

## **Création de jeux vidéo paroles d'amateurs**

**Author :** pyhurel

**Date :** 14 octobre 2014

**Dans le dernier numéro de C4 (220), nous vous racontions le défi que l'asbl «D'une Certaine Gaieté» s'était lancé à travers un nouvel atelier multimédia : rassembler une dizaine de personnes néophytes en programmation pour créer un prototype de jeu vidéo. Son déroulement s'est construit à la croisée de la création assistée et du do it yourself, du serious game et du jeu traditionnel, de l'humour et de l'engagement politique. Douze séances de trois à quatre heures, une release party et une couverture de l'affaire par La Meuse plus tard, revenons sur cette aventure à travers quelques propos entendus lors de l'atelier.**

**« Ce n'est pas une obligation, mais on peut parler du chômage... ? »**

Avant de commencer l'atelier, j'avais fait quelques recherches sur les thèmes de prédilection de l'asbl «D'une Certaine Gaieté». Aucun jeu vidéo, à ma connaissance, ne s'était jusqu'ici emparé du thème du chômage. La situation des participants de l'atelier – sans-emploi, étudiants, travailleurs intermittent – ainsi que la récente législation visant à exclure des dizaines de milliers de chômeurs au 1er janvier 2015 ont sans doute joué dans l'adhésion spontanée à l'idée proposée.

De là, plusieurs idées furent avancées : allait-on faire de la plate-forme (à la Mario Bros) ? Proposer de gérer des chômeurs ? Tester les réflexes du joueur en le mettant dans la peau d'un employé de l'Onem ? À ce stade, notre ambition était de créer un prototype. Techniquement, n'ayant pas pour but d'aboutir à une production finalisée, nous n'avions aucune limite dans le système à créer. Enfin, quasiment aucune : le logiciel utilisé, *Construct 2*, permet de créer des jeux en 2D, mais pas question de viser une progression dans un univers en trois

dimensions.

## « **Donc, les chômeurs avanceraient comme des zombies ?** »

À la question du chômage est venue s'ajouter la volonté d'incarner un « méchant », ainsi que l'envie de créer un *tower defence* : le joueur a la position d'un stratège qui doit se défendre d'une invasion (une armée ennemie, extra-terrestres, ou... chômeurs). Les envahisseurs se déplacent le long d'un chemin décidé à l'avance et le joueur peut placer des unités défensives sur les côtés pour tuer les vilains, les ralentir, etc.

Les participants se sont non seulement approprié les techniques informatiques, mais se sont aussi *réapproprié* un sujet de société traitant la « chasse aux chômeurs » au premier degré – pour en rire ! Le « pitch » final se révèle alors provocateur et simple : « Dans un futur lointain (2015), le Parlement d'une Belgique devenue totalitaire demande des mesures radicales contre les chômeurs : fini les contrôles et l'oppression, place à la suppression ! ». Vous avez à votre disposition des unités de police et de l'armée pour tuer ces hordes incessantes de chômeurs cherchant à rejoindre la place pour y démarrer leur révolution. Le Forem et l'Onem vous aideront aussi – grâce à la diffusion de propagande et d'une paperasse interminable – à ralentir les ardeurs de ces satanés tire-au-flanc. Chômeur Blaster propose donc un second degré *trash* qui repose sur une complicité auteurs-joueurs.

## « **OK, mais il faut quand même un score!** »

Certains *serious games* négligent la recherche de fun, de plaisir à jouer. D'autres l'abandonnent en frustrant volontairement le joueur : l'obliger à vivre des situations pénibles ou à perdre sont alors autant de moyens pour transmettre un message. En atelier, cependant, les participants – des passionnés du jeu vidéo – souhaitaient non seulement rendre Chômeur Blaster engagé et drôle sur le fond, mais aussi amusant à jouer. Il fallait que le défi du score motive le joueur, que les actions possibles à jouer se débloquent au fur et à mesure et qu'il y ait une véritable évolution dans la difficulté. S'il est impossible de gagner la partie, le joueur étant tôt ou tard débordé par les manifestants, la barre de score offre la possibilité de se mesurer à l'ordinateur ou de jouer entre amis.

