Nom : _____ **(** _____ **/100)**

Matricule : _____

Signature : _____

STR – Contrôle final

Lundi 25 avril 2016, de 9 h à 12 h, local D7-2022

Ceci constitue le contrôle final pour les cours ***IFT611*** et ***IFT729 – Conception de systèmes temps réel*** (STR). Il est présumé, avant d'attaquer la présente, que vous avez pris une part active à l'intégration des concepts du cours dans votre projet de session et que vous avez pris sur vous de faire les exercices qui, parmi ceux proposés, vous auront aidés à parfaire votre compréhension des éléments de matière qui vous semblaient un peu plus nébuleux.

Ce contrôle est récapitulatif, ce qui signifie qu'il couvre l'ensemble de la session (ce qui ne veut pas dire qu'il en couvre chaque parcelle, l'espace et le temps n'étant pas élastiques à ce point). En conséquence, gérez votre temps intelligemment.

Les consignes pour ce contrôle sont :

- le droit aux notes de cours est accordé;
- l'envoi de notes écrites, par télépathie, via pigeons voyageurs, hiboux, ou à l'aide de tout autre mode de communication, est interdit;
- la calculatrice est bannie (en fait, vous pouvez l'utiliser, mais ça ne donne honnêtement pas grand-chose), de même que le téléphone cellulaire, l'ordinateur portatif, le *Walkie Talkie* et leurs différents dérivés;
- si vous bloquez sur une question, alors passez à la suivante. Il est plus sage de revenir plus tard sur ce qui vous pose problème, s'il vous reste du temps;
- les questions de ce contrôle n'ont pas toutes le même poids. Planifiez la manière dont vous investirez temps et efforts;
- à moins d'indications contraires, vous pourrez répondre par du texte, du code, des schémas, etc., mais je ne pourrai corriger convenablement que si vos réponses sont claires. Il est donc à votre avantage de soigner le tout, et d'être aussi précis et limpide que possible;
- exprimez-vous. Attention toutefois aux grimaces qui captent l'attention;
- répondez aux questions, simplement. Chaque réponse sera corrigée en fonction de l'énoncé de la question, alors assurez-vous de bien répondre à ce qui est demandé, pas à ce qui aurait peut-être pu l'être. Évidemment, justifiez vos réponses!
- pas de stress, la vie est belle. Sans blagues. J'ai beaucoup apprécié ma session avec vous;
- savourez vos vacances… ou votre carrière… ou la poursuite de vos études… ou…
- je vous souhaite une fin de session pleine d'énergie!
- répondez directement sur le document, aux endroits prévus à cet effet.

Vous trouverez une annexe à la toute fin, un aide-mémoire, juste au cas… De même, l'examen permet d'accumuler 110 points, juste au cas où vous accrocheriez sur une question, mais la note finale sera plafonnée à 100.

Que de plaisir en perspective!

Q00 Des visages et des ambiances (_____/30)

Vos études universitaires viennent de prendre fin, diplôme en poche bien sûr, car le milieu de la R&D (un peu de 'R' et beaucoup de 'D') vous a ouvert ses portes. C'est bien connu, l'un des créneaux les plus importants au Québec est celui du divertissement, et c'est avec la firme K.E.G., pour *Karaoke Ergonomics Gaming*, que vous avez fait le choix de cheminer.

Une fois l'accueil fait selon les usages, avec café et rencontre des nouveaux collègues, on vous accompagne pour un bref tour des lieux. Cette visite comprend un survol du principal produit envisagé pour le proche futur de K.E.G., soit **FaceSync®**, un outil pour synchroniser les expressions faciales des joueuses et des joueurs de Karaoké avec celles des chanteurs et des musiciens originaux.

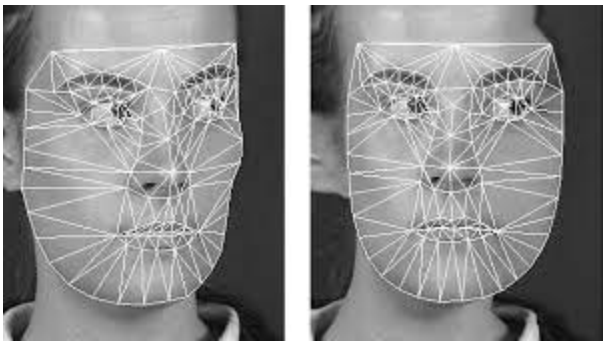

Figure 1 http://cvssp.org/faceweb/facialBiometric/

À travers son sophistiqué système de reconnaissance d'expressions faciales, K.E.G. vise à générer des pointages de conformité par joueur avec l'objet de son effort de mimétisme, mais aussi à créer des jeux qui adapteront les événements et l'ambiance proposée aux participantes et aux participants à l'émotion apparemment ressentie. K.E.G. estime pouvoir aller chercher un créneau d'interactivité jusqu'ici peu ou pas exploité.

