**MM systems 5 : Ganzenbord**

**1. Inleiding**

Voor het vak MM Systems 5 moesten we dit semester een multiplayer bordspel maken voor de Windows Phone in combinatie met Windows Azure. Wij, Kevin Heyvaert en Joris Huybrechts hebben de opdracht samen aangenomen en we hebben besloten om het klassieke 'Ganzenbord' te maken.

**2. Spelregels**

**Uitleg van** <http://nl.wikipedia.org/wiki/Ganzenbord>

Het bestaat uit een speelbord, een dobbelsteen en enkele pionnen van verschillende kleuren. De bedoeling van het spel is in zo weinig mogelijk beurten een pion van het begin naar het eind van een reeks velden te voeren, waarbij elke speler in elke beurt zijn pion zoveel velden moet verplaatsen als men ogen gooit met de dobbelstenen. Het speelbord is voorzien van een aantal speciale velden zoals de put en de gevangenis, die de speler hinderen op de weg naar zijn einddoel.

Ganzenbord heeft een speelvlak van 63 velden. Vakjes met een speciale betekenis zijn:

|  |  |  |
| --- | --- | --- |
| **nummer** | **naam** | **betekenis** |
| 6 | brug | Ga verder naar 12 |
| 19 | herberg | Een beurt overslaan |
| 31 | put | Wie hier komt moet er blijven tot een andere speler er komt. Degene die er het eerst was speelt dan verder. |
| 42 | doolhof of doornstruik | Terug naar 39 |
| 52 | gevangenis | Zelfde als 31 |
| 58 | dood | Terug naar begin, opnieuw beginnen |
| 63 | einde | Wie hier als eerste komt heeft gewonnen |

Op sommige hokjes staat een gans afgebeeld. Het zijn die van de negenvouden en de negenvouden minus vier, dus de hokjes 5, 9, 14, 18, 23, 27, 32, 36, 41, 45, 50, 54, 59. Wie hierop terechtkomt, moet hetzelfde aantal ogen verder tellen. Het is dus niet mogelijk dat een speler zijn beurt beëindigt op een hokje met een gans.

Wie te veel ogen gooit en daardoor voorbij 63 zou spelen, moet vanaf 63 weer terugspelen. Dit vergroot het risico dat men op het gevreesde hokje 58 of 52 terechtkomt. Komt men bij het terugtellen op een hokje met een gans, dan telt men weer het gegooide aantal ogen terug. Wie bijvoorbeeld op 60 staat en 7 gooit, komt in de gevangenis.

Wie bij de eerste worp een 5 en een 4 gooit, gaat meteen door naar 53. Wie bij de eerste worp een 6 en een 3 gooit, gaat door naar 26. Als deze regel er niet was, zouden deze spelers, via de ganzen, meteen doorlopen naar het einde.

Als twee spelers het spel spelen, kan het mogelijk zijn dat één speler in de put terechtkomt en de ander in de gevangenis. Er is dan geen mogelijkheid meer om er een andere speler te laten komen, waardoor de partij in een gelijkspel eindigt. Dit is echter een zeldzame situatie.

**3. Use Cases**

|  |  |
| --- | --- |
| **Use Case ID**: | 1 |
| **Use Case Name:** | Inloggen op spel |
| **Priority** | 2 |

|  |  |
| --- | --- |
|  |  |
| **Description:** | Inloggen met username + password op spel. Username met bijhorend paswoord wordt gecontroleerd met de data die per speler in de data base staat. |
| **Trigger:** | The need for playing a bord game |
| **Preconditions:** | Internetconnectie + gebruiker zit in de database |
| **Postconditions:** | Ingelogd op de applicatie |
| **Normal Flow:** |  |
| **Exceptions:** | Geen internetverbinding / verbinding met database / user bestaat niet |

**Alternatieve flow**

**A. Wat als username/password combination incorect is**

1. System validation of the username/password combination fails due to incorrect entry.

2. Systems asks the user to re-enter the username/password combination.

3. Go back to basic flow met ID nummer 1.

**B. Gebruiker bestaat niet in de database**

1. System validation finds that the user record does not exist in the database.

2. Systems alerts the user that their record does not exist.

|  |  |
| --- | --- |
| **Use Case ID**: | 2 |
| **Use Case Name:** | Zien van speloverzicht |
| **Priority** | 1 |

