoir fut une autre paire de manches !



Le professeur, dans un premier temps, nous a conseillé de faire appel à Processing, un logiciel de programmation associé à un aspect graphique simple.

2. Progression.

Ainsi, dès la séance suivante, nous avons décidé de partager les tâches à réaliser : la distinction de deux équipes a été naturelle : moi et Yoann avons décidé de nous charger du labyrinthe en lui-même, c'est-à-dire les chemins à tracer et l'aspect graphique tandis qu'Isabelle, Katel et Tiphaine se sont chargés du déplacement du personnage et de ses interactions avec le décor.

2.1. Premier groupe

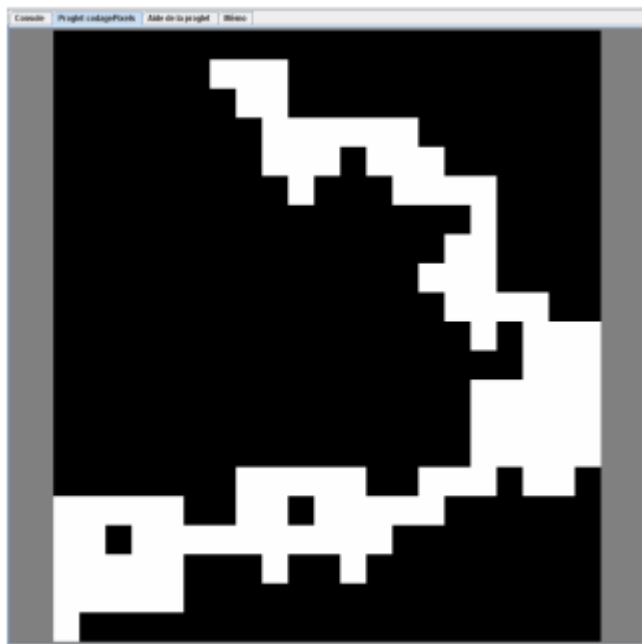
Très vite, le professeur nous a conseillé d'ajouter au labyrinthe une dimension aléatoire au labyrinthe de base (non aléatoire donc). Ce fut la stupéfaction au sein du groupe. En effet, ce ne fut absolument pas dans notre projet initial : comment pouvait-on penser à un labyrinthe aléatoire alors que les labyrinthes que nous jouions étant petit n'étaient pas aléatoires ! Cela nous semblait déjà compliqué de créer un labyrinthe normal mais cette notion supplémentaire rajouta une seconde difficulté. Il faut savoir que le labyrinthe fut déjà construit sans la dimension aléatoire à ce moment-là en utilisant la fonction `for` et l'ajout de formes géométriques pour des murs plus fins. Mais indéniablement, cette notation incluait



forcément un code très long et l'intérêt fut proche du néant en terme de programmation, si ce n'est la parfaite acquisition de la compétence : "savoir utiliser la fonction for" et "savoir tracer une forme géométrique". Néanmoins, ce labyrinthe, que j'ai réalisé entièrement, m'a permis de bien maîtriser comment on pouvait insérer une image du web dans le labyrinthe plutôt que d'avoir recours à un graphique uniquement composé de couleurs aussi fades les unes que les autres. Suite à quelques séances, le groupe s'est concerté : nous avons finalement décidé de faire un jeu type labyrinthe, un jeu ressemblant à Packman en somme, qui se génère aléatoirement,.

Nous avons décidé, Yoann et moi, de scinder davantage le projet que nous nous étions fixés pour un peu plus d'un mois maintenant afin d'être plus efficace (la dimension aléatoire du labyrinthe). Yoann s'est davantage occupé de traduire du papier à l'ordinateur ce que nous souhaitions. Il est indéniable que ce fut le meilleur programmeur du groupe en ce qui concerne Java's Cool. De plus, sa vision relativement claire des mathématiques nous a permis très rapidement d'obtenir un labyrinthe potable. Pour ma part, je me suis chargé davantage de l'aspect graphique du labyrinthe étant donné que je souhaite travailler dans l'aspect graphique dans le milieu de jeu vidéo plus tard. Je maîtrisai mieux le langage Processing, ainsi que les notations nouvelles, qui pouvaient différer de Java's cool (notamment les « void » beaucoup plus nombreux). La tâche fut

difficile à accomplir mais la cohésion de notre groupe a permis une avancée relativement rapide. Après de longues journées de réflexion, cette fameuse dimension aléatoire du labyrinthe est vite devenue un calvaire. En effet, nous ne savions pas du tout par où commencer et surtout, comment ! C'est pourquoi le professeur est intervenu et nous a conseillé d'utiliser la notation matricielle et probabiliste pour