## « **Oui oui j'écoute, je finis juste ma partie** »



Il n'y avait aucun doute lors de l'atelier : les participants étaient bien des joueurs. On a non seulement rapidement abandonné le deuxième module (récit interactif textuel), mais chacun se laissait aussi distraire par son propre prototype. Avant le travail final, qui devait se faire nécessairement depuis un seul ordinateur, chacun avait sa propre version du prototype, qu'il travaillait et développait au fil des séances. Lors d'une des premières séances, notre but était de réussir à faire apparaître un personnage (un carré bleu) et de lui faire suivre un chemin indiqué. Jusqu'ici, rien de très compliqué : le programme permet d'indiquer que telle image aura un comportement de type « chercher un chemin » (*pathfinding*), que telle autre sera un obstacle sur sa route, etc. Étape suivante : permettre au joueur de positionner une tourelle (un carré rouge) qui va tirer automatiquement des balles (un carré jaune) sur les personnages. Dès cette étape réussie, chacun avait entre les mains la base du jeu avec un défi ludique à relever. C'est à partir de là que l'on sentait conjointement le plaisir de jouer à sa propre création et la curiosité s'étendre : « *Il faudrait qu'on limite le nombre de tourelles à mettre maintenant!* », « *Comment peut-on faire pour que les personnages n'apparaissent pas tous en même temps ?* »

## « La division euclidoienne ? »

La programmation peut donc se résumer à deux phases : se poser un problème et chercher à le résoudre. Il est ici impossible d'expliquer l'ensemble de nos 400 lignes de codes (bientôt accessibles en ligne sur [entonnioir.org](http://entonnioir.org)), mais tentons tout de même d'illustrer le genre de problèmes surmontés. Une de nos envies était de faire venir les ennemis par vagues. D'abord, deux chômeurs débarquent, et plusieurs secondes plus tard, ils arrivent à quatre, puis huit, et ainsi de suite. Il nous a fallu ici créer des « variables » – des données particulières qui ont un

nom et certaines caractéristiques. Ainsi avons-nous le NombreEnnemiVoulus et le NombreEnnemisCréés. À chaque fois que l'ordinateur crée un ennemi, on ajoute +1 à la variable correspondante. Quand le nombre d'ennemis créés est égal au nombre d'ennemis voulus, alors on lance le système de pause entre les deux vagues, on augmente le nombre d'ennemis voulus (la prochaine vague sera plus grande) et on remet le nombre d'ennemis créés à zéro (pour recommencer le comptage). Et, au passage, on ajoute +1 à une autre variable qui compte le nombre de vagues. Avoir en stock une variable qui compte les vagues, cela permet de limiter certaines actions. Cela permet aussi de déclencher l'apparition des ennemis uniquement lors des vagues multiples (de 5, 10 ou 20) – ce qui suppose de se rappeler de ses cours de secondaire et de la division euclidienne...

## « Les images ? Elles viennent de Sprite Database »

Si je pensais qu'il était futile de donner une apparence particulière à notre prototype, il n'en allait pas de même pour tous. Certains participants ont rapidement été frustrés d'avoir à jouer avec des images abstraites en forme de carrés de couleur. Ils se sont donc débrouillés pour donner une apparence personnelle à leur travail : ici des pillages d'images sur Internet – dans un cadre privé uniquement – repris de références appréciées (Pokemon, Megaman, etc.) ; là, des images libres de droit. Après

quelques recherches sur Google, on tombe effectivement rapidement sur des images dont les licences prévoient un usage gratuit sans but commercial. Difficile de dire cependant s'il serait possible de réaliser une œuvre cohérente et correcte visuellement en utilisant uniquement des sources ouvertes. Un défi pour un prochain atelier ?