3. Go back to basic flow met ID nummer 1.

|  |  |
| --- | --- |
| **Description:** | UI : zien van het bordspel op scherm : uitvergroot en in detail + alle andere spelers zien (positie) |
| **Trigger:** | The need for playing a bord game |
| **Preconditions:** | Inloggen op de applicatie + verbinding met de database |
| **Postconditions:** | Spelen van het bordspel met meerdere spelers + het spel volgen |
| **Normal Flow:** |  |
| **Exceptions:** | Geen internetverbinding of geen verbinding met database |

Voor verder in detail te gaan van het bordspel, hebben we gedacht om het "bord" in stukken (vierkanten) te verdelen en elk een chronologisch nummer te geven. Want in een bordspel wordt meestal een cyclus doorlopen in 1 richting. Dus zo kan men beter de volgende stap op het scherm tonen. Het is eigenlijk bijhouden van plaatsen in een stijgende lijn. Want anders is het onmogelijk om het (volledige) spel te volgen op een klein scherm van een smartphone.

|  |  |
| --- | --- |
| **Use Case ID**: | 3 |
| **Use Case Name:** | Eigen zet doen |
| **Priority** | 1 |

|  |  |
| --- | --- |
|  |  |
| **Description:** | Speler kan dobbelen op zijn beurt en kan een zet doen |
| **Trigger:** | Vorige speler heeft zijn beurt afgerond |
| **Preconditions:** | Verbinding met database + beurt van de speler |
| **Postconditions:** | Speler heeft zijn zet gedaan : bijhorende acties bij die zet worden gedaan ( pion kwijt, plaatsen terug gezet,...) + beurt aan volgende speler |
| **Normal Flow:** |  |
| **Exceptions:** | Geen internetverbinding of geen verbinding met database |

Voor het 'soepele' verloop van het spel op de Windows Phone, gaan we moeten bepalen hoe we de data naar de database gaan pushen en hoeveel keren we data gaan vragen aan de database. We moeten data krijgen om te zien wat de andere spelers hebben gedaan in het spel. En voor een snel (en soepel) verloop, gaan we dat vragen op bepaalde tijdstippen of bij bepaalde handelingen ( vb : bij maken van zet van een andere speler)

|  |  |
| --- | --- |
| **Use Case ID**: | 4 |
| **Use Case Name:** | Zetten van andere spelers zien en binnenkrijgen |
| **Priority** | 1 |

|  |  |
| --- | --- |
|  |  |
| **Description:** | Zien wie aan de beurt is + wat voor zet ze hebben gedaan |
| **Trigger:** | Eigen beurt is verlopen |
| **Preconditions:** | Eigen beurt is verlopen en zet is naar data base gepusht |
| **Postconditions:** | Zien wat het verdere spelverloop is + eigen beurt |
| **Normal Flow:** |  |
| **Exceptions:** | Geen connectie met de data base |

Hier gaan we bepalen wat we gaan doen als men het einde van het spel bereikt of klaar is met spelen.

|  |  |
| --- | --- |
| **Use Case ID**: | 5 |
| **Use Case Name:** | Einde spel aangeven + 'score's' van tegenspelers |
| **Priority** | 2 |

|  |  |
| --- | --- |
|  |  |
| **Description:** | Einde van spel tonen met winnaar en verliezers ( met bijhorende eindplaatsen en aantal zetten?) |
| **Trigger:** | Einde van spel of dichste bij einde |
| **Preconditions:** | Spel is bijna uitgespeeld |
| **Postconditions:** | Spel afsluiten + winnaar en verliezers tonen (met bijhorende plaatsen) + (Win/verlies bijhouden in database) |
| **Normal Flow:** |  |
| **Exceptions:** | Geen internetverbinding of geen verbinding met database |

We kunnen ook de score van de verschillende spelers laten zien in een aparte pagina van de applicatie. We pullen de data per speler en laten die zien in een lijst : Van hoge score naar lage score.

|  |  |
| --- | --- |
| **Use Case ID**: | 6 |
| **Use Case Name:** | Tonen van score lijst |
| **Priority** | 4 |

|  |  |
| --- | --- |
|  |  |
| **Description:** | Tonen van spelers en hun aantal win/verlies tonen in lijst |
| **Trigger:** | Einde van spel of aparte pagina/toets in applicatie |
| **Preconditions:** | Klikken op pagina 'score list' + internetconnectie met database |
| **Postconditions:** | Tonen van 'scores' per speler |
| **Normal Flow:** |  |
| **Exceptions:** | Geen internetverbinding of geen verbinding met database |

