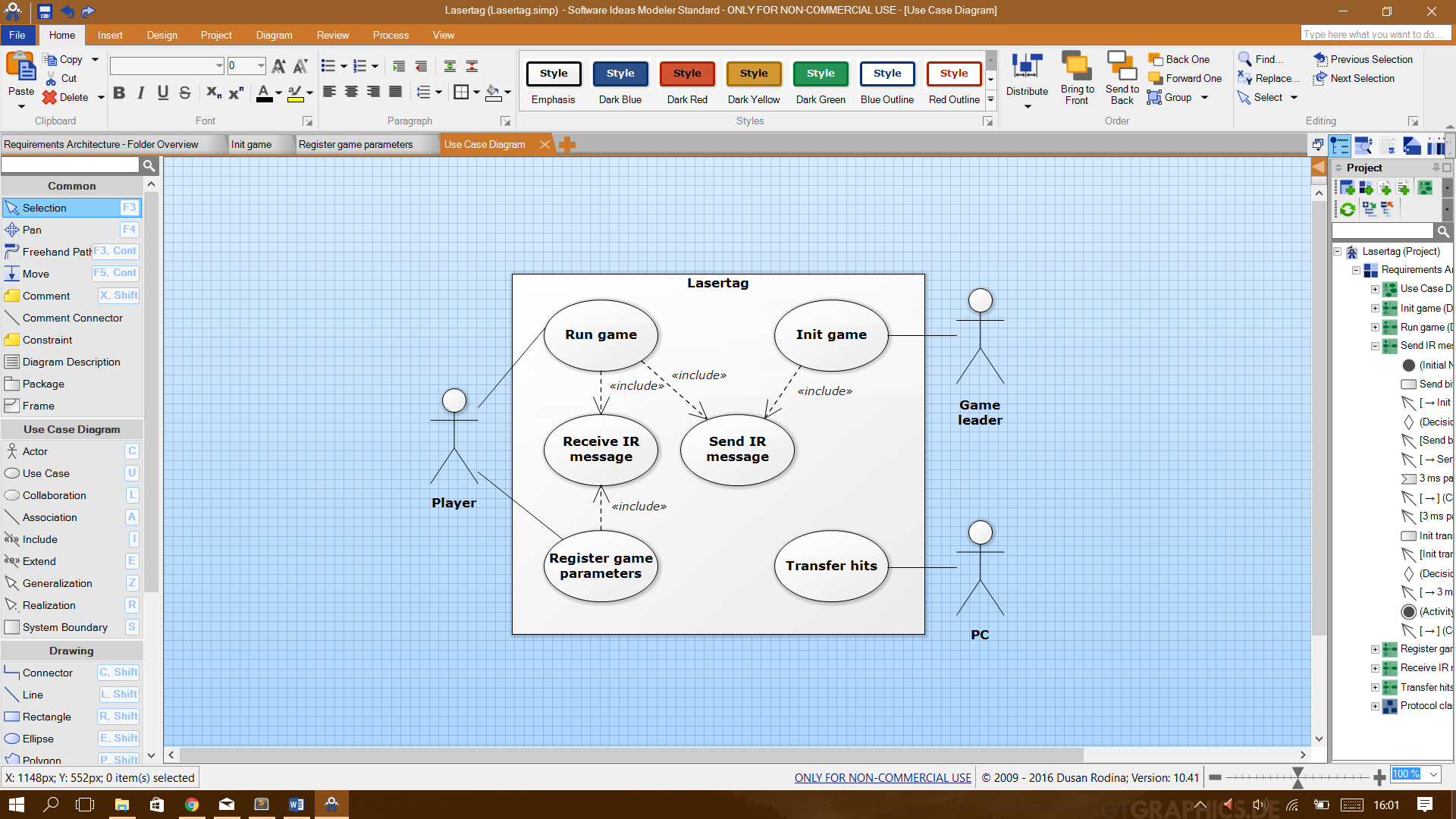
# Solution architecture

**Inleiding**

Hieronder staat de solution architecture van ons project. Hiervoor hebben we gebruik gemaakt van het volgende use-case diagram:



**Objectenlijst**

Voor de objectenlijst hebben we alle use-cases een object gegeven. Daarnaast hebben we geprobeerd alle actoren een boundary object te geven.

