

# Game Development - Revaluation Examination - January 31th 2019

YOUR FULL NAME: **Solution Provided by the teacher**

- You have 2 hours to complete the assignment.
- Only valid text will be the one inside each box, everything else will be ignored by the teacher

1. (2 points) Find the parsing errors and suggest any chances of improving this XML:

```
<entities>
  <static>
    <pylon hp=100>
      <instance1 coords=50,25/>
    </pylon>
    <base hp=500>
      <instance1 coords=40,50/>
    </base>
  </static>
  <dynamic>
    <archon hp=200 damage=25 distance=1 speed=3/>
      <instance1 coords=80,50/>
      <instance2 coords=70,80/>
    </archon>
    <dragoon hp=75 damage=10 distance=15 speed=4/>
      <instance1 coords=75,70/>
      <instance2 coords=76,80/>
    </dragoon>
  </dynamic>
```

Better that all XML should start with a header like

```
<?xml version="1.0" encoding="utf-8"?> koo
```

Entities node is not closed

Archons and dragoons are closed incorrectly with “/” in the end of their opening node

It is strongly recommended to surround attribute values with quotes, in some cases like using spaces or other characters in the values, it will raise an error

Instead of using instance1 / instance2 we should just “instance” since we can freely repeat node, it makes it easier to navigate and scale.

2. (2 points) Describe your approach to **draw** an isometric map. Use pseudocode to describe its core loop. Be specific about the math formula needed to find final screen coordinates.

We will have to take in account the following elements:

- Sprite sheet used
- Visible layers to draw, from back to front
- Size of the map (maximum rows and columns)

for each layer

```
{  
  for each row in layer  
  {  
    for each column in layer  
    {  
      find the sprite id for row,column on layer  
      find the sprite_coordinates in the sprite  
sheet  
  
      screen_x = (row-column) * (tile_width / 2)  
      screen_y = (row+column) * (tile_height / 2)  
  
      draw sprite_coordinates at screen_x, screen_y  
    }  
  }  
}
```

3. (3 points) Define and explain the pros and cons of **Fixed Time Step** versus **Variable Time Step**. Why would it be interesting to limit the framerate in a video game ?

**Fixed Time Step** assumes that every frame takes the same amount of time. This is only predictable if we know exactly our target hardware (console):

**Pros:** simpler calculations, more predictable outcome

**Cons:** cannot deal with unexpected variation of frame time that would produce a bad user experience

**Variable Time Step** calculates every frame the time it takes and uses it as a factor for most movement and animation calculations.

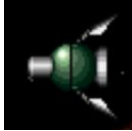
**Pros:** smooth experience even with frame rate changes

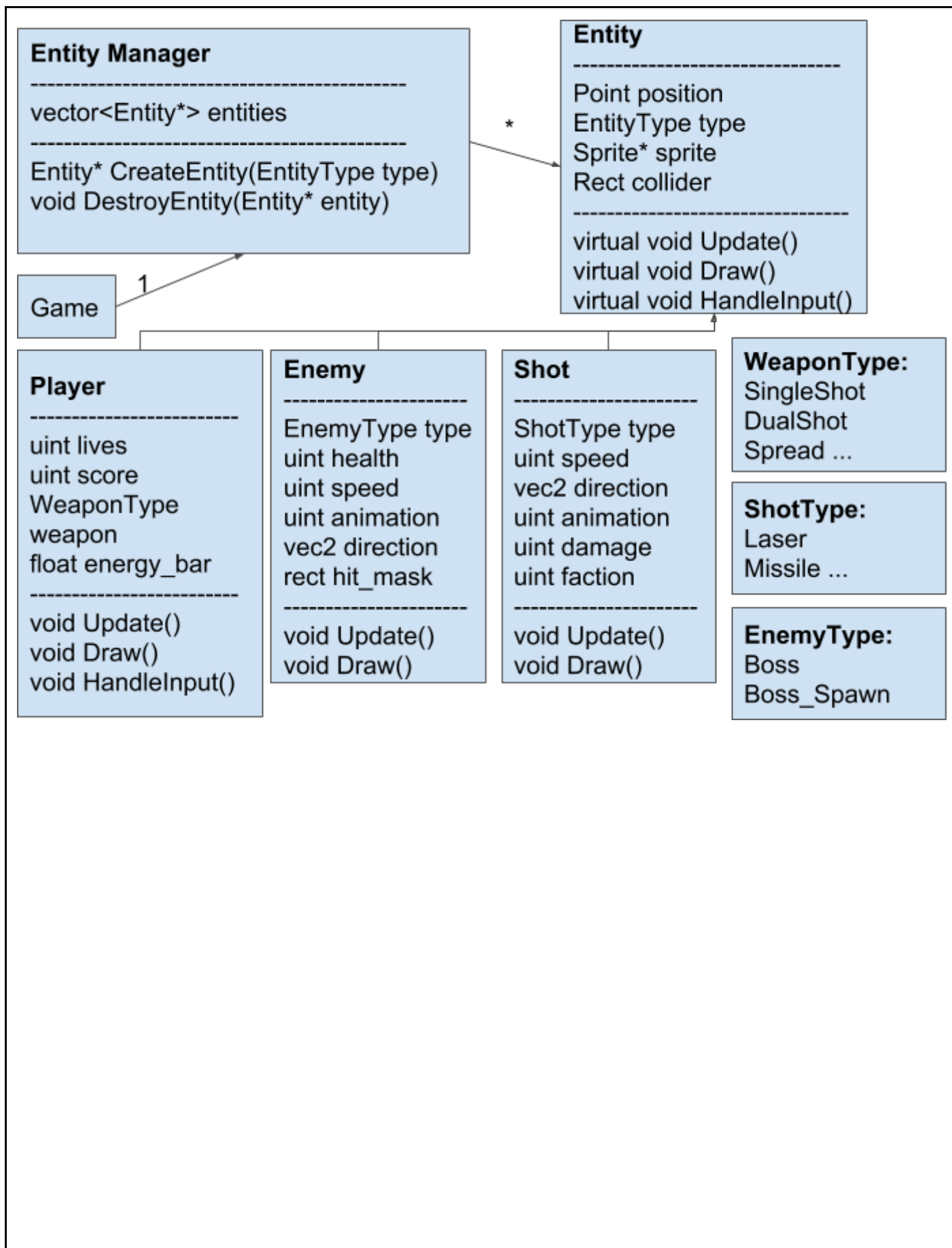
**Cons:** non-deterministic, weird behaviour in very small/large steps

Limiting frame rate allows us to hide peaks of CPU usage, code becomes more predictable and the general experience is smoother for the user.

4. (3 points) Given can be seen in those two screenshot from *Final Fight*, draw the UML for an entity system that would have all those elements we can see in the picture. Take in account that:

- Two player active, yellow and red
- Notice each player can have different weapon setup and energy bar charged.
- The boss shoot and spawn smaller enemies:





*Use this page for your own notes. You cannot use any other page. Do not remove this page from the set.*