Emergent Architecture Design (draft)

TI2805 Context Project, Computer Games, Group 1 (Building Blocks)

Gert Spek	Niels Bakker	Jehan da Camara
gspek	ncbakker	jdacamara
4216806	4161394	4207858

Maarten van Elsas	Leon Helsloot
mvanelsas	lhelsloot
4176898	4235991

May 15, 2014

The emergent architecture design is a continuously updated document describing the current state and envisioned design of the hardware and software architecture behind the *Building Blocks* game, created by group 1 of the *Computer Games* context project. The document contains a brief overview of the design goals influencing the architecture, followed by a high level description of used software components and the interaction between these components. Finally, a description of the hardware architecture necessary to facilitate the game is given, as well as a mapping between software and hardware components.

Contents

1 Introduction

1.1 Design goals

Apart from the design goals regarding the functionality of the product as described in the Design Document, some design goals with respect to the architecture of the game have to be taken into account. The main design goal is for the game to be usable by a large and diverse audience. This goal can be split up into two smaller goals: ease of use from the user's perspective and ease of setup from a server administrator's perspective.

1.1.1 Easy for the user

In order to make the game reachable for a large audience, effort must be made to make joining and playing the game straightforward to the user. Joining a game session must be as easy as scanning a QR code from the game application. Furthermore, placing blocks in the building must be intuitive to the user by visualizing where a block would be placed to avoid misplacement of that block.

1.1.2 Easy to setup the game

One of our main goals is that the game should be easy to setup. With just a placemant of a marker and running server. That can handle all the requests to join and a game and able to distribute the global structure of a team to a team player.

2 Software Architecture

When creating *Building Blocks* we wanted a game that would be applicable to big groups of people and it should also be easy to access.Because smartphones are very common and have a very broad potential we decided to built our game for smartphones. Because the majority of the smartphone market is Android and iOS, approximately 90% of the market[1], we decided to start by implementing our game for these two operating systems.

2.1 Programming languages and Development Environment

For *Building Blocks* we use the Unity3D game engine combined with the Vuforia SDK for the augmented reality libraries. Within Unity we decided to write the necessary scripts in C#. For all our server purposes we also use Unity. With Unity it is also possible to build directly to a connected phone, this way we can quickly see if our implementation has the correct behaviour.

2.2 Argumentation for the choice in programming languages and Development Environments

As already mentioned we are using the Unity package of Vuforia. First off we tried to use the Metaio SDK for the augmented reality but Metaio's augmented reality was quite jittery and thus we came across Vuforia which was more what we were aiming for. Unity was a easy choice because we already chosen Vuforia and since we are making a game it seemed like a good idea to use a game engine. As for our choice to use C# instead of the other programming languages is that in our opinion C# is better equiped to handle what we want to implement for *Building Blocks*.

2.3 Tests

To check if our game does what it was suppose to we carry out test on the code. For the testing of the code, Unity has a tool called Unity Test Tools with is obtainable in the Asset Store in Unity. This tool uses NUnit to carry out the test. So to use Unity Test Tools one would have to import this in the framework. Futhermore to fully test the code the use of Mocks is required. The mocks are created using the MoQ framework.

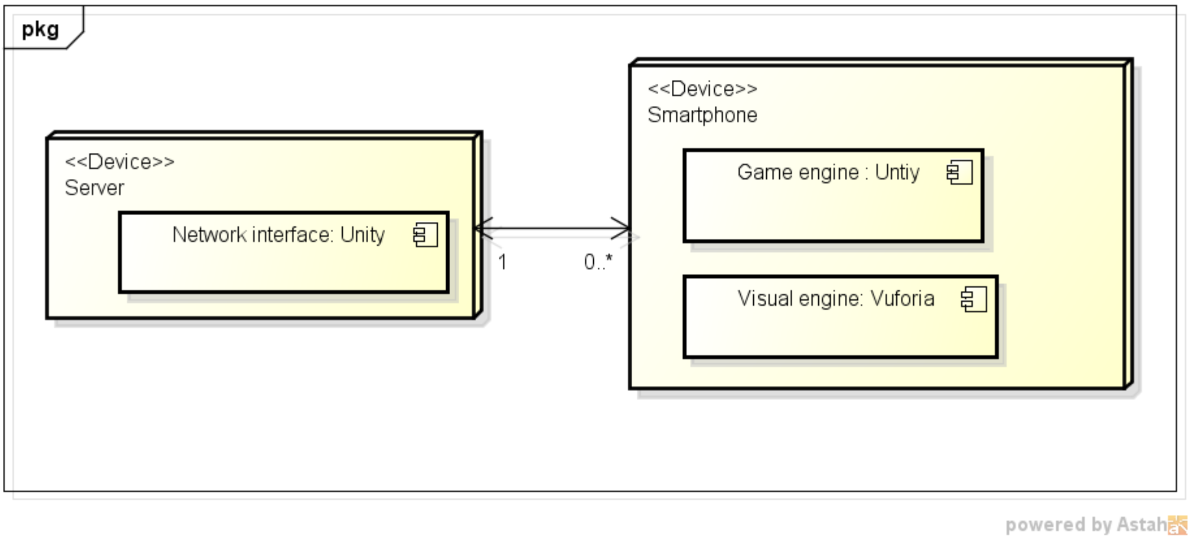
2.4 Code quality

To achieve a high code quality for the game we use GitHub and a continuous intergration server. With Maarten appointed as our lead programmer, his job is to make sure all pull requests to the master branch have decent tests and the code is properly formatted.