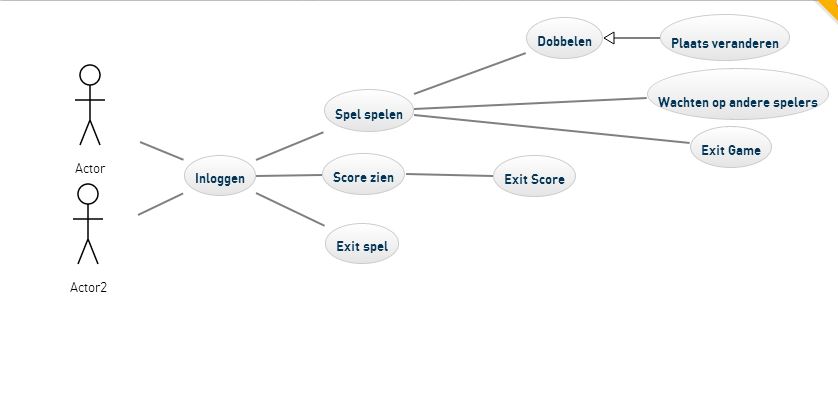
|  |  |
| --- | --- |
| **Use Case ID**: | 7 |
| **Use Case Name:** | Exit |
| **Priority** | 1 |

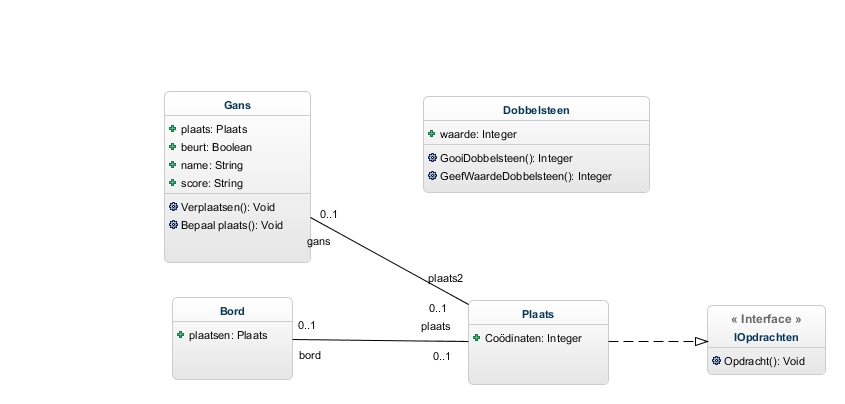
|  |  |
| --- | --- |
|  |  |
| **Description:** | Gebruiker kan het spel verlaten met exit button |
| **Trigger:** | Indrukken van 'exit' toets |
| **Preconditions:** | Speler is ingelogd |
| **Postconditions:** | Speler is niet meer ingelogd |
| **Normal Flow:** |  |
| **Exceptions:** | Geen internetverbinding of geen verbinding met database |

|  |  |
| --- | --- |
| **Use Case ID**: | 8 |
| **Use Case Name:** | Dobbelsteen gooien |
| **Priority** | 1 |

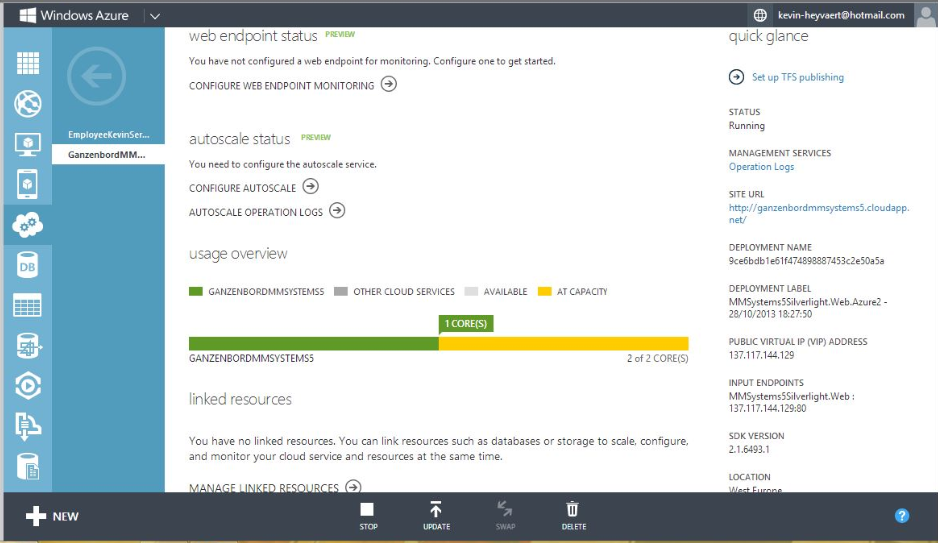
|  |  |
| --- | --- |
|  |  |
| **Description:** | Spel vraagt om dobbelsteen te gooien en speler kan gooien. |
| **Trigger:** | Indrukken van 'gooi' knop |
| **Preconditions:** | Speler is aan de beurt |
| **Postconditions:** | Speler krijgt een nummer tussen 1-6 of 2-12 |
| **Normal Flow:** |  |
| **Exceptions:** | Geen internetverbinding of geen verbinding met database  Speler krijgt geen nummer / server heeft geen nummer opgeslagen |

Hieronder staan de use cases nog visueel voorgesteld met een use case met actors + een klassediagram.

