

# Eseményvezérelt Alkalmazásfejlesztés

## 2. Beadandó

Bittner Barnabás  
TC8TT8

2016. Április

# Contents

<b>1</b>	<b>Feladat</b>	<b>2</b>
<b>2</b>	<b>Programterv</b>	<b>2</b>
2.1	Osztálydiagram . . . . .	2
<b>3</b>	<b>További dokumentáció</b>	<b>4</b>

## 1 Feladat

Készítsünk programot, amellyel a következő játékot játszhatjuk. Adott egy  $n \times n$  mezőből álló erdő, amelyben Maci Lacival kell piknikkosarakra vadásznunk, amelyek a játékpályán helyezkednek el. A játék célja, hogy a piknikkosarakat minél gyorsabban begyűjtsük. A játékpályán a piknikkosarak mellett akadályok (pl. fa) is elhelyezkedhetnek, amelyekre nem léphetünk. A pályán emellett vadőrök is járőröznek, akik adott időközönként lépnek egy mezőt (vízszintesen, vagy függőlegesen). A járőrözés során egy megadott irányba haladnak egészen addig, amíg akadályba (vagy a pálya szélébe) nem ütköznek, ekkor megfordulnak, és visszafelé haladnak (tehát folyamatosan egy vonalban járőröznek). A vadőr járőrözés közben a vele szomszédos mezőket látja (átlósan is, azaz egy  $3 \times 3$ -as négyzetet). A játékos kezdetben a bal felső sarokban helyezkedik el, és vízszintesen, illetve függőlegesen mozoghat (egyesével) a pályán, a piknikkosárra való rálépéssel pedig felveheti azt. Ha Maci Lacit meglátja valamelyik vadőr, akkor a játékos veszít.

A pályák méretét, illetve felépítését (piknikkosarak, akadályok, vadőrök kezdőpozíciója) tárolhatjuk fájlban, vagy létrehozhatjuk véletlenszerűen (előre rögzített paraméterek mellett). A program legalább 3 különböző méretű pályát tartalmazzon. A program biztosítson lehetőséget új játék kezdésére a pálya kiválasztásával, valamint játék szüneteltetésére (ekkor nem telik az idő, és nem léphet a játékos). Ismerje fel, ha vége a játéknak, és jelezze, győzött, vagy veszített a játékos. A program játék közben folyamatosan jelezze ki a játékidőt, valamint a megszerzett piknikkosarak számát.

## 2 Programterv

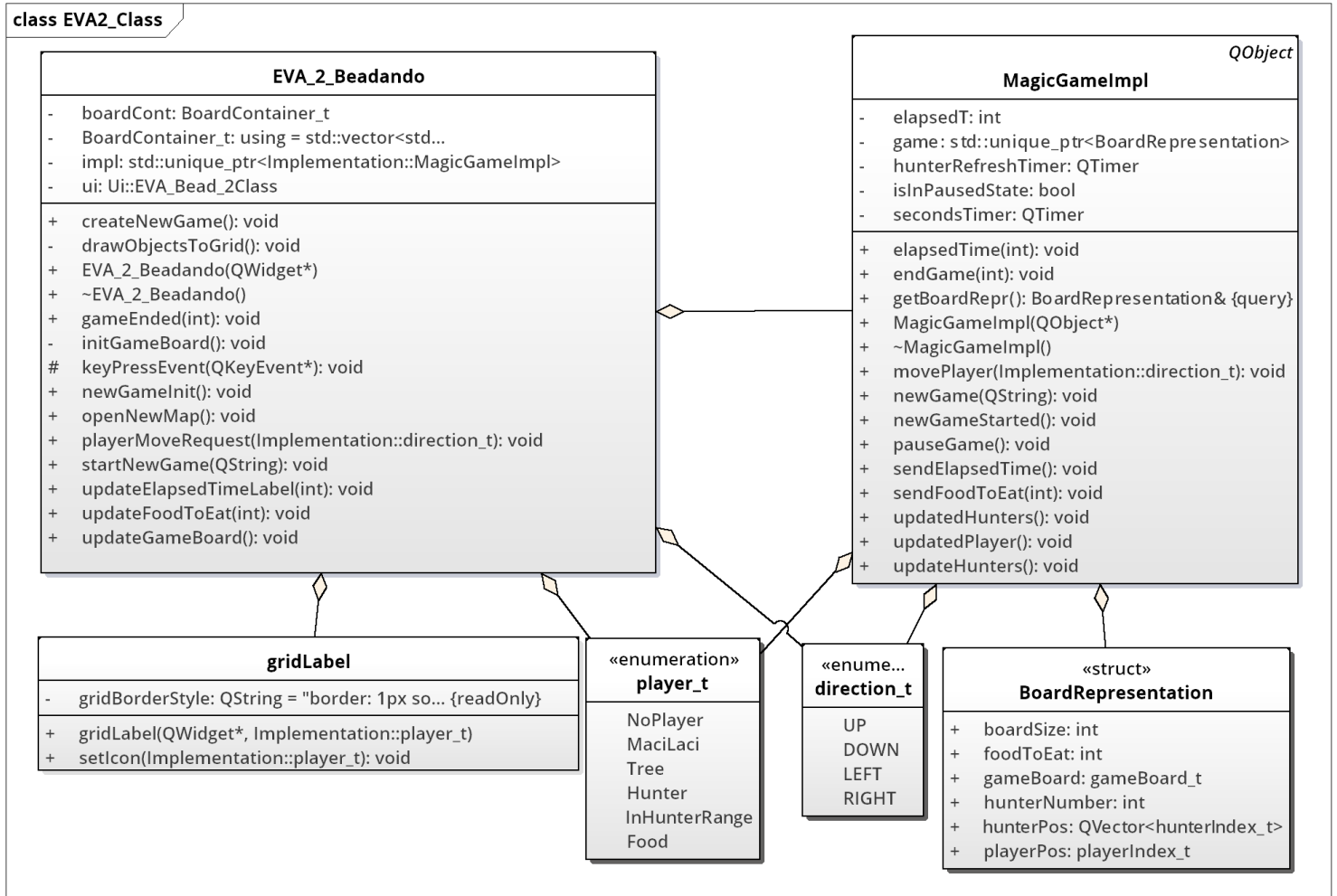
A feladat megoldásához a Qt keretrendszert fogjuk használni. A játék terve, hogy készítünk egy  $n \times n$ -es táblát, amelyen reprezentáljuk az objektumokat, majd a megadott időközönként frissítjük a vadőröket illetve, ha a játékos lép, akkor őt is, hogy a játék szabályainak eleget tegyen.

