|  |  |  |  |
| --- | --- | --- | --- |
| **Must haves** | **Should haves** | **Could haves** | **Won’t/would haves** |
| Het moet commando’s via infrarood kunnen verzenden | Het versturen van de informatie via infrarood moet foutloos gebeuren | Spelers kunnen beschikken over een aantal levens.  De speler is pas uitgeschakeld als hij/zij 0 levens over heeft. | Een houten p90 frame waar het systeem in zit. |
| Het moet commando’s via infrarood kunnen ontvangen en decoderen | Een werkend schermpje voor elke speler met informatie over: aantal levens dat je nog hebt (als deze functie geïmplementeerd wordt)  De cooldown nadat je geraakt bent | Schieten maakt een geluidje  (pew pew pew) | De communicatie met de game master arduino kan geregeld worden via een ander draadloos medium dan infrarood. |
| De game master Arduino moet het spel kunnen starten, score kunnen weergeven aan het einde. | Een speler is nadat hij geraakt is enige tijd uitgeschakeld: een cooldown.  Tijdens deze tijd kan de speler niet geraakt worden, of andere spelers beschieten. | Verschillende soorten wapens.  Wapens die sneller schieten dan andere maar minder schade doen.  Wapens die langzamer schieten maar meer schade doen. |  |
| Duidelijke Doxygen documentatie | Als een speler geraakt wordt, wordt er een geluidje afgespeeld. |  |  |
| Klassendiagram |  |  |  |
| Concurrency model |  |  |  |
| STDs |  |  |  |
| Github gebruik |  |  |  |
| Een demo van het embedded systeem |  |  |  |
| RTOS onderzoeksrapport |  |  |  |
| Werkverdeling en planning |  |  |  |
| Het infrarood signaal moet twee keer verstuurd worden, maar maximaal 1 keer geregistreerd worden. |  |  |  |
| Elke speler moet een eigen speler ID hebben die wordt meegestuurd bij het schieten, zodat aan het eind van het spel kan worden bepaald wie door wie is geraakt en hoeveel kills je hebt. |  |  |  |