

Computer Games Development

TDD (Technical Design Document)

Year IV

**Mantas Zalnierius**

**C00241626**

**28/04/2022**

**Declaration**

This examination/assessment will be submitted using Turnitin as the online submission tool. By submitting my examination/assessment to Turnitin, I am declaring that this examination/assessment is my own work. I understand that I may be required to orally defend any of my answers, to the lecturer,at a given time after the examination/assessment has been completed, as outlined in the student regulations.

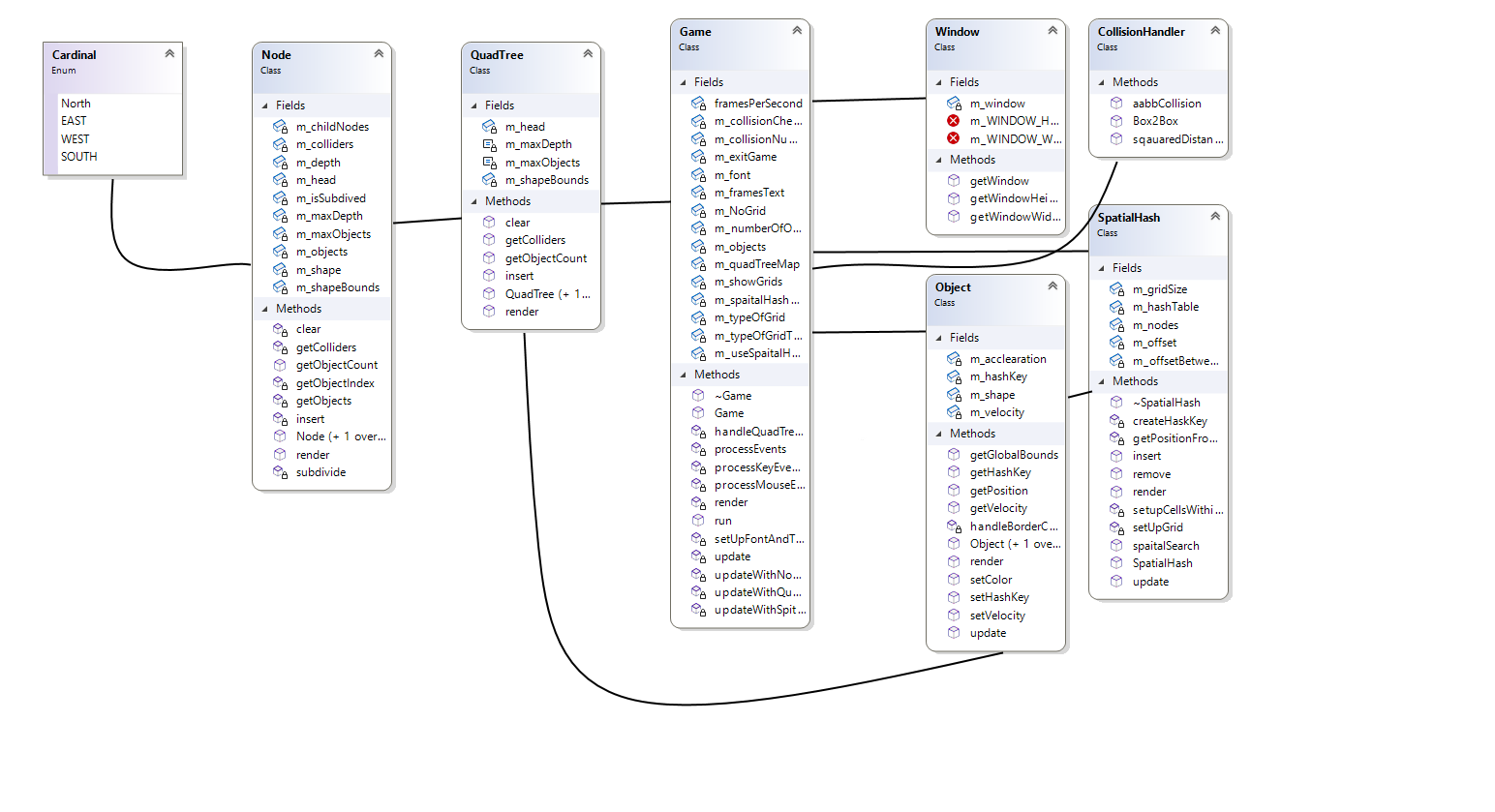
**Contents**

Software Architecture 4

Features 5-8

CRC Cards 9-14

**Software Architecture**



**Features**

The following will be my features and the steps needed to create those features.