****

****

**4. Service**

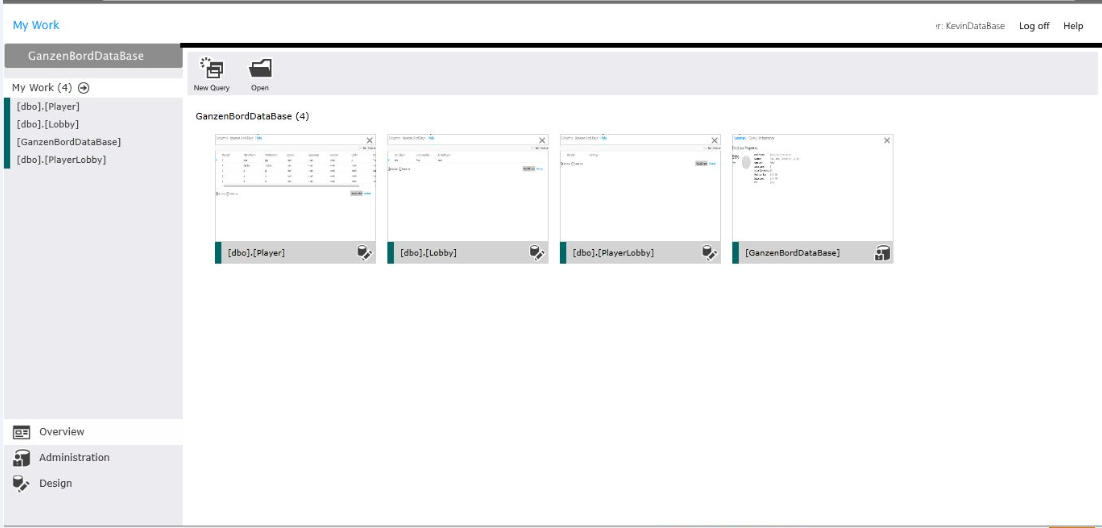
****

Wij hebben een service op Windows Azure gemaakt voor ons ganzenbordspel. Deze hebben we GanzenbordMMSystems5 genoemd. In deze service zetten we methodes die we niet nodig hebben op de phone. De service koppelen we aan de aan de phone applicatie. Daardoor kan de phone applicatie alle methodes in de service aanroepen. Deze worden dan aan geroepen met een event.

De volgende methodes worden aangeroepen via de service :

* Inloggen : inloggen op spel met username en wachtwoord player
* MaakAccount : username en wachtwoord voor player maken
* JoinLobby : joinen in een bestaande lobby
* MaakLobby : Maken van Lobby met player username
* ExitLobby : uit een lobby gaan
* Stophost : host van lobby kan de lobby sluiten
* Gooi : dobbelsteen waarde geven en zo plaats (locatie) speler verder bepalen
* BeschikbareLobbys : geeft een lijst met beschikbare lobby's
* LobbyInfo : geeft een lijst met de spelers in een lobby
* updatelobby : gaat wanneer een speler wordt toegevoegd, de lijst per lobby updaten

**5. Database**

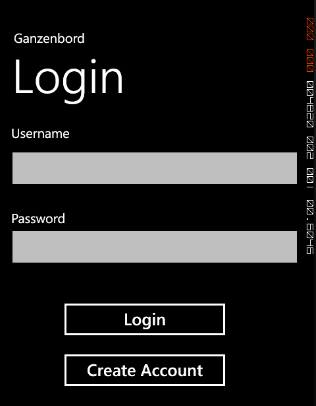
****

Voor het spel hebben we ook een database gemaakt om data (omtrend het spel) op te slaan. Hieronder vindt je de verschillende lijsten die we in de database gebruiken :

* **Player** : PlayerId, PlayerNaam, Wachtwoord, Locatie, Gewonnen, Verloren, Lobby, IsHost
* **Lobby** :HostPlayer, Can/JoinLobby, AantalPlayers
* **PlayerLobby** :
* **Gamestate** : ID, Update, YourTurn
* **Update** : HostName