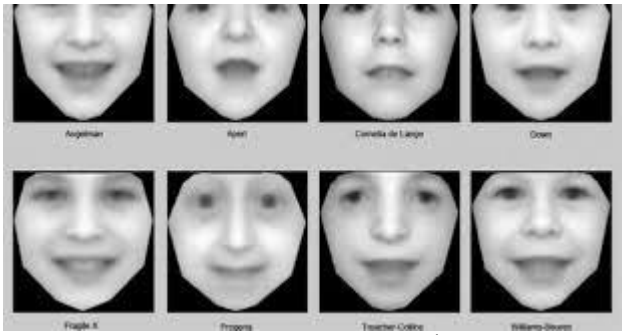

Figure 2 Divers sourires[1]

Vous comprenez bien sûr que, que ce soit pour fins de pointage ou que ce soit pour fins adaptatives de l'expérience de jeu, réagir à temps est chose essentielle. Heureusement, vous avez eu du succès dans votre cours de `STR`, et vous avez même (vaguement) mémoire de quelques petits trucs discutés ici et là dans ce cours. Vous espérez bien vous en tirer…

Le FaceSync® se veut un appareil grand public, et doit conséquemment être offert à un prix qui en fasse un objet accessible, limitant la sophistication des composants. Ainsi, un FaceSync® comprendra plusieurs capteurs de mouvement capables, individuellement, de rapporter au plus 30 positions par seconde (taux régulier, pas constant). Une position sera représentée par un 5-uplet $\{id, t, x, y, z\}$ dont chaque élément est un entier signé encodé sur 32 bits. L'élément id doit représenter de manière unique au système le capteur ayant émis le 5-uplet. L'élément t est un compteur dont la valeur dépend du capteur et qui incrémente de manière monotone avec chaque envoi. Ces capteurs communiquent avec **FaceScan®**, un système embarqué muni de `QNX v.4.1`, donc un `SETR` supportant l'héritage de priorités. FaceScan® comprend : un processeur 32 bits muni de deux cœurs dont l'horloge est cadencée à 500 MHz, 1 Mo de mémoire vive, et un disque rigide de 32 Mo. Selon vos tests, le WiFi de FaceScan® supporte un débit 150 $Mbps$. C'est sur FaceScan® que portera votre contribution à FaceSync®.

[1] http://images.atelier.net/sites/default/files/imagecache/scale_crop_587_310/articles/430151/atelier-diagnostic.jpg

Q00.0 – Le logiciel dont vous serez responsable doit être déployé sur FaceScan®. Il aura pour responsabilité de consommer les données des capteurs et de transférer à la console de Karaoké un portrait cohérent du visage capté à un moment. La garantie à offrir est de transférer les portraits cohérents en question à un rythme régulier de 30 par seconde.

On vous informe que l'une des technologies possibles pour les capteurs est événementielle et unidirectionnelle, au sens où le capteur émettra un nouveau 5-uplet à chaque fois qu'il détectera un mouvement, et ce jusqu'à concurrence du maximum par seconde mentionné plus haut. **Pour 6 points**, décrivez comment votre logiciel construira ce qui sera envoyé à la console de Karaoké dans ce contexte, et prenez soin d'expliquer les forces (**2 points**) et les faiblesses (**2 points**) de votre approche : _____

Q00.1 – Une autre des technologies possibles pour les capteurs procède par sondage. Dans ce cas, un capteur saisit chaque mouvement et génère un 5-uplet, jusqu'à concurrence du maximum par seconde mentionné plus haut, sans toutefois l'émettre en direction de votre logiciel. Par contre, chaque capteur offre un service permettant d'en obtenir sur demande le 5-uplet le plus récemment généré. **Pour 6 points**, décrivez comment votre logiciel construira ce qui sera envoyé à la console de Karaoké dans ce contexte, et prenez soin d'expliquer les forces (**2 points**) et les faiblesses (**2 points**) de votre approche : _____