## 3 Tervezés

A program szerkezete két rétegre van bontva, a modell illetve, nézet. A modell (implementáció) feladata, hogy a program működését szabályozza, a felhasználói interakciók itt kezelődnek le, az egész játék működése ennek a résznek köszönhető.

Ezzel szemben a játék kinézetét kizárólagosan a nézet névtérben létrehozott objektumok alakítják ki. Ez a réteg teremt kapcsolatot a felhasználó és a

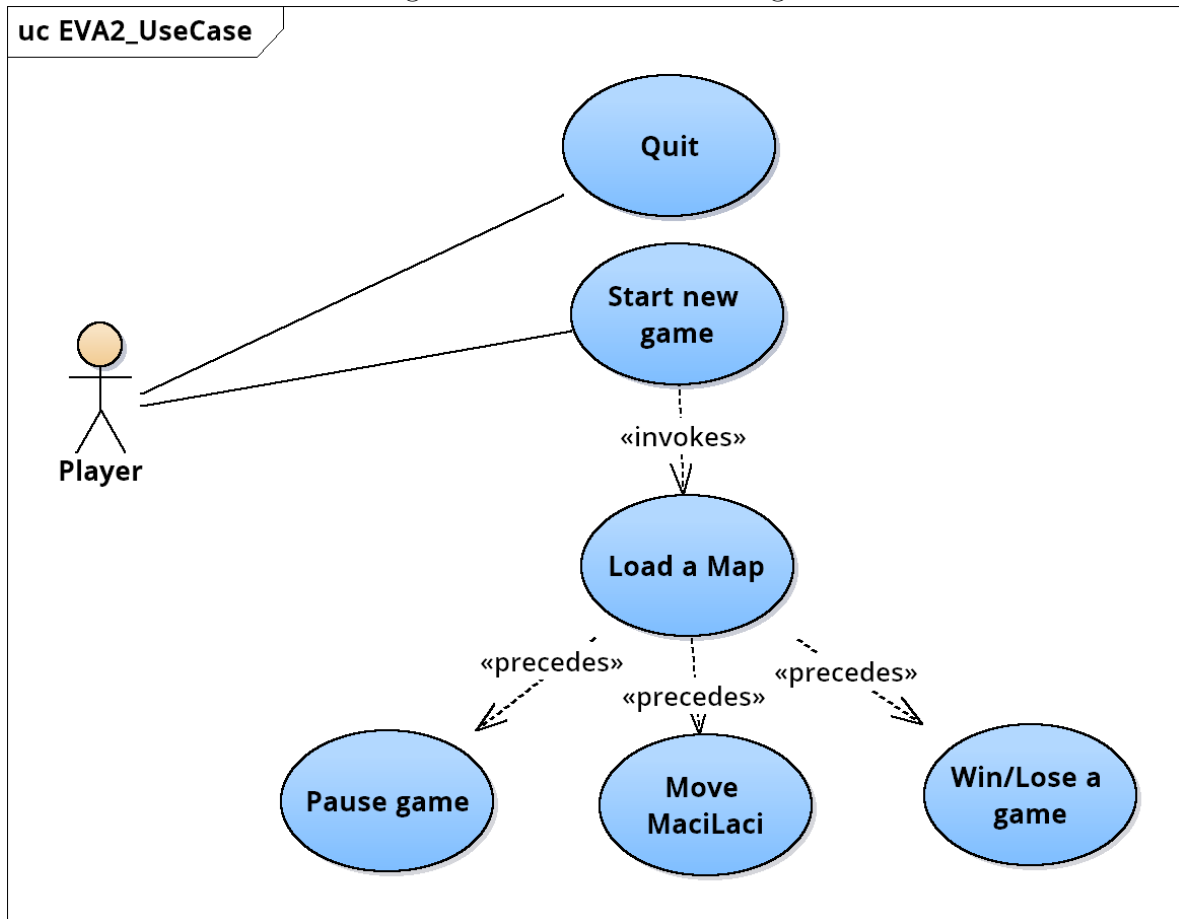
Figure 1: A feladat osztálydiagramja



modell között. Minden egyes funkcionalitás, egy a modellben definiált (és implementált) függvényt, vagy függvényoszt hajt végre a Qt keretrendszer-beli signal-slot metodika révén.

### 3.1 Osztálydiagram

Figure 2: Használati eset diagram



## 4 További dokumentáció

A programot mélységében jobban leíró, automatikusan generált dokumentáció megtalálható itt: [Doxygen dokumentáció](#)