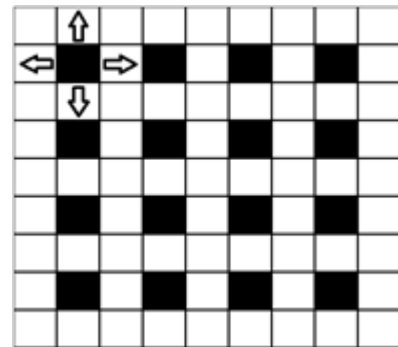


avancer. Nous devons bien admettre que cette indication ne nous a pas permis de résoudre

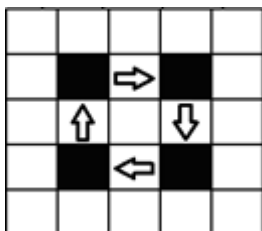
immédiatement le problème. Yoann a formé le premier labyrinthe aléatoire qu'il a transféré avec mon aide de Java's Cool à Processing. Mais le problème fut le chemin très grossier que celui faisait apparaître.

Puis ce fut un déclic ...

Pendant les vacances, suite à des tests interminables et sans résultats concluant, j'eus une idée : pourquoi ne pas décider de créer un labyrinthe où 1 matrice sur 2 ne seraient pas aléatoire (mur) et leur autoriser chacune à générer 1 matrice (mur) collée à la matrice (mur) non aléatoire ! L'idée est bien inaccoutumée, je vous l'accorde ! Ainsi, bien que des petits soucis fussent directement perceptibles, nous obtiendrions un labyrinthe semi-aléatoire. Nous pouvons résumer l'idée de la forme ci-contre :



La matrice mur (en noir) génère 1 autre matrice mur afin de créer le labyrinthe aléatoire, la flèche représentant le possible emplacement où la matrice mur va en générer une seconde. Ainsi, il fut très facile de concevoir rapidement un labyrinthe digne de ce nom. Yoann s'est assuré qu'un chemin entre l'entrée et la sortie du personnage était possible. En effet, l'idée ci-dessus peut parfois créer des problèmes, le plus fréquent étant la génération en carré de 4 matrices mur, les flèches représentant l'emplacement où la matrice mur va en générer une seconde.



Le problème, c'est quand la case « arrivée » se trouve, par un malheureux hasard, à l'emplacement de la case entourée par des matrices mur (fixes et générées).

Par suite, on a remarqué que le test qui assurait au moins un chemin possible pour finir le labyrinthe ne marchait qu'une fois sur trois essais environ. En effet, le test sortait du tableau préétabli et, de ce fait, empêchait son démarrage. Yoann, suite à de longues heures

de travail, a réussi à trouver le problème : la fonction « random » posée se révéla assez ambiguë (voir annexe).