Hieronder staat onze objectenlijst:

*Boundary objecten:*

**Input: Output:**

* keypad - LCD
* IR\_receiver - IR\_sender
* fire\_button - speaker
* system\_time - rgd\_led

*Control objecten:*

**Coordinator: State depended control: Timer:**

* Run\_game - Init\_game - game\_time
* Transfer\_hits - Receive\_IR\_message

**-** Send\_IR\_message

**-** Register\_game\_parameters

**-** play\_sound

**-** LCD-chars

**-** led

*Entity objecten:*

* my\_player\_information
* received\_information

**Associaties**

Hieronder staat voor elk control object wat deze verzenden naar andere klassen en wat ze ontvangen

**Run\_game:**

Sturen:

Stuurt score naar LCD

Stuurt game\_time naar LCD

Stuurt 'je bent geraakt' naar LCD

Stuurt start\_sein naar game\_time

Stuurt schiet\_signaal naar play\_sound

Stuurt schiet\_signaal naar send\_IR\_message

Stuurt hit\_signaal naar play\_sound

Ontvangen:

Ontvangt systeem\_tijd van game\_time

Ontvangt score van register\_game\_parameters

Ontvangt schiet\_signaal van fire\_button

Ontvangt hit van received\_information

**Init\_game:**

Sturen:

Stuurt command naar send\_IR\_message

Ontvangen:

Ontvangt toetsen van keypad

**Receive\_IR\_message:**

Sturen:

Stuurt id en weapon naar received\_information

Stuurt command naar register\_game\_parameters

Ontvangen

Ontvangt bit van IR\_receiver

**Send\_IR\_message:**

Sturen:

Stuurt bit naar IR\_sender

Ontvangen:

Ontvangt id en weapon van my\_player\_information

**Register\_game\_parameters:**

Sturen:

Stuurt player\_number en weapon naar my\_player\_information

Stuurt playtime en start naar init\_information

Ontvangen:

Ontvangt player\_number van keypad

Ontvangt weapon van keypad

Ontvangt command van receive\_IR\_message

**Play\_sound:**

Sturen:

Stuurt decoded\_sound\_schiet naar speaker

Stuurt decoded\_sound\_hit naar speaker

Ontvangen:

Ontvangt schiet\_signaal van run\_game

Ontvangt hit\_signaal van run\_game

**LCD\_letters:**

Sturen:

Stuurt pixels\_char naar lcd

Ontvangen:

Ontvangt text van run\_game

**Game\_time:**

Sturen:

Stuurt overgebleven\_tijd naar init\_information.

Ontvangen:

