

## 3BEES&ME



## Product backlog

USER STORIES	GEWICHT
BACKEND: Javalin server klaarzetten	10
HTML: login- en registratiescherm maken	17
CSS: login- en registratiescherm opmaken	30
JS: fetch voor registratie	15
JS: fetch voor login	15
MySQL: Database opzetten	29
MOCKDATA: Login	10
MOCKDATA: Registratie	10
BACKEND: login	35
BACKEND: registratie	40
HTML: startscherm maken	18
CSS: startscherm opmaken	20
JS: click 'Play game'	6
JS: click 'Deck builder'	6
JS: click 'Options'	6
JS: click 'Quit :('	6
HTML: Option menu	15
CSS: Option menu	16
(JS: Sound)	50
JS: click credits	6
HTML: credits	23
CSS: credits	12
JS: click back	6
HTML: Loading screen	5
CSS: Loading screen	5
JS: Loading screen: Animation	15
HTML: Play game: Hero selector	23
CSS: Play Game: Select Hero	23
JS: Alle heroes opvragen van server en weergeven	30
JS: Hero naar server sturen	16
BACKEND: Alle heroes sturen naar pagina	17
BACKEND: Geselecteerde hero opslaan	10
HTML: Play Game: Select deck	22
CSS: Play Game: Select deck	9
JS: Alle opgeslagen decks ophalen (of basic deck)	15
JS: Deck naar server sturen	14
BACKEND: Alle decks ophalen van MySQL database	26
BACKEND: Alle opgeslagen decks naar pagina sturen	15

BACKEND: Geselecteerd deck opslaan	10
MOCKDATA: Decks	10
HTML: VS scherm maken	3
CSS: VS scherm maken	5
JS: Hero en Enemy ophalen	12
BACKEND: Hero en Enemy naar pagina sturen	14
MOCKDATA: VS scherm: Hero en Enemy	4
HTML: speelveld	25
CSS: beginspeelveld onzichtbare clickables	13
CSS: beginspeelveld onzichtbare clickables upgraden met schijn, veranderen uiterlijk	30
CSS: Hover over kaart in hand (zoom achtig gedoe)	25
CSS: Hover over heropower	25
CSS: Hover over Enemy power	25
HTML + CSS: Uitleg bij hover over mana crystals	15
HTML + CSS: Uitleg hover bij deck (hoeveel kaarten in hand en deck)	15
HTML + CSS: Uitleg over bij enemy deck	15
CSS: Hover over battlelog: Achtergrond wordt donkerder, show card history	33
CSS: hover over alle clickables: white glow	5
JS: Heroes weergeven op speelveld	24
JS: Click 'End turn': Stuur naar server	13
JS: tijd check naar server sturen: na 50 sec: toon lont (20 sec) -> End turn	40
JS: Sleep kaart op speelveld (wanneer het je beurt is)	35
JS: Selecteer target (ability, weapon, hero spell)	20
JS: Begin spel: Kaart(en) wisselen: Vraag nieuwe kaart(en) aan server	19
JS: Begin spel: vraag beginner op van server	13
JS: Val hero of minion aan: Sturen naar server	17
JS: Kaarten in hand weergeven	13
JS: Deck grootte updaten	5
JS: Mana updaten	5
JS: Battlelog opvragen	30
JS: Enemy: Vraag aantal kaarten in hand op	12
JS: Enemy: Update: Hover over kaart, speel kaart/...	35
BACKEND: autoplay AI op End turn (request of time up)	45
BACKEND: Resterende tijd beurt naar pagina sturen	35
BACKEND: Val minion of hero aan + check voor TAUNTS (erna kijken of minion/ Hero dood is)	25
BACKEND: Random damage implementeren (tijdens kijken of minion/Hero dood is)	28
BACKEND: Random generate beginner	5
BACKEND: Uitvoeren van special abilities	33
BACKEND: Abilities	20
BACKEND: Battlecry	23
BACKEND: Charge	22
BACKEND: Stealth	23
BACKEND: Devine Shield	21
BACKEND: Deathrattle	24
BACKEND: Freeze	22

