# **INFO6015– Animation 1 - Mid-term Exam – Fall 2014**

December 12th, 2014

Instructor: Michael Feeney

## The exam format:

* You may use any resources you feel are necessary to complete the exam, but you are to answer the questions **on your own**. I will be looking for plagiarism (i.e. copying) very carefully. There is *no possible way* that the specific code to answer these questions, or the output to the screen, would be very similar to the look of another student’s code. Remember, this is a test and there are very clear policies about cheating on tests.   
  + <http://www.fanshawec.ca/admissions/registrars-office/policies/cheating-policy>
  + <http://www.fanshawec.ca/sites/default/files/assets/Ombuds/cheating_flowchart.pdf>
* The questions are ***NOT*** of equal weight.
* There are three (3) pages with seven (7) questions
* The answers may be one or a combination of the following:
  + Short answer (in your own words)
  + Snippets of code
  + Complete running solutions
* CLEARLY indicate which answer goes to which question. My suggestion is that you place each answer in its own folder, named “Question\_01”, “Question\_02” and so on (or something equally clear). Another option is to create a Visual Studio solution and add a number of projects – one per question – to it. If I can’t make heads or tails of what question is what, I probably won’t even mark it.
* Place any written answers into a Word, RTF, or text file. Again, *clearly* indicate which question you are answering.
* If you are combining answers (which is likely), please indicate this with a “readme” file or some note (*not* buried in the source code somewhere).
* For applications: if it doesn’t build and run, *it’s like you didn’t answer it*. I’ll correct trivial, obvious problems (like you clearly missed a semicolon, etc.), but you need to be sure that it compiles and/or runs.
* You have until **11:59 PM** on **Wednesday, December 12th** to submit all your files to the appropriate drop box on Fanshawe Online.   
    
  **NOTE:** Although this may “look and feel” like a project, it isn’t, it’s an **exam**, so there is **no concept of “late marks**”; if you don’t submit your files by 11:59 PM, you don’t get any marks at all. *Don’t Be Late submitting.*

(Also be **SURE** that you are actually submitting the correct files)

* You can reach me through e-mail ([mfeeney@fanshawec.ca](mailto:mfeeney@fanshawec.ca)) or by calling the school.

## Questions:

Unless otherwise stated, assume *you have to create an application that demonstrates the following.* In other words, assume each question starts with “Create an application that...”

1. (10 marks) You are to create a “Batman simulator”. To start, create the following interface “Badguy” class (using the understanding of what an “interface” is in the C++ language):

|  |  |  |
| --- | --- | --- |
| **Name** | **Parameters** | **Returns** |
| TakesPunch | “Damage” which is a float | Nothing (void) |
| GetHealth | None (void) | Float |
| CauseMischief | “Time”, which is a float | Boolean |

1. (10 marks) Create a “Batman” class. This class needs to have the following methods:
   * SpotBadGuy, which takes a pointer to a Badguy class (from Q1)
   * DefendGotham, which takes nothing and returns nothing
   * Has a container of some sort (vector, may, whatever) to keep track of any Badguys that have been passed to “SpotBadGuy”
2. (30 marks) Create an abstract factory class which can create the following bad guys. They have a location (2D point), health, and a unique identifier (a number of string). To show that an event has happened, print something to the screen (for instance, on TakesPunch, print out how much health they have lost, if any):  
   * The Joker:
     + Health starts at 125
     + TakesPunch reduces health by a random amount between 0 and 25
       1. 1 in 5 chance of changing location
     + Mischief:
       1. Kill someone one by a random 1 in 3 chance
       2. Steals $1,000 to $250,000, by a 1 in 2 chance
   * The Penguin:
     + Health starts at 90
     + TakesPunch reduces health by a random amount between 0 and 35
       1. 1 in 8 chance of changing location
     + Mischief:
       1. Kills between 1 and 10 people, by a 1 in 3 chance
       2. Steals $10,000 to $1,000,000 every time
   * Cat Woman:
     + Health starts at 150
     + TakesPunch reduces health by 10
       1. 1 in 2 chance of changing location
     + Mischief:
       1. Steals $5,000 to $150,000, by a 1 in 4 chance
3. (30 marks) Create a “World” class that does the following:
   * Keeps track of the downtown of Gotham City, which is a 10 km by 10 km area
   * Creates Batman and places him somewhere in the city
   * Creates 10 random bad guys and places them somewhere in the city (yes, I know that this means there will be more than one “Joker”, or whatever – you’ll just have to get over it – let’s say they are clones or “copy cat” criminals, or something... Use the unique identifier to differentiate the different instance of the same class)
4. (50 marks) Create a “game loop” that does the following on each cycle (“frame” or “update”). In other words, this is called over and over again:
   * The world class determines the closest bad guy to batman
   * Informs Batman of this bad guy’s location (**SpotBadGuy**)
   * Batman moves to that location
   * Batman punches the bad guy (only once punch per cycle) (world calls **DefendGotham**, then Batman calls **TakesPunch** on selected bad guy)
   * If a bad guy changes location, the *world* has to pick the next closest bad guy
     + This can be done by Batman informing the world, somehow, or having the world “notice” that the bad guy isn’t at Batman’s location
   * **CauseMischief** is called on all bad guys. If the bad guy \*isn’t\* being punched, will execute it’s normal call, but if it was just punched, it won’t (because Batman is punching them...)
   * Batman sticks around until the bad guys health is gone
   * This cycle continues until all the bad guys are gone (have no health). When this happens, you need to print out “I’m Batman” to the screen.
5. (50 marks) Update the application so that Batman keeps track of how many people are killed and how much money is stolen by updating the bad guy interface and/or the Batman class in some manner.   
     
   HOWEVER, you:
   * cannot pass \*all\* the bad guys to bat man in order to do this. In other words, Bat man can only be aware of the bad guy his is currently pursuing. If the bad guy “gets away” (i.e. changes location during a punch), he *can* keep a pointer to this bad guy if needed
   * can use the World class as an intermediary to pass this information to Batman (as the world class “knows” all about all the bad guys)

When there are no bad guys left, print out a summary of:

* The total money each took (individually)
* The total number of people each one killed (individually)
* The grand total of money and killed people (all bad guys in total)
* This summary has to be printed out \*before\* you print “I’m Batman”

1. (10 marks) Update the application so a random number of bad guys (from 5 to 150) is created each time (i.e. there isn’t just 10 bad guys).

(That’s it for the exam).