**6. Ganzenbord**

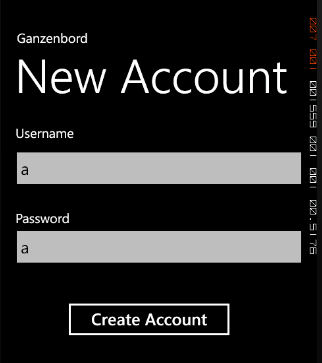
**6.1 Inloggen**

****

* Inloggen : met username en paswoord

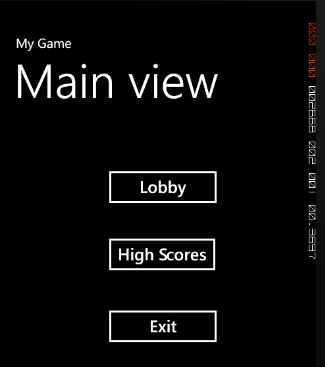
--> wordt gecontroleerd met query in de database voor te zien of deze erin staat en dan gaat de applicatie naar de main view.

* Create account : ga naar New Account pagina

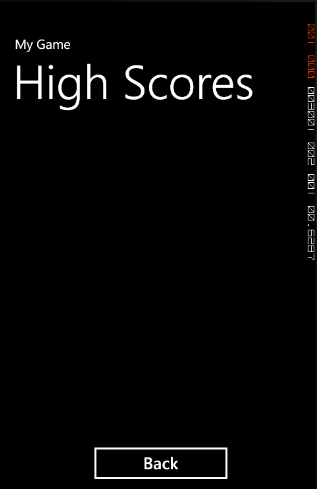
**6.2 Maken van een nieuw account**

* Maak nieuw account

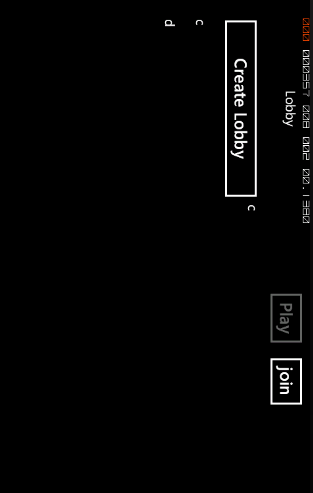
--> in de textboxen wordt er een nieuwe username en wachtwoord ingegeven en deze worden dan met een query in de data base gezet. En dan gaat de applicatie inloggen en naar main view gaan.

**6.3 Main view**

* Lobby : ga naar lobby pagina
* High scores : toon high scores van alle spelers in een lijst. (met behulp van query alles in een list krijgen)
* Exit : ga terug naar inlog pagina

**6.4 High Scores**

* Na laden van pagina komt er een list van alle spelers met hun gewonnen en verloren spelletjes (met query oproepen)
* Back : ga terug naar main view

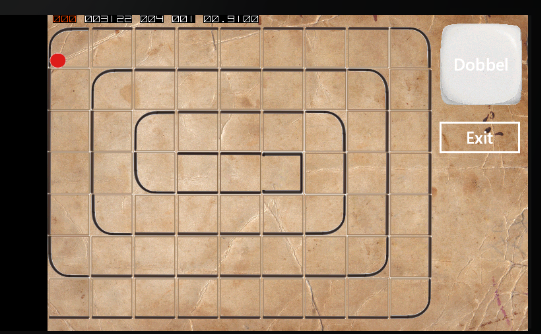
**6.5 Lobby**

* Bij laden van de pagina komen de beschikbare lobby's tevoorschijn in de linkse list. Lobby namen zijn de namen van de hostplayers die de lobby hebben aangemaakt.
* Als men op 1 van de items duwt in die list (vb. Textbox) dan komt er in de rechtse list een lijst met de spelers die in die lobby zitten ( host + 3 andere spelers)
* Als de host kiest voor een andere lobby, dan zal zijn (al dan niet) aangemaakte lobby verdwijnen en dan zal deze in de lobby zitten van een andere hostplayer

--> met query de lobby property van de speler veranderen + 'ishost' property veranderen)

# Create lobby : speler zal een lobby aanmaken die de naam van de hostplayer krijgt.

--> met query de lobby property van de speler veranderen + ishost property veranderen)

**6.6 Spelbord**

* Dobbel : dobbelsteen aanroepen via service en via service ook de plaats bepalen van de speler. Op deze manier altijd de game status property's aanpassen. En dan deze data (zoals locatie) pushen naar de database.
* Om de beurt gaan we vragen aan de service vragen wie zijn beurt het is, de locatie van andere spelers opvragen + ook polling (tijd bepalen --> vb. 5s )
* ...