BACKEND: Taunt	21
BACKEND: Enrage	24
BACKEND: Windfury	21
BACKEND: Acties uitvoeren bij kaart op veld	45
BACKEND: Weapons implementeren	28
BACKEND: Heropowers implementeren	27
BACKEND: Battlefield implementeren (bijhouden gespeelde kaarten)	30
BACKEND: Mana systeem implementeren	18
BACKEND: Handkaarten bijhouden	14
BACKEND: Deckkaarten (krijg dit al van voor het spel start)	12
BACKEND: battlelog naar pagina sturen	15
BACKEND: Enemy autogenerate deck	24
BACKEND: Enemy: Update aantal kaarten in hand	13
BACKEND: Enemy: Update: hover over kaart, Speel kaart/...	50
BACKEND: Battlelog updaten	20
MOCKDATA: Beurt bepaling + doorsturen naar pagina	12
MOCKDATA: End of turn	10
MOCKDATA: timer for end turn	14
MOCKDATA: Minion aanval	16
MOCKDATA: Hero aanval (Weapon mock)	17
MOCKDATA: Dubbel aanval principe	20
MOCKDATA: alle abilities	26
MOCKDATA: Spellcard mocken	32
MOCKDATA: Not enough mana mocken	18
MOCKDATA: Deck mocken	19
MOCKDATA: Kaarten mocken	11
MOCKDATA: Mock of attackability of minion	21
MOCKDATA: Mock of Winning or Losing	23
MOCKDATA: Mock of Minions on playing field	14
MOCKDATA: Mock of Cards in hand	13
MOCKDATA: Mock of Random damage	11
MOCKDATA: TAUNT	18
MOCKDATA: Battlelog	17
HTML: Deckbuilder (leeg deck)	40
CSS: leeg layout	45
CSS: Als er kaarten in deck zitten	25
CSS: grote info van kaarten	17
JS: fetch alle kaarten en geef ze weer	33
JS: back click (kan van overall komen ;) )	35
JS: 'New deck' click	6
JS: search	6
JS: sort all cards	6
JS: filter cards	6
JS: deck switching	6
JS: preview deck filters	6
JS: doubleclick to add	19
JS: doubleclick to remove	19

JS: drag to add	30
JS: drag to remove	30
JS: rightclick to view info	20
JS: Toon gekregen warnings van server	10
JS: Stuur save request naar server	6
JS: Scroll/pijltjes deck kaarten	25
BACKEND: Alle kaarten aanpassen aan geselecteerde hero	35
BACKEND: Switch hero or deck: Stuur een warning wnr deck niet is opgeslaan	25
BACKEND: Save deck wanneer requested	30
BACKEND: add a card	13
BACKEND: deleting a card	13
BACKEND: filters	35
BACKEND: sort	35
BACKEND: search	30
BACKEND: give all cards	50
MOCKDATA: cards	22
BACKEND: Genereer een standaarddeck per hero	34
HTML: tutorial kaders	35
CSS: tutorial kaders	13
JS: tutorial kaders (click 'next' -> volgende kader)	29