Q00.2 – Un de vos collègues dit avoir lu que l'ordonnanceur de QNX est inéquitable et privilégie systématiquement les tâches de plus haute priorité parmi celles qui sont prêtes à s'exécuter (état *Ready*). L'architecture sur la table tient pour prioritaire la consommation des données provenant des capteurs et fait en sorte que ces données soient placées dans des tampons synchronisés pour être consommées ultérieurement par des tâches de priorité élevée, mais inférieure. Votre collègue se dit préoccupé que le consommateur, en synchronisant la lecture sur un tampon, soit préempté par une tâche plus importante tout en maintenant la possession du verrou sur le tampon, provoquant un interblocage. **Pour 4 points**, ses craintes sont-elles justifiées? **Pour 6 points**, expliquez votre réponse.

Q01 Expertise et expérience (_____/35)

L'un des informaticiens collaborant avec vous, Bertrand, vient à votre rencontre, perturbé. Il s'approche de vous d'un air dépité pour profiter de votre épaule accueillante.

Bertrand vous raconte qu'il vient de vivre un échange un peu rude avec un informaticien d'expérience. Leur échange avait trait aux exemples de code ci-dessous, à savoir lequel est préférable dans le contexte de votre système :

```
struct uplet {
   int32_t val[5] = {}; // zéro par défaut
};
uplet consommer(int capteur);
void utiliser(uplet);
void lire_capteurs(int ncap) {
   uplet* p = new uplet[ncap];
   for(int i = 0; i != ncap; ++i)
      p[i] = consommer(i);
   for(int i = 0; i != ncap; ++i)
      utiliser(p[i]);
   delete [] p;
}
```

```
struct uplet {
   int32_t val[5] = {}; // zéro par défaut
};
uplet consommer(int capteur);
void utiliser(uplet);
vector<uplet> consommer_tout(int ncap) {
   vector<uplet> v;
   v.reserve(ncap);
   for(int i = 0; i != ncap; ++i)
      v.emplace_back(consommer(i));
   return v;
}
void lire_capteurs(int ncap)
{
   for(const auto &val : consommer_tout(ncap))
      utiliser(val);
}
```

Dans ces exemples, la fonction `consommer()` est appelée pour le capteur `capteur` et retourner une copie du 5-uplet représentant la plus récente lecture sur ce capteur. La fonction `utiliser()` joue le même rôle à gauche comme à droite, soit utiliser un 5-uplet.

Bertrand a proposé la version de droite, qui est courte et pour laquelle il s'est inspiré d'un extrait lu dans un magazine, alors que le programmeur d'expérience est d'avis que la version de gauche est préférable.

`Q01.0` – **Pour `10` points**, à quelles conditions donnerez-vous raison à Bertrand (à droite), et à quelles conditions donnerez-vous raison à son illustre collègue (à gauche)? Votre réponse doit tenir compte des caractéristiques d'un `STR` (exécution prévisible, résilience, brièveté d'exécution, basse latence lors d'un appel, etc.).

`Q01.1` – Bertrand vous raconte que chez K.E.G., plusieurs craignent le recours aux exceptions dans le code, tous langages confondus. **Pour 5 points**, dans le comparatif de code proposé au début de `Q01`, quel sera l'impact sur le code de gauche d'une levée d'exception par la fonction `utiliser()`? **Pour 5 points**, dans le comparatif de code proposé ci-dessus, quel sera l'impact sur le code de droite d'une levée d'exception par la fonction `utiliser()`? **Pour 5 points**, dans le comparatif de code proposé ci-dessus, si la fonction `utiliser()` retourne un code de succès plutôt qu'une levée d'exception dans le cas d'un problème, expliquez l'impact sur les exemples de gauche et de droite : _____

Q01.2 – Bertrand se dit d'avis que la programmation de votre logiciel serait simplifiée si, plutôt que de retourner une copie du 5-uplet représentant la plus récente lecture sur un capteur, `comsommer()` bloquait jusqu'à ce que le capteur dont l'identifiant est passé en paramètre ait effectivement lu un nouveau 5-uplet. **Pour 10 points**, expliquez à Bertrand les conséquences d'un tel changement : _____

Q02	Questions noyautées	(_____/20)

Certains de vos collègues estiment que QNX, qu'ils connaissent peu, pourrait avantageusement être remplacé dans FaceScan® par Windows 10, très à la mode, ou encore par Ubuntu (vous suspectez d'ailleurs qu'ils aient alors espoir de pouvoir utiliser leurs outils de tous les jours dans les deux cas).

