

## Game Development TRS - Final Examination - January 12th 2018

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. (3 points) Describe the Dijkstra algorithm, its steps in pseudocode/python/C and create an example where it can be useful to use over A\* in a video game.

Dijkstra builds on top of BFS but add costs to each node, generating a non-uniform expansion over the navigation mesh:

```
--
frontier = PriorityQueue()
frontier.put(start, 0)
came_from = {}
cost_so_far = {}
came_from[start] = None
cost_so_far[start] = 0

while not frontier.empty():
    current = frontier.get()

    for next in graph.neighbors(current):
        new_cost = cost_so_far[current] + graph.cost(current, next)
        if next not in cost_so_far or new_cost < cost_so_far[next]:
            cost_so_far[next] = new_cost
            frontier.put(next, new_cost)
            came_from[next] = current
--
```

An example of practical use would require a very common node that we will pathfind to frequently and does not move: the main base of the player for example. Since Dijkstra will calculate the path from any other node to the base, if we can afford the memory, the creation of paths to that point would be all precalculated.

2. (2 points) Find out the mistakes in this XML and suggest improvements.

```
<entities>
  <static>
    <bushes>
      <instance coords="50,25"></instance>
      <instance coords="51,25"></instance>
      <instance coords="52,25"></instance>
    </bushes>
  <dynamic>
    <player coords="47,27" facing=east total_hp=4 hp=1
green_gems=216 arrows=10 bombs=0/>
    <chickens hp="1">
      <instance coords="80,50" facing="west" flying="true"/>
      <instance coords="80,50" facing="west" flying="true"/>
      <instance coords="82,50" facing="east" flying="false"
color="yellow"/>
    </chickens>
  </dynamic>
</entities>
```

1. Lack of header

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

2. Each instance can close itself without another element

a. `<instance coords="50,25"/>`

3. Both `<static>` and `<dynamic>` are not closing up

a. Add `</static>` and `</dynamic>` at the right place

4. Player attributes do not use “” (*important: this is not required but advised*)

a. `<player coords="47,27" facing="east" total_hp="4"
hp="1" green_gems="216" arrows="10" bombs="0"/>`

3. **(2 points)** Explain the concept of Profiling in the context of video game programming. Why we need to do it, who should do it and what kind of areas of the code we apply it.

Profiling means measuring how long takes to execute different parts of the code. It is used for optimization and required since the code complexity is way beyond what any developer can grasp.

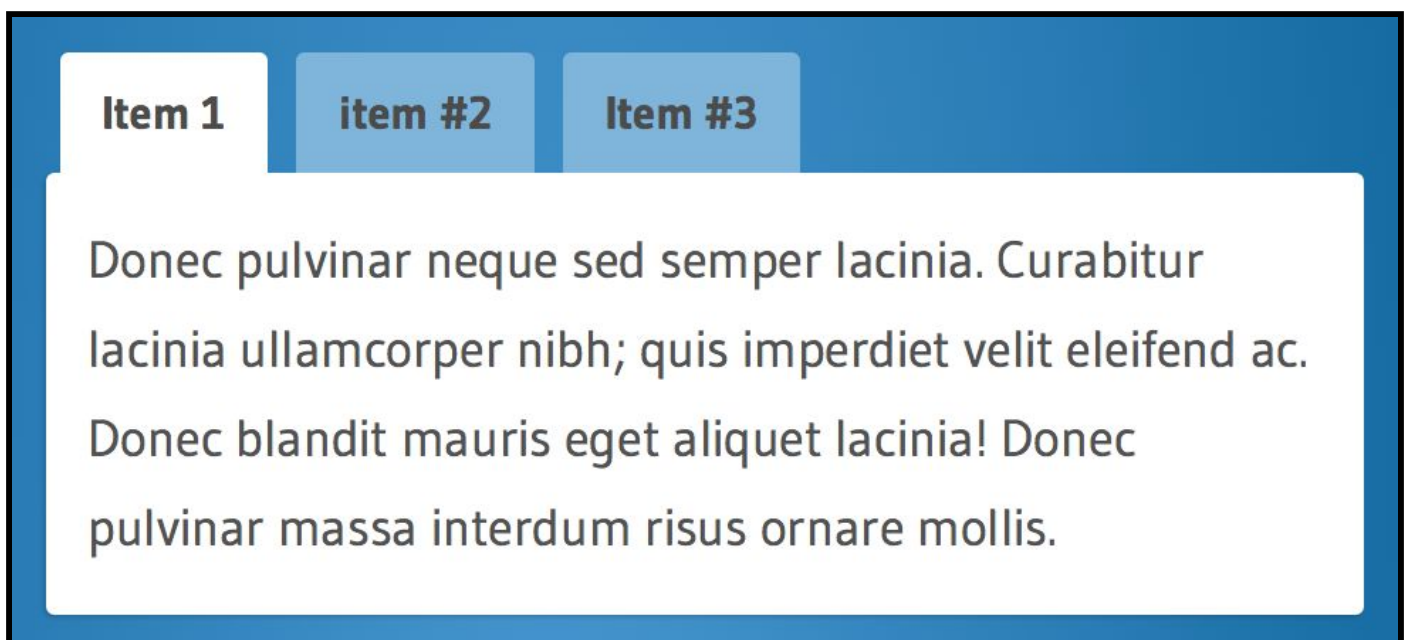
On top of that, the variability of the development process makes it unpredictable to understand the ramifications of each decision taken to improve the game experience.

