

Prog & Play

ou

apprendre à programmer en s'amusant

Stéphanie Combettes & Jérémie Guiochet (Enseignants)

Cassandre Anjorand & Florian Jeay (Etudiants)

Journée pédagogique de l'ARIUT Midi-Pyrénées

Pédagogie, Pédago-jeu : Comment les motiver avec les TICE

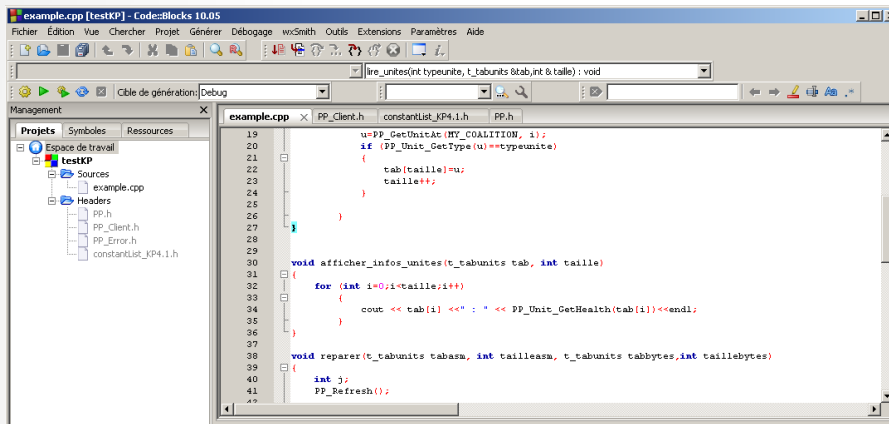
Figeac – 30 Avril 2015

Prog and Play : qu'és acco ?

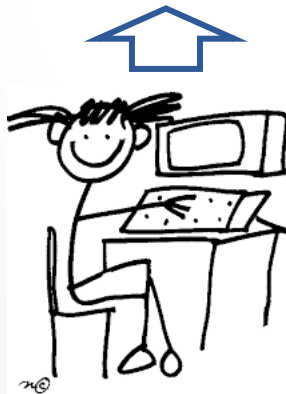
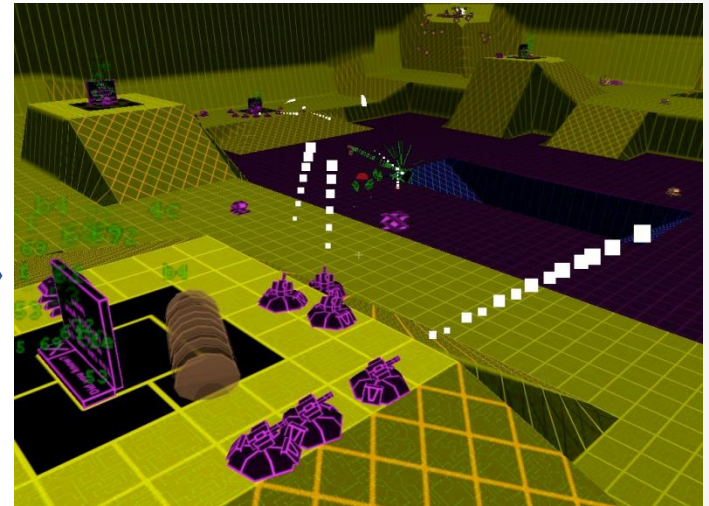
- Un jeu sérieux destiné à l'apprentissage de la programmation
 - Utilise un moteur 3D libre : **Spring** (mode KernelPanic)
 - Développé à l'IRIT (M. Muratet, F. Viallet & P. Torguet)
 - Utilisé dans plusieurs départements dont GElI
 - Donné lieu à des publications [STICEF -2014]
- Ecrire des programmes informatiques qui pilotent des unités dans un jeu de stratégie en 3D
 - Elaborer une stratégie gagnante et la mettre en œuvre
- ➔ Motiver les étudiants pour développer
 1. Des compétences en programmation
 2. De l'autonomie

[STICEF - 2014] M. Muratet, E. Delozanne, F. Viallet, P. Torguet : « Etude de l'intégration d'un jeu sérieux pour l'enseignement de la programmation dans différents contextes universitaires » - STICEF 2014

Prog and Play : qu'és acco (2)



```
19 void afficher_infos_unites(t_tabunits &tab, int &taille)
20 {
21     if (PP_Unit_GetType(u) == t_typeunite)
22     {
23         tab[taille]=u;
24         taille++;
25     }
26 }
27
28 void afficher_infos_unites(t_tabunits tab, int taille)
29 {
30     for (int i=0; i<taille; i++)
31     {
32         cout << tab[i] << " * << PP_Unit_GetHealth(tab[i]) << endl;
33     }
34 }
35
36 void reparer(t_tabunits tabasm, int tailleasm, t_tabunits tabbytes, int taillebytes)
37 {
38     int i;
39     PP_Refresh();
40 }
41
42
```



Prog and Play : contexte

- Étudiants en semestre de remédiation
 - Suite au redoublement d'un semestre ou
 - À une réorientation
- Dans le cadre de l'Apprendre Autrement
 - Heures supplémentaires par rapport aux Cours/TD/TP « classiques »
- Env. 1 groupe /an depuis 2012

Prog and Play : comment ?

- Travail seul ou en binôme
- 8 séances de 2h + examen final
- Documentation en anglais
- Pas de support de type sujet TP -> « missions » à réaliser
 - Avancement progressif en fonction du niveau de chaque étudiant
 - Chaque mission amène à appréhender un nouveau concept de programmation
 - Mission ultime -> confrontation des programmes sur une partie en réseau (entre les étudiants)

Exemple de mission

- [Prog&PlayFigeac.MOV](#)

Point de vue des étudiants

- Mise en pratique ludique : *jeu de stratégie => bon moyen de nous motiver pour travailler en s'amusant*
- Bonne compétition avec la mise en réseau des parties
- Prise en main facile avec des bibliothèques détaillées même si certaines missions sont difficiles
- Autonomie : liberté dans la structure du code

Bilan

Constats et Améliorations

- Evaluation classique en fin de cycle ne donne pas toujours de bons résultats :
 - nécessité d'effectuer un suivi personnel important
- Résolution de certaines missions en contournant les points théoriques
 - adapter les missions en fonction du niveau pour ne pas en bloquer certains (frustration)
- Côté jeu très prégnant au départ et réconfortant sur les missions simples : difficulté à revenir à une approche plus théorique pour résoudre des problèmes plus complexes :
 - Insuffler des éléments théoriques de façon progressive tout au long du cycle
- Très adapté à notre public (redoublants)
 - à valider pour débuter en programmation

Conclusion

- Motivation très forte des étudiants (même ceux en désengagement)
- Mise en autonomie sur les outils très rapide
 - Possibilité d'installer sur leur propre machine
- Très bons retours des étudiants
- Etre attentif à l'avancement (et à la façon de résoudre les problèmes) de chaque étudiant
 - Beaucoup de libertés

Merci de votre attention

