**INFO6044 – Game Engine Frameworks & Patterns**

**Midterm Exam – Thursday, October 25th, 2023**

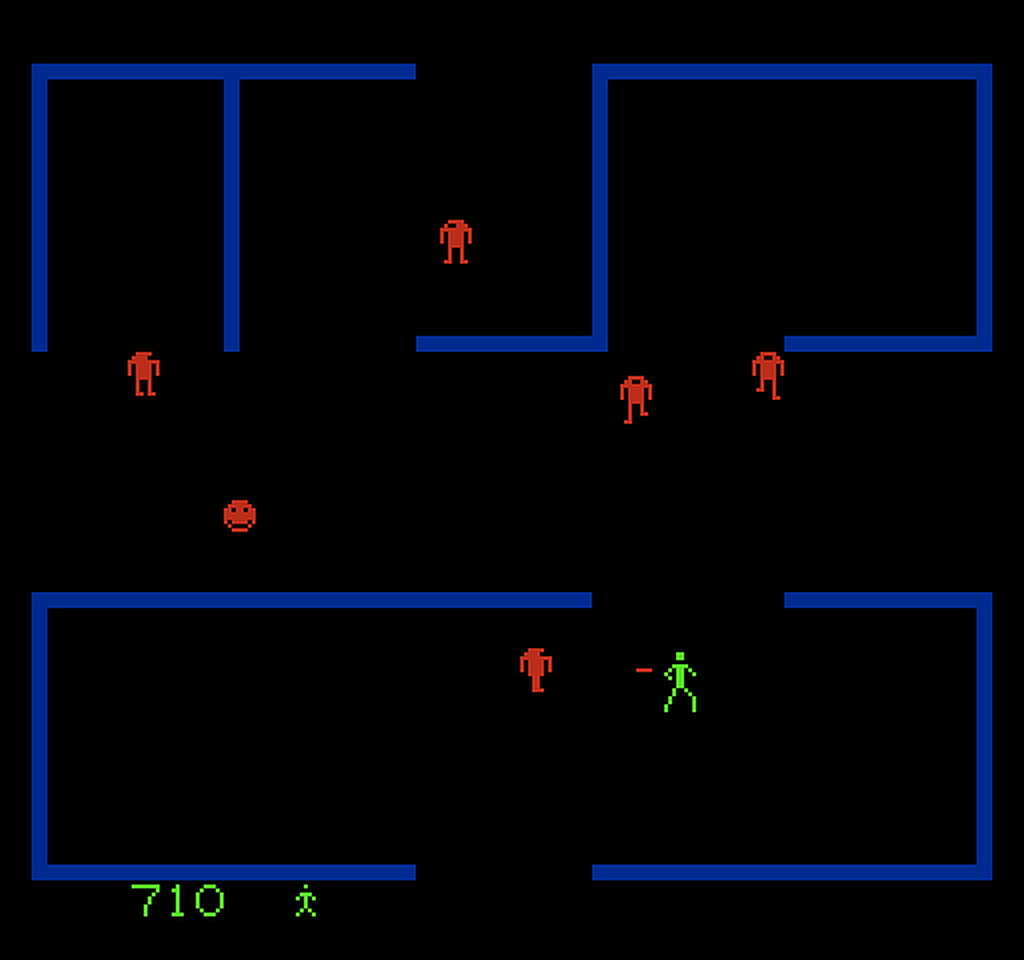
Instructor: Michael Feeney

## 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>
* It is an “open book” exam. You have access to anything you book or internet resource you’d like
* The questions are ***NOT*** of equal weight. The exam has **six (6)** questions and **six (6)** pages.
* Your solution **must have a graphical output** (or graphical + console based if that’s helpful).  
  + Note: You are not being marked for the “beauty” of your graphical output.   
    I’m only interested in being able to see what’s going on. I’d suggest a single large point or directional light so that you can clearly see everything.  
    A camera that can move around the scene might be helpful, but is not required – so long as I can see the entire maze *and* see what’s going on.
* If the questions are separate **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.
* The exam questions “build” on each other, somewhat, so the questions can be combined. If you are combining answers, please indicate this with a “readme” file or some note (*not* buried in the source code somewhere).  
  i.e. which questions you are attempted and which ones you didn’t.
* Do ***NOT*** do some clever “*oh, you just have to comment/uncomment this block of code*” nonsense. However, if the questions ***CLEARLY AND OBVIOUSLY*** build on each other, you may combine them (like if one question places objects, then the next one moves objects around with the keys) – even so, **MAKE IT 100% CLEAR** to me what questions the solution is attempting to answer.
* **If the solution does not build (and run), I will not mark it** (so you will receive zero on questions that can't be built and/or won't 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.
* Your solution may **not** contain any third party “core C++” libraries (like boost). I will not have boost installed, and will not install it; as a result, if you using boost, your solution will *not* build (so a mark of zero).
* You many have other “utility” libraries, like ones to load textures, models, sounds, etc. However, make sure your submission is complete so that I can build your solution.
* When ready to submit, please delete all the “extra” Visual Studio files before zipping it up (remember this is C++, so all I really need is the .h and .cpp files, right?), like the “Debug” and “Release” folders with the “obj” files, as well as the intellisense file (in VS2017, that’s the “.vs” folder).
* solutions must be using a x64 Release library C++ project using the **default** settings of Visual Studio 2022 (which uses “Default (ISO C++14 Standard)”) using the OpenGL 4.x API (with glfw, glad, and glm).