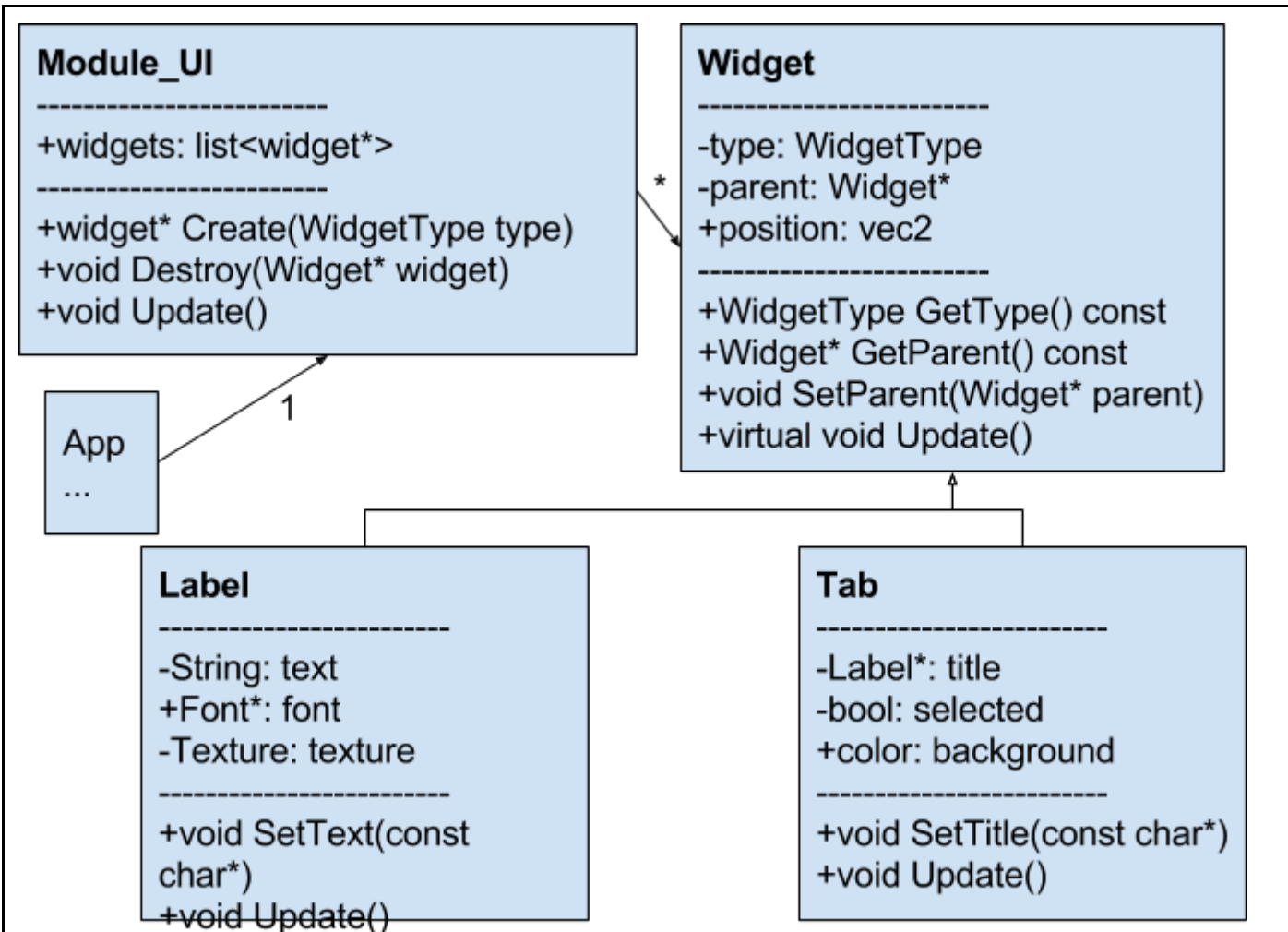
On top of that, we want to be able to deliver a smooth experience in the lowest possible hardware to attract more players.

Pretty much all developers, including artist and designers, need to be aware of the profiling tools available and use them to understand the consequences on the performance from the content they want to put into the game.

A Profiler mainly concerns about the cpu time spend in the code, but there are other factors like memory usage, gpu load and thread overhead.

4. **(3 points)** Create an UML structure for an UI system that would support the elements (and only those elements) in this picture. Aim for simplicity and modularity. Write the code in C that would create those items before the scene starts.





The code would be:

```
Widget* tab1 = App->Module_UI->Create(TAB);
Widget* tab2 = App->Module_UI->Create(TAB);
Widget* tab3 = App->Module_UI->Create(TAB);
```

```
tab1->SetTitle("Item 1");
tab2->SetTitle("Item #2");
tab3->SetTitle("Item #3");
```

```
Widget* text = App->Module_UI->Create(LABEL);
text->SetText("Donec ...");
text->SetParent(tab1);
```

```
// We assume that default color for tabs is white and is none is selected the
// first one will be shown
```

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