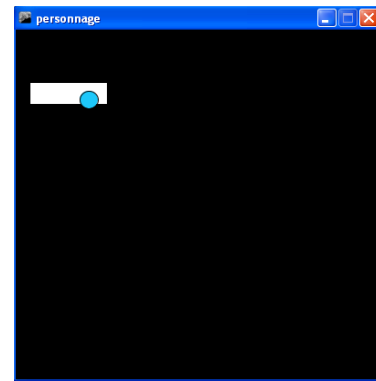
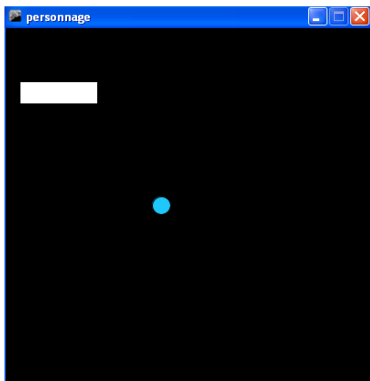
Pour ma part, je me suis occupé majoritairement de l'aspect graphique du labyrinthe. Une série de décors différents (forêt, désert, ville, volcans, plage,...) permet de rendre le labyrinthe plus vivant et plus agréable. Pour ce faire, l'intégration d'images provenant d'internet a été indispensable. Il a fallu déclarer une variable comme image, la charger dans le dossier où elle a été enregistrer et enfin, adapter une bonne partie de la notation matricielle afin qu'elle fasse apparaître une image et non plus une couleur (fonctions stroke, fill,... plus nécessaires). Il a fallu, pour obtenir une matrice imagée, sélectionner un morceau d'image puis le redimensionner à l'aide de « Gimp » en 800x800 et l'enregistrer sous fichier « .gif ». Ainsi, nous avons fait du « Tile Mapping » sans le savoir puisque notre vœu premier était de se déplacer sur un fond qui recouvrirait toutes les matrices plutôt que l'image de fond pour chaque matrice.

Le plus difficile à réaliser dans ce domaine a été l'identification d'un smiley comme personnage se déplaçant. Auparavant, les filles avaient eu recours à l'utilisation d'un disque coloré qu'il suffisait de diriger. Mais la taille et la nature même de l'image a fait que les coordonnées de l'image étaient sensiblement différentes que celle de la forme géométrique. De ce fait, le personnage interagissait mal avec les murs, l'arrêt n'est pas immédiat (déplacement du personnage possible sur une partie du mur). Une modification des coordonnées du déplacement du personnage a été nécessaire mais pas évidente. Le professeur nous avait conseillé d'utiliser la fonction matrix qui avait le même rôle mais le peu de temps qui nous restait nous a seulement permis d'obtenir des bugs. Enfin, la dernière séance a été consacré à la finalisation du projet : pour ma part, je me suis occupé de nourrir la variété des paysages possibles à sélectionner, concevoir un arrière-plan de menu principal et enfin, confectionner des pièges s'identifiant à chaque paysage (crabes pour la plage, flammes pour les volcans,...). Yoann, en revanche, a essayé de résoudre un problème de souris relié au programme : sans succès. Par moment, le programme ne semble plus réagir à l'action

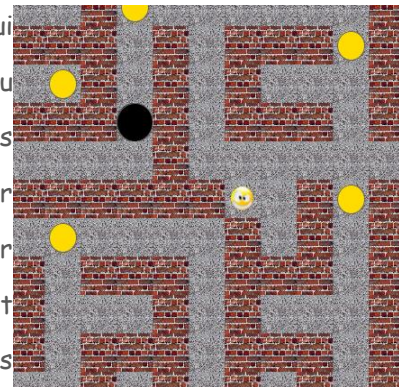
de la souris sur des textes, comme s'il était en pause permanente, il faut alors redémarrer le programme. Il est également prêter main forte à chaque poste.

2.2. Second groupe

Tiphaine, Katel et Isabelle se sont occupées du déplacement du personnage et ce ne fut pas une simple affaire ! Un peu plus d'un mois a été nécessaire pour réaliser un déplacement correct. En connectant les touches du clavier (flèches) et l'ordinateur, nous avons pu, par l'intermédiaire du programme, à diriger le personnage et le faire déplacer en fonction des touches appuyées. La grande difficulté des filles fut non pas tant de déplacer le personnage mais de le faire arrêter quand il rencontre un mur, il passait souvent à travers à vrai dire.



Après de nombreux tâtonnements, les filles ont réussi à bloquer le personnage quand il rencontre un mur à l'aide de pixels. La fonction « get », qui permet d'effectuer des tests de couleurs autour du personnage, a été utilisée afin de se baser sur les couleurs pour les interactions avec les murs. Par suite, le professeur nous a conseillé de se baser sur des matrices plutôt que sur les couleurs, c'est pourquoi une nouvelle difficulté est apparue : bloquer le personnage en se basant sur les matrices puis intégrer ce programme dans le programme de processus aléatoire du labyrinthe. A l'aide



de Yoann, Tiphaine, Katel et Isabelle ont réussi à résoudre le problème pendant que je travaillais sur l'aspect graphique du labyrinthe non aléatoire.

