# Project requirements

## Project scope

There are two main objectives of the practical part of this thesis:

1. Design and develop Quake II bot that is able to compete with human players and other third-party bots.
2. Apply on-line reinforcement learning methods to chosen bot’s decision problem.

The first one will allow us to test in practice basic methods of FPS AI programming and will provide a base for further development.

Completing the second objective will let us evaluate suitability of on-line reinforcement learning in this application as well as its implementing and testing complexity, which is especially interesting for commercial game developers.

Whether a bot is able to compete with human players and third-party bots will be verified with an experiment under conditions similar to a typical Quake II match.

## Assumptions

In this section basic assumptions of the practical part of the thesis are described.

A typical commercial bot AI is a complex program, for instance [1] or [2], and it is usually developed by a small group of people. This to a degree motivates the following assumptions, as having limited resources, the author had to focus on chosen, more specific aspects of bot development.

### QASE API

QASE API (described in chapter XXX ) allows rapid development of Quake II bots focusing on AI rather than on game-specific issues like communication with a game server.

### Navigation module

The navigation module will be kept as simple as possible while still being effective enough to compete with other bots and human players.

In order to achieve it, the QASE API’s waypoint map generation module will be used to facilitate the process of supplying the bot with a knowledge about a game map.

### Main focus

It is assumed that this thesis will focus on the combat skills of programmed bots.

In the early stage of the work, the author has noticed that it is relatively difficult to evaluate the quality of a particular bot’s navigation or general decision-making skills. At the same time the evaluation of bot’s combat skills can be quite direct, for instance, by measuring the inflicted damage to the opponent.

## Limitations

The main limitations of the project are determined by QASE API - all developed bots will be client-side bots.

Quake II server in order to keep all the clients up to date with the world information sends new messages to each client every 100 milliseconds. This creates a delay between bot’s perception and actions execution. For instance, if bot receives the information about enemy’s position at time *t*, and decides to shoot at him, the shooting will take effect at time *t+1*, at which the enemy may already be at different position.

On the other hand, a server-side bot may be programmed in such a way that it will react and perceive at the same, from other client’s point of view, time. This gives an important advantage to a server-side bot.

Another aspect of client-side bots is that the server informs them only about the world in their immediate surroundings, while the server-side bot can have access to full knowledge about the current game state. This includes positions of all the enemies and availability of items even in the most remote parts of the map, which again, makes it easier for a server-side bot.

Although the development of a client-side bot has more constraints, it is worth noticing that from a human player’s point of view the client-side bot may be seen as a more fair opponent in comparison with a server-side one.

[1] J. van Waveren, “The Quake III Arena Bot,” Delft University of Tehchnology, 2001.

[2] R. Straatman, W. van der Sterren, and A. Beij, *Killzone's AI: dynamic procedural combat tactics*, Amsterdam: Guerilla Games, 2005.