#### VOLLEDIG TOTAAL

3000

HTML	CSS	JS	MOCKDATA	TOTAAL
248.5	398.5	791	368	1806

## Sprint 01:

### *Sprint backlog*

USER STORIES	GEWICHT
HTML: login- en registratiescherm maken	17
CSS: login- en registratiescherm opmaken	30
JS: fetch voor registratie	15
JS: fetch voor login	15
HTML: startscherm maken	18
CSS: startscherm opmaken	20
JS: click 'Play game'	6
JS: click 'Deck builder'	6
JS: click 'Options'	6
JS: click 'Quit :('	6
HTML: Option menu	15
CSS: Option menu	16
(JS: Sound)	50
JS: click credits	6
HTML: credits	23
CSS: credits	12
JS: click back	6
HTML: Loading screen	5
CSS: Loading screen	5
JS: Loading screen: Animation	15
HTML: Play game: Hero selector	23
CSS: Play Game: Select Hero	23
JS: Alle heroes opvragen van server en weergeven	30
JS: Hero naar server sturen	16
HTML: Play Game: Select deck	22
CSS: Play Game: Select deck	9
JS: Alle opgeslagen decks ophalen (of basic deck)	15
JS: Deck naar server sturen	14
MOCKDATA: Decks	10
HTML: VS scherm maken	3
CSS: VS scherm maken	5
JS: Hero en Enemy ophalen	12
MOCKDATA: VS scherm: Hero en Enemy	4
HTML: speelveld	25
CSS: beginspeelveld onzichtbare clickables	13

CSS: beginspeelveld onzichtbare clickables upgraden met schijn, veranderen uiterlijk	30
CSS: Hover over kaart in hand (zoom achtig gedoe)	25
CSS: Hover over heropower	25
CSS: Hover over Enemy power	25
HTML + CSS: Uitleg bij hover over mana crystals	15
HTML + CSS: Uitleg hover bij deck (hoeveel kaarten in hand en deck)	15
HTML + CSS: Uitleg over bij enemy deck	15
CSS: Hover over battlelog: Achtergrond wordt donkerder, show card history	33
CSS: hover over alle clickables: white glow	5
JS: Heroes weergeven op speelveld	24
JS: Click 'End turn': Stuur naar server	13
JS: tijd check naar server sturen: na 50 sec: toon lont (20 sec) -> End turn	40
JS: Sleep kaart op speelveld (wanneer het je beurt is)	35
JS: Selecteer target (abilty, weapon, hero spell)	20
JS: Begin spel: Kaart(en) wisselen: Vraag nieuwe kaart(en) aan server	19
JS: Begin spel: vraag beginner op van server	13
JS: Val hero of minion aan: Sturen naar server	17
JS: Kaarten in hand weergeven	13
JS: Deck grootte updaten	5
JS: Mana updaten	5
JS: Battlelog opvragen	30
JS: Enemy: Vraag aantal kaarten in hand op	12
JS: Enemy: Update: Hover over kaart, speel kaart/...	35
MOCKDATA: Beurt bepaling + doorsturen naar pagina	12
MOCKDATA: End of turn	10
MOCKDATA: timer for end turn	14
MOCKDATA: Minion aanval	16
MOCKDATA: Hero aanval (Weapon mock)	17
MOCKDATA: Dubbel aanval principe	20
MOCKDATA: alle abilities	26
MOCKDATA: Spellcard mocken	32
MOCKDATA: Not enough mana mocken	18
MOCKDATA: Deck mocken	19
MOCKDATA: Kaarten mocken	11
MOCKDATA: Mock of attackability of minion	21
MOCKDATA: Mock of Winning or Losing	23
MOCKDATA: Mock of Minions on playing field	14
MOCKDATA: Mock of Cards in hand	13
MOCKDATA: Mock of Random damage	11
MOCKDATA: TAUNT	18
MOCKDATA: Battlelog	17