Ontvangt systeem\_tijd van system\_time

Ontvangt systeem\_tijd\_start van run\_game

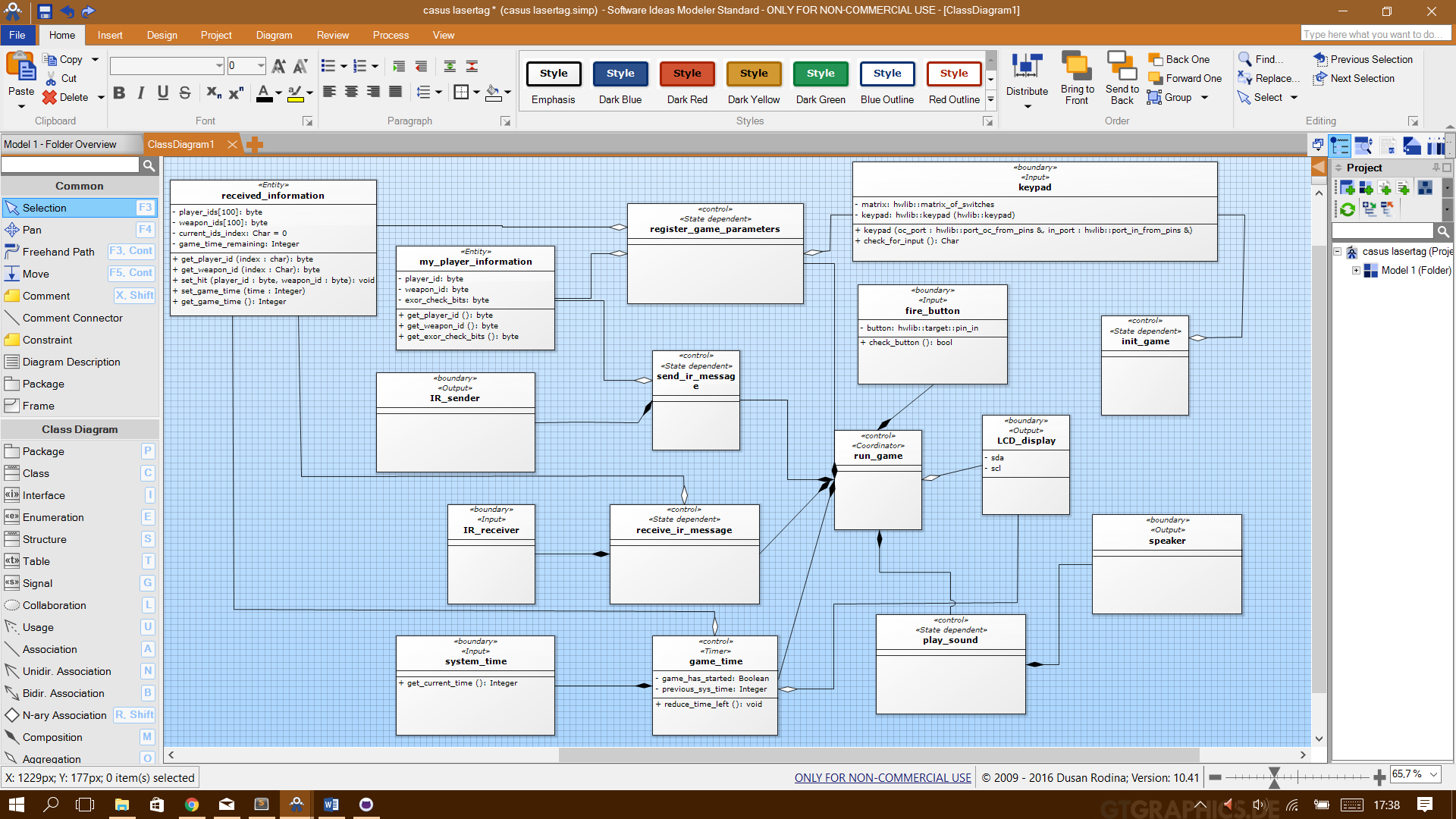
**Transfer\_hits:**

Sturen:

Stuurt score naar pc

Ontvangen:

Ontvangt score van received\_information

**Klassediagram:**

**Takenlijst**

Toen we de objectenlijst en het klassendiagram hadden hebben zijn we een takentabel gaan maken.

Hierbij hebben we eerst alle objecten in de objectlijst omgezet in een taak, waar we vervolgens, waar nodig, een periode en deadline aan toegevoegd hebben.

Hieronder vind je ons eerste takenlijst(hierbij hebben de taken nog dezelfde :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Taakbeschrijving | Soort taak | Periode | Deadline | Prioriteit |
| IR\_receiver | Asynchroon I/O | - | 0.1 ms | 0 |
| receive\_IR\_message | Asynchroon | - | 0.1 ms | 0 |
| fire\_button | Periodiek I/O | 50 ms | 0.2 ms | 1 |
| IR\_sender | Asynchroon I/O | - | 0.2 ms | 1 |
| send\_IR\_message | Asynchroon | - | 0.2 ms | 1 |
| run\_game | Asynchroon | - | ? | ? |
| game\_time | Asynchroon | - | 100 ms | 5 |
| LCD | Passief I/O | - | 50 ms | 4 |
| play\_sound | Passief | - | 200 ms | 6 |
| speaker | Passief I/O | - | 200 ms | 6 |
| keypad | Periodiek I/O | 50 ms | 5 ms | 2 |
| Init\_game | Asynchroon | - | 30 ms | 3 |
| register\_game\_parameter | Asynchroon | - | 30 ms | 3 |
| Rgb\_led\_task | Passief I/O | - | 400 ms | 7 |

Zelf shots\_fired task maken zodat run\_game er niet meer naar hoeft te kijken.

Hierna zijn we met cohesie gaan werken om het aantal taken te verminderen.

Na samenvoeging:

IR\_receiver\_task = IR\_receiver – receive\_IR\_message

IR sender task = fire button – IR\_sender – send IR message

LCD task = LCD – LCD\_char

Sound\_task = speaker – play-sound

Time task = game\_time

|  |  |  |  |
| --- | --- | --- | --- |
| taakbeschrijving | Cohesie: | uitleg | nieuw taak |
| IR\_receiver | Control | Wordt exclusief aangeroepen door receive\_IR\_message | IR\_receiver\_task |
| receive\_IR\_message | Control | Roept als enige de taak IR\_receiver aan | IR\_receiver\_task |
| fire\_button | Control | Wordt exclusief aangeroepen door send\_IR\_message | IR\_sender\_task |
| IR\_sender | Control | Wordt exclusief aangeroepen door send\_IR\_message | IR\_sender\_task |
| send\_IR\_message | Control | Roept als enige taken fire\_button en IR\_sender aan | IR\_sender |
| run\_game | - | - | - |
| game\_time | - | - | - |
| LCD | - | - | - |
| play\_sound | Control | Roept als enige taak speaker aan | Sound\_task |
| speaker | Control | Wordt exclusief aangeroepen door play\_sound | Sound\_task |
| keypad | - | - | - |
| Init\_game | - | - | - |
| register\_game\_parameter | - | - | - |
| Rgb\_led\_task | - | - |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Taakbeschrijving | Soort taak | Periode | Deadline | Prioriteit |
| IR\_receiver task | Asynchroon | - | 0.1 ms | 0 |
| IR\_sender\_task | Periodiek I/O /Asynchroon | 50 ms | 0.2ms | 1 |
| run\_game | Asynchroon | - | 10 ms | 3 |
| Game\_time | Periodiek I/O | 800 ms | 200 ms | 6 |
| LCD\_task | Passief I/O | - | 50 ms | 5 |
| Sound\_task | Passief I/O | - | 200 ms | 6 |
| keypad | Periodiek I/O | 50 ms | 5 ms | 2 |
| Init\_game | Asynchroon | - | 30 ms | 4 |
| register\_game\_parameter | Asynchroon | - | 30 ms | 4 |
| Rgb\_Led\_task | Passief | - | 200 ms | 6 |