## « Pour l'unité de base, à quoi ressemble un chômeur ? »

Dans le cadre de cet atelier, nos ambitions ont été revues à la hausse : « *On a un très bon prototype. Pourquoi ne pas aller jusqu'au bout ?* » Nous avons eu recours à un dessinateur professionnel, Valentin Mélon. Celui-ci est venu discuter de l'ambiance souhaitée. « *On part sur quelque chose de sombre ou pas du tout ?* ». Se basant sur les *tower defence* classiques, les participants avaient mis au point quatre types d'ennemis : l'ennemi de base, le rapide, le résistant et le boss qui crée de nouvelles unités. Ceux-ci ont rapidement été associés aux stéréotypes des « catégories de chômeurs » : le jeune, la personne en chaise roulante, l'ancien combattant et la mère de famille. Alors que le sujet et la musique plongent le joueur dans une ambiance dramatique, le dessin est plutôt « mignon » : impossible de ne pas avoir d'empathie devant ces individus qu'on abat systématiquement.

Gregory Mertz, un des participants, a lui-même proposé de composer les musiques pour les différents niveaux. Sa principale référence : le son *chiptune* des premiers jeux vidéo.



## « J'avais une après-midi de libre, j'ai créé ceci... »

Dans un article sur la démocratisation de la programmation, nous nous interrogeons dans le numéro 220 de C4 sur l'arrivée possible de ce type de média dans la sphère privée. Si le cinéma était à ses débuts réservé à des professionnels, la Super 8 a rendu le matériel accessible et les films de vacances possibles. Le média vidéoludique, libéré de la complexité des langages de programmation grâce à des logiciels comme *Construct 2*, serait-il un support possible à des interactions privées ? Premier élément de réponse : au bout de quelques semaines, un des participants a spontanément partagé un petit programme qu'il avait réalisé pour faire un clin d'œil à une amie le jour du premier avril. Reprenant un principe ludique simple (attraper des éléments), le jeu faisait référence à des éléments personnels. Si l'on considère que les créateurs amateurs se comptent probablement en dizaines de milliers sur la planète, combien d'entre eux utilisent déjà le support vidéoludique comme un média personnel ?

## « C'est un rêve d'enfant... »

La version « beta » de Chômeur Blaster a été présentée à la péniche InsideOut le 21 mai dernier à Liège. L'occasion pour toute l'équipe de proposer au public de jouer à leur œuvre et de récolter des retours, des avis, des conseils – tout comme c'est le cas actuellement sur [entonnioir.org](http://entonnioir.org). Mais voir le public s'emparer de sa création, s'enthousiasmer ou rouspéter face à une difficulté ou à une surprise était encore une expérience différente. « *Ça donne envie de faire des jeux* » m'affirmait un participant, avant d'ajouter : « *C'est un rêve d'enfant qui vient un peu de se réaliser, en fait!* »

## Terrain paradoxal

Nous pensions créer un prototype, nous avons sorti un jeu vidéo. Celui-ci incite à *scorer*, comme c'est l'habitude de ce média, mais ne permet jamais de gagner, à l'image de beaucoup de *serious games*. Tout y est pensé pour que Chômeur Blaster soit amusant (musique, image, points, besoin de stratégie) mais aussi pour jeter un pavé dans la marre et dénoncer une chasse au chômeur déjà à l'œuvre. Cet atelier aura prouvé que l'on peut créer des œuvres vidéoludiques sans savoir

programmer – si on est prêt à y mettre beaucoup, beaucoup de temps et d'énergie – et que cela peut être très amusant ! À titre personnel, je me souviendrai longtemps de cette soirée banale où je testais sans entrain *Construct 2* par curiosité... et surtout de ce moment où le personnage à l'écran s'est mis à bouger quand je le lui indiquais. La programmation devenue accessible à tous ouvre de nouveaux champs – de perception ? De réflexion ? D'action ? N'hésitez pas à nous envoyer vos propres œuvres (cf. encadré) et à faire vivre, par vous-mêmes, ces nouveaux outils.