|  |
| --- |
| You have until **11:59 PM** on **Thursday, October 26th** 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 the time the drop box closes, you don’t get any marks at all.  *Please don’t be late submitting.*  (Also be **SURE** that you are actually submitting the correct files) |

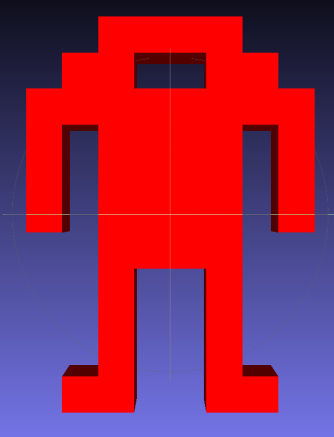
**“Berzerk, on the weekend”**

****



The Berzerk game has you playing as a human that has to quickly move through a number of simple maze like rooms either avoiding the guarding robots or shooting them (they are shooting at you, so fair is fair, I suppose). In the images above, you play as the snappy dressed guy with the bowl hair cut, who’s shown as the green stick man in the image on the left.

In this exam, we are not going to do that. Instead, we’re going to explore what the robots do when there *isn’t* a human to chase down and kill. What did you think they were doing, standing around waiting for a human?



In the “Sprite map.7z” file (which extracts to the “Sprite map” folder), you’ll find a set of 3D versions of the Berzerk robot models. The one to the right is “Robot\_Eye\_Centre.ply”, which is the only one you need, though there are a few others that you can use if you’d like (the only change is the location of the “eye”).

Based on this post: <https://www.cemetech.net/forum/viewtopic.php?t=9345&start=0> the INFO6044\_Midterm\_Helper program will generate a level based on your student number.

You pass your student number on the command line (either through the cmd prompt, or you can edit the “”RunTheFile.bat” batch file with your student number).

For example, if yours was 12345, you’d type: “INFO6044\_Midterm\_Helper.exe 12345”

|  |
| --- |
| You ***must*** submit the **text** *and* **ply** file output to me, to show you’ve used *your* student number.  If you **don’t** do this, I will ***not*** accept your submission.  (And it goes without saying that there’s no way your level is the same as anyone else’s...) |

As well as the screen output, this generates two files:

* A “**Berzerk\_Level\_XXX.txt**” file where “XXX” is your student number. This has the same text based arena maze it displayed (but flipped along the y axis to match the ply file generated).
* A **Berzerk\_Level\_Ouput.ply** file which you can use if you’d like. This is a 3D variation of the arena maze in the text file. Note that it *doesn’t* have normals generated.

Here’s an example arena maze text output (left) and the ply output (right). This is for SN 123456 (I think?).

(The red squares are explained in question 2)

|  |  |
| --- | --- |
| XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  X X  X X  X X  X X  X X  X X  X X  X X  X X  XXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX X X  X X X X X  X X X X X  X X X X X  X X X X X  X X X X X  X X X X X  X X X X X  X X X X X  X X X X X  X XXXXXXXXXXX X XXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXX X X  X X X X X  X X X X X  X X X X X  X X X X X  X X X X X  X X X  X X X  X X X  X X X  XXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX XXXXXXXXXXX  X X X X X  X X X X X  X X X X X  X X X X X  X X X X X  X X X X X X X  X X X X X X X  X X X X X X X  X X X X X X X  XXXXXXXXXXX X X XXXXXXXXXXX XXXXXXXXXXX X XXXXXXXXXXX  X X X X X X X  X X X X X X X  X X X X X X X  X X X X X X X  X X X X X X X  X X X X X X X  X X X X X X X  X X X X X X X  X X X X X X X  X XXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXX X X XXXXXXXXXXX  X X  X X  X X  X X  X X  X X  X X  X X  X X  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX |  |

1. (10 marks) Using the INFO6044\_Midterm\_Helper program, generate an arena text and ply file.