Q02.0 – **Pour 5 points**, expliquez à vos collègues en quoi le caractère équitable d'un ordonnanceur pertinent pour un système d'exploitation grand public peut ne pas être approprié pour FaceScan® : _____

Q02.1 – Un SETR repose typiquement sur un micronoyau minimaliste de très basse priorité. **Pour 5 points**, pourquoi en est-il ainsi? Et **pour 10 points**, donnez deux circonstances qui peuvent donner une opportunité au noyau de faire son travail malgré sa faible priorité : _____

Q03	**Questions techniques**	**(_____/15)**

Q03.0 – Un nouvel employé de K.E.G. propose d'utiliser Java dans FaceScan® et vous explique que les machines virtuelles Java contemporaines ont la qualité d'être offertes avec un profileur intégré qui améliore dynamiquement le comportement du programme. **Pour 5 points**, indiquez si sa suggestion convient à FaceScan® et expliquez pourquoi. _____

Q03.1 – Un groupe de vos collègues envisage utiliser un exécutif dans FaceScan®, mais certains se plaignent de ne pas trouver d'exemples d'exécutifs sur *Stack Overflow*. Connaissant votre pedigree, on vous consulte pour mieux comprendre ce dont il en retourne. **Pour 5 points**, considérant le niveau 0 d'un exécutif est sa bande critique, si une tâche A est dans une bande de niveau i d'un exécutif et une tâche B est dans une bande de niveau $i + 1$, que peut-on affirmer sur le nombre d'exécutions de A entre deux exécutions de B?

Q03.2 – **Pour 5 points**, expliquez comment il est possible d'utiliser un conteneur standard tel qu'un `vector` tout en conservant un comportement déterministe lors de l'ajout d'éléments, donc sans avoir recours à de l'allocation dynamique de mémoire. _____

Q04	**Quelques petits trucs rapides**	**(_____/10)**

Q04.0 – **Pour 2 points**, donnez un désavantage de l'acte de déplacer des calculs du moment de l'exécution d'un processus vers celui de sa compilation : _____

_____

_____

_____

_____

Q04.1 – **Pour 2 points**, pourquoi un accès concurrent à deux indices distincts d'un même tableau peut-il entraîner un comportement erratique à l'exécution d'un programme? _____

_____

_____

_____

_____

Q04.2 – **Pour 2 points**, donnez une circonstance où on préférera une assertion dynamique à une assertion statique : _____

_____

_____

_____

_____

Q04.3 – **Pour 2 points**, outre le gain en termes de vitesse d'exécution, donnez un autre avantage de représenter une unité de mesure par un type spécifique (p.ex. : `Metre`) plutôt que par un primitif :_____

_____

_____

_____

_____

Q04.4 – **Pour 2 points**, pourquoi une implémentation morcelable d'un algorithme est-elle susceptible de prendre plus de temps à mener sa tâche à terme que son équivalent s'exécutant d'un seul tenant (sans morcellement)? _____

_____

_____

_____

_____

Annexe 00 – Aide-mémoire

Les unités de mesure utilisées dans les questions de cet examen vont comme suit. Supposez huit bits par *byte* (donc un *byte* équivaut à un octet), même si cela ne s'avère pas toujours en pratique.

Unité	Sens
Bit	Unité élémentaire, valant `0` ou `1`
Bps	Bits par seconde
Hz	Hertz. Mesure de fréquence ($n\ Hz$ signifie n fois par seconde)
$MBps$	Mégabits par seconde. Mesure de débit ($n\ MBps$ signifie $n \times 1024^2$ bits par seconde)
MHz	Mégahertz. Mesure de fréquence ($n\ MHz$ signifie $n \times 1000^2$ fois par seconde)
Mo	Mégaoctet. Mesure de capacité ($n\ Mo$ signifie $n \times 1024^2$ octets)
Octet	Bloc de huit bits (souvent synonyme de *byte*, bien que ce ne soit pas formellement exact)

J'entends par rythme régulier une forme telle que « n fois par seconde ». J'entends par rythme constant une forme telle que « à chaque $\frac{1}{n}$ seconde ».