**Basic collision detection (N^2) between shapes:**

* Create a base “Object” class that will hold all the data for the primitive shapes we need for the project. This Object class will contain data such as position, velocity, and shape. In this case we will use rectangles.
* Create mouse button events to create objects within the scene.
* Have the objects move in a random direction with a random velocity.
* Have objects collide off the bounds of the window.
* Have collision detection between objects.

**Implement a Node class to be used with a Quad Tree:**

* Create a Node class that stores variables that hold data for all the children of that node, the current head node, the current depth of a node, the max depth a node can reach, the max objects allowed to in the game, the shape bounds of the node and the if the node is currently subdivided.

**Implement Quad-Tree data structure:**

* Create a Quad-Tree class that holds that for the max depth it can reach, the max number of objects it can hold and the current head node.
* Implement a render function that will draw all the nodes within a quad-tree. Sdf
* Implement an insert function that will be passed a vector of nodes and will insert all these nodes into the head node. Implement another insert function into the head node, that will insert nodes either into child nodes or the head node.
* Implement a subdivide function, that will nodes if they need to. Also, I will implement a render function that will render every single node and child nodes.
* Implement a getColliders function that will get all the colliders that are necessary to handle collision.
* Implement a clear function, that will clear all nodes.

**Implement a Spatial Hashing grid:**

* Create a spaital Hashing grid that will hold, the grid size, the hash table that will contain the key value pairs of Objects within the scene, the offset between cells.
* Implement a createHashKey function that will create and return a hash key value from the position of the object. Also create a getHashKey function that will return the hash key of a cell.
* Create an addObject function that will create a hash key from the position of the object and store into the Object. Modify the Object class to hold a hashKey variable and a function called setHashKey that will set that objects hashKey and function called getHashKey that will get the hash key. I will also implement a removeObject function that will remove an object from the hash map using its hash key value.
* Implement an update function that will update each Object. This function will update the hash keys of each Object.
* Implement a spaitalSearch function that will take an Object and a search value. This function will use this search value to search for close by Objects and return Objects that are in the search area.

**Implement UI for the project and switching of grids:**

* Modify the Game class, to have 4 text and 1 font variable. The text will contain information about the current frame rate of the game, the number collisions occurring per frame, the number of Objects within the scene and the current type of grid active in the scene.
* Implement key events to allow for switching of grids.
* Implement updates for the number of collisions for no grid, spatial hashing, and quad-tree.

**CRC Cards**

The following will by my CRC cards for all my classes, their responsibilities, and collaborators.

|  |  |
| --- | --- |
| **Class Name: Game** |  |
| **Responsibilities** | **Collaborators** |
| Runs the scene | None |
| Updates the scene | Objects,  SpatialHasing,  QuadTree,  Window |
| Renders the scene | Objects,  SpatialHasing,  QuadTree,  Window |

|  |  |
| --- | --- |
| **Class Name: Window** |  |
| **Responsibilities** | **Collaborators** |
| Renders Objects | Sf::RenderWindow |

|  |  |
| --- | --- |
| **Class Name: Object** |  |
| **Responsibilities** | **Collaborators** |
| Allows creation of Rectangles. | Sf::Rectangle |
| Allows of setting/getting a hashKey | Sf::Vecter2U |
| Allows for rendering of Rectangles | Window |

|  |  |
| --- | --- |
| **Class Name: Spatial Hashing** |  |
| **Responsibilities** | **Collaborators** |
| Creating a 2D grid | Sf::Rectangle |
| Creating a hash map corresponding to Objects Position | Object |
| Updating hash values of Objects | Object |
| Searching an Objects area for nearby Objects | Object |
| Creating and sending hash keys to Objects | Object |
| Rendering of Grid | Window |

|  |  |
| --- | --- |
| **Class Name: Quad-Tree** |  |
| **Responsibilities** | **Collaborators** |
| Inserting nodes into the head node | Node |
| Clearing all nodes from the head node. | Node |
| Getting all the nearby colliders. | Node |
| Count all the objects within the head node. | Node |
| Rendering all nodes. | Node |

|  |  |
| --- | --- |
| **Class Name: Node** |  |
| **Responsibilities** | **Collaborators** |
| Updating all nodes by Subdividing them when they need to. | Object |
| Rendering all Nodes to the scene. | Window Object |
| Getting all the colliders that are near an object. | Object |
| Clearing all nodes. | None |