Note a few things about the arena maze:

* The arena is 4x the original Berzerk level (twice the width and height).
* The exits don’t line up with the walls... I realized that the original game had an odd number of walls, but when I doubled it, it became even – i.e. the exit couldn’t be in the “centre” of the wall.   
  (Oh well, we’re not using the exits anyway)

1. (50 points): There are ten (10) robotsthat will interact.   
     
   **At this point, the output should be using the 3D application.**

Referring to this post (https://www.cemetech.net/forum/viewtopic.php?t=9345&start=0) that talks about the “pillars” in the room, and the walls coming out from them – this is how the helper program generates the level – note that the pillar are 10 units/block apart. In other words, every four (4) pillars make a 10x10 unit “room” or “area”.

The red squares in on the images on the previous page (page 4) show an example of one of these “areas”.  
Imagine that the level was divided up into a grid of 10x10 squares if that helps.   
  
For the rest of the exam, we’ll refer to these 10x10 spaces as “areas” inside the room.   
  
Place your ten (10) robots spaced throughout the level, where there is only one robot in each “area”.

You can place these in code\* or you can “hard code” the placement.

(\*one possible way to do this is when you randomly place a robot, check to see if there’s at least 10 units up/down/left/right to any other robot – if there *is* a robot already there, then pick another random, and keep going until the location is OK)

1. (200 points) Help them find their best friend.

The 1st task they will do is find their best friend.

* Group the robots into pairs of “best friends” (10 robots, so 5 pairs)
* Assign a “Friendship Value” to this pairing. This is a random number between 5 and 10 and indicates the strength of the friendship (this will come into play in question 5).
* These robots will try to find each other.
* They will move randomly around (slowly enough to see what’s happening, so either move them very slowly or just move them every X frames or seconds).
* They can change direction in any way you’d like, but they can’t go through walls.   
  (You can see if they are within a certain distance from the walls, and *not* go that direction, or always stay a few units from any of the “cubes” that make up the wall, or whatever)
* When they are “right beside” their best friend (use your judgement, but they should be physically close) they should:
  + Stop moving
  + Change colour

Once they’ve all found their best friends, they can go to task 4 (question 4).

1. (200 points) Now they need to play “Euchre” or “Exploding Kittens”.

Euchre requires four (4) players.

Exploding Kittens requires six (6) players.

* Choose two robots that *aren’t* friends to have the cards (like one robot has cards for Euchre and one for Exploding Kittens).
* Using any shape/3D model you’d like (a cube, sphere, bunny, whatever), indicate these two robots by placing this shape above them.
* Now repeat what happened in question 3 but this time, three (3) of the robots would like to play Euchre and five (5) others would like to play Exploding Kittens.   
  Move these robots to the ones that have the appropriate game.
* Once they get there, have them stop moving and change their colour again (to indicate they have found the other players).

1. (200 points) Now it’s a birthday party.

Fun fact: All the robots were made on the same day, so they all have the same birthday.

AND it’s their birthday today!!! What a coincidence.

The robots have decided to do a gift exchange with one other robot (so they don’t have to get gifts for every other robot, which would be expensive and they don’t get paid much shooting humans).

* Each robot picks a random other robot that *ISN’T* their best friend.   
  Every robot should have one and only one robot giving them a gift.
* They find that robot. This can be the way they did it in questions 3 & 4 or  
  (**BONUS 15%:** They move directly towards the other robot, like in the game)
* Once they are “close enough”, they “give” their gift to the other robot.
* The robot that receives the gift decides how much they like it, on a scale of 5 to 10 (i.e. pick a random number between 5 and 10). If this number is *MORE* than their current “Friendship Value” then they have a NEW BEST FRIEND!
* If they have given their gift, the go back to finding their best friend again.   
  + i.e. run the “finding” part of question 3, but this time, there might not be pairs of robots.
* This might be chaotic: either the robots will “clump up” (in pairs or more) and they might even all congregate together in one big group.
* Like in question 3, once they find their best friend, they can stop moving.

1. (BONUS: 15%) Have them get bored with this, decide they need some “alone time”:

* They move away from any other robot, like at the start when each robot was in its own “area”.
* Once they’ve been there, alone for a while, they decide that they miss their best friend.
* ...but they are fickle, so they randomly pick a new best friend (which could be their original one, but likely not, since there are 10 robots – i.e. run the “pick a random best friend” process again.
* Repeat from question 3 onward... forever!  
    
  (or until a pesky human comes in a ruins everything)

|  |
| --- |
| **That’s it** |