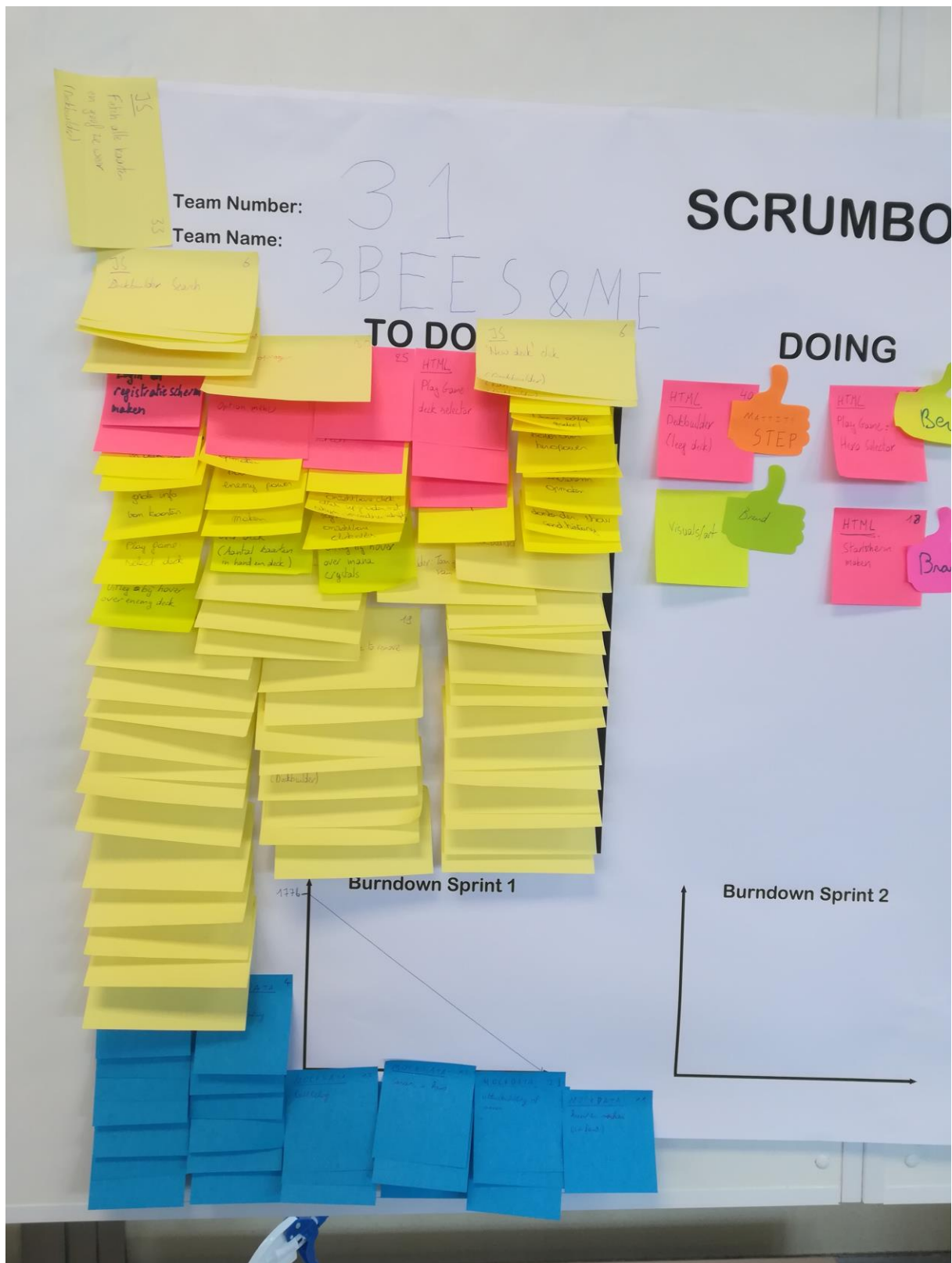
HTML: Deckbuilder (leeg deck)	40
CSS: leeg layout	45
CSS: Als er kaarten in deck zitten	25
CSS: grote info van kaarten	17
JS: fetch alle kaarten en geef ze weer	33
JS: back click (kan van overal komen ;) )	35