Ensuite, elles se sont chargées de la construction de pièges tel que le puis par exemple et ont enchaîné, comme je leur avais demandé, à remplacer les trappes qui s'ouvraient en fonction du temps (toutes les 3 secondes) par de dangereux animaux (crabes qui ouvrent et ferment leurs pinces et crocodiles qui ouvrent et ferment leur gueule).

3. Bilan : difficultés rencontrées et perspectives possibles.

3.1. Difficultés rencontrées

Durant cette longue période où nous avons mené le projet à bien, de nombreuses difficultés sont venues nous faire obstacle : nous les avons tous résolus. En voici la liste récapitulative :

- En fonction de la taille des différents écrans du lycée, nous avons dû adapter le programme.
- La dimension aléatoire du programme fut difficile à concevoir.
- Stopper le déplacement du personnage dès qu'il rencontre un mur du labyrinthe.
- Confusion avec un nombre important de variables.
- La fonction « get » ne supporte pas les calculs dans l'appel de la fonction. Cela nous a obligé à faire des contournements pénibles (appel d'autres variables,...). Ce ne fut plus un problème quand nous sommes passés à la notation matricielle.
- La fonction random nous a joué bien des tours (sa notation a été trompeuse), ce qui fait que le test sortait une fois sur trois du labyrinthe, d'où l'impossibilité de son exécution.

- Le lien à construire entre le clic de la souris sur le texte et l'envoi à une autre fenêtre.
- Exécuter un test qui assure la possibilité de finir le labyrinthe (l'existence d'au moins un chemin possible de l'emplacement initial du personnage à la sortie sans pour autant avoir l'assurance de pouvoir récupérer toutes les pièces).

Après mûre réflexion, nous avons décidé, certes, de réaliser un labyrinthe mais en lui ajoutant une dimension aléatoire, c'est-à-dire qu'à chaque fois qu'on exécuterait le programme, il se générerait aléatoirement (chemin tracé aléatoirement tout comme le positionnement des pièces, des pièges ou des monstres).

3.2. Prolongements possibles.

Malheureusement, bien que nous ayons eu de nombreuses séances d'ISN pour mener à bien notre projet, nous aurions souhaité aller plus loin dans la finition du labyrinthe. A la base, nous nous étions fixés comme objectif quelque chose de bien plus modeste mais dès que l'idée de la dimension aléatoire du labyrinthe fut trouvée, on a pu accélérer notre progression. En raison du temps trop court et d'une difficulté parfois trop élevée, nous n'avons pas réalisé les points suivants :

- Un palmarès qui serait basé sur les meilleurs temps avec le plus de points.
- Un menu d'options récapitulant le fonctionnement des pièges, le but du jeu (bien que qu'il soit relativement aisé de le comprendre).
- Une musique de fond et des bruitages que nous aurions pu importer à Processing puisque ce logiciel n'est pas prévu pour à la base (Isabelle et moi avions cherché à émettre un son du programme mais il resta muet. Les enregistrements de sons que j'avais faits à l'aide de mon synthétiseur n'ont donc servi à rien).
- Un graphisme de fond un peu moins répétitifs (masquer les connections des matrices).
- Mettre en ligne le labyrinthe et avoir la possibilité de le jouer sur Android par exemple.

- Une animation des pièges (animaux,...) un peu trop brutale (pas assez limpide),...
- Un test qui permettrait de s'assurer de la possibilité de prendre tous les artefacts dans un niveau (cœur et pièces).

Conclusion.

Ce fut une expérience nouvelle et agréable. J'ai mené un projet qui m'a permis de progresser tant dans la programmation qu'à la modification de l'image. Ce ne fut pas facile mais la cohésion du groupe et l'exploitation des compétences de chacun a permis d'obtenir un labyrinthe aléatoire satisfaisant à notre niveau. Je tiens à remercier toute l'équipe qui s'est beaucoup investi alors qu'elle n'était pas notée (étant donné qu'ils font l'option de spécialité maths) ainsi que le professeur d'ISN, M.TYMEN, qui nous a suivis tout le long du projet.

Annexe (copier-coller du programme).