3 Hardware Architecture

3.1 Hardware / Software Mapping

For the *Building Blocks* every player or spectator needs a IOS smartphone or Andriod smartphone with a QR scanner. With will connect to server. The server will be running a network interface from Unity. On the smartphone will run game engine from Unity with a visual engine from Vuforia SDK. With is depicted in the first figure.

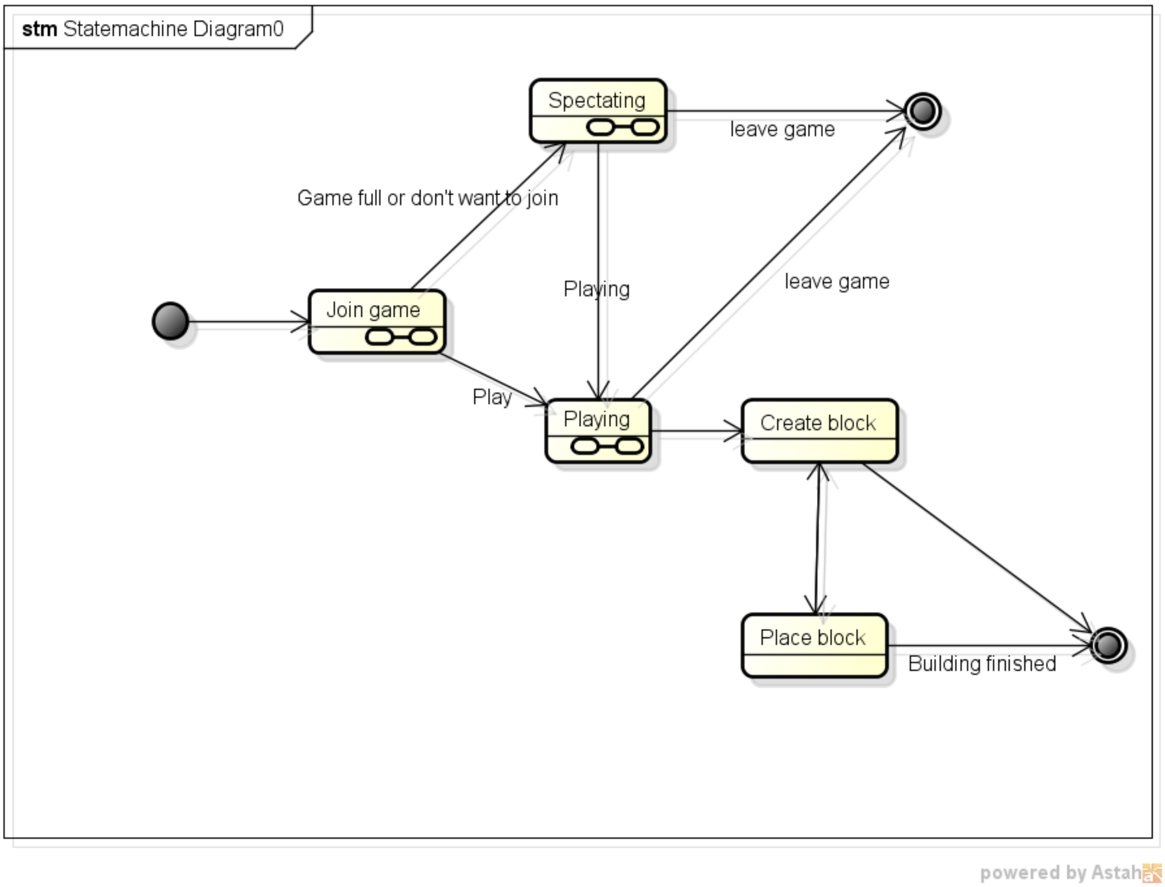


3.2 System concurrency

The server with the *Building Blocks* would have to handle a lot of things simultaneously. The server should be able to differ if the person is a player or spectator. Because player can only see there structure and spectators can see every teams structures. Also the server has make sure that global structure with the team is working on is update real time, to make sure that two players are not working on the same thing. Finally the server has to be aware with use has with block and with color the block is.

3.3 Systemstatus

In figure 2 all the different states for *Building Blocks* game.A person can start playing. And when they are playing they can create a block. After a block is created they can place a block in the game and if the structure is complete the game has ended. Or is the player is playing the player has option to leave at any time. A person can also start as a spectator and have to possibility to join the game at any point if the maximum amount of players is not exceeded.



References

[1] Llamas, R.,Reith, R. & Shirer, M.(2013, February 14). *Smartphone market*. Retrieved from `http://www.idc.com/getdoc.jsp?containerId=prUS23946013#.USuIPVrF2Bg/`.