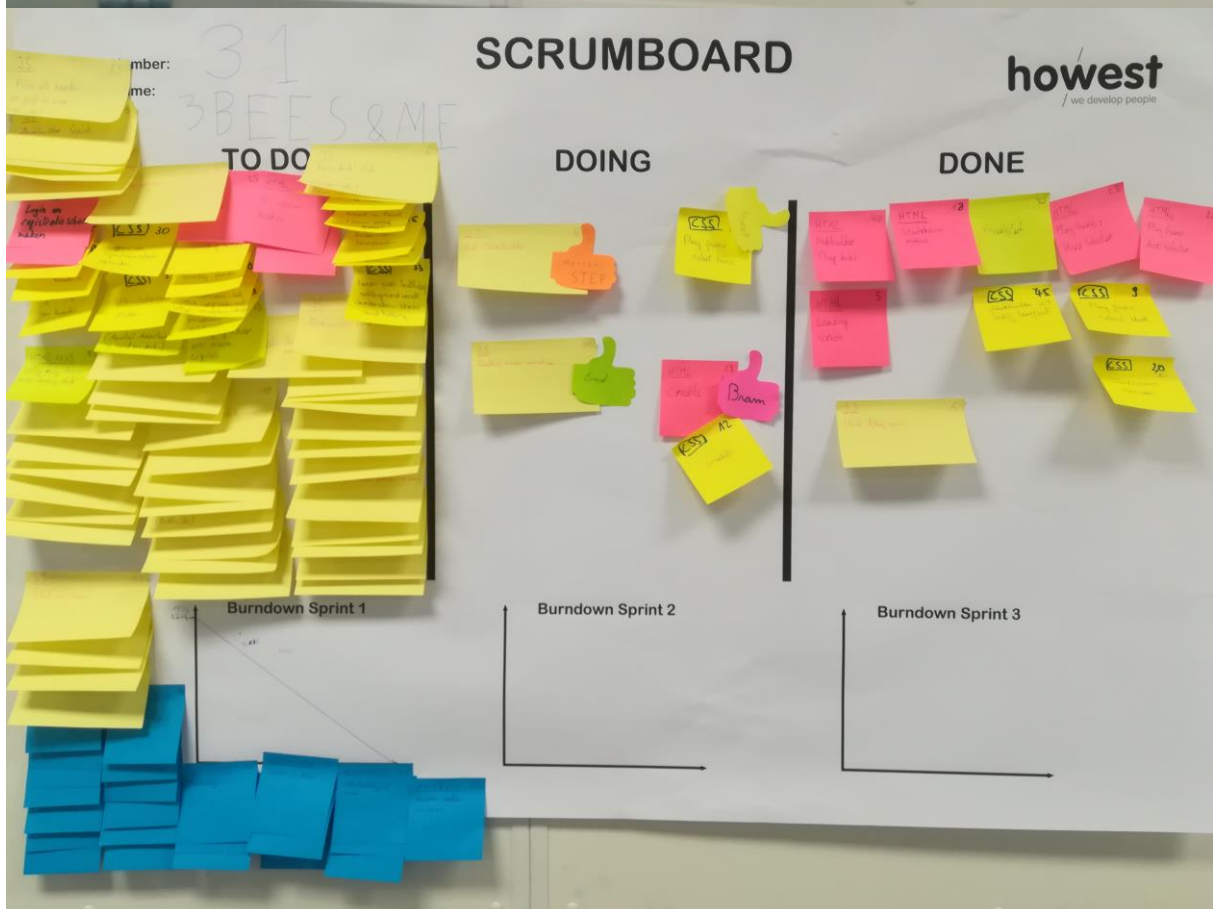
JS: 'New deck' click	6
JS: search	6
JS: sort all cards	6
JS: filter cards	6
JS: deck switching	6
JS: preview deck filters	6
JS: doubleclick to add	19
JS: doubleclick to remove	19
JS: drag to add	30
JS: drag to remove	30
JS: rightclick to view info	20
JS: Toon gekregen warnings van server	10
JS: Stuur save request naar server	6
JS: Scroll/pijltjes deck kaarten	25
MOCKDATA: cards	22
HTML: tutorial kaders	35
CSS: tutorial kaders	13
JS: tutorial kaders (click 'next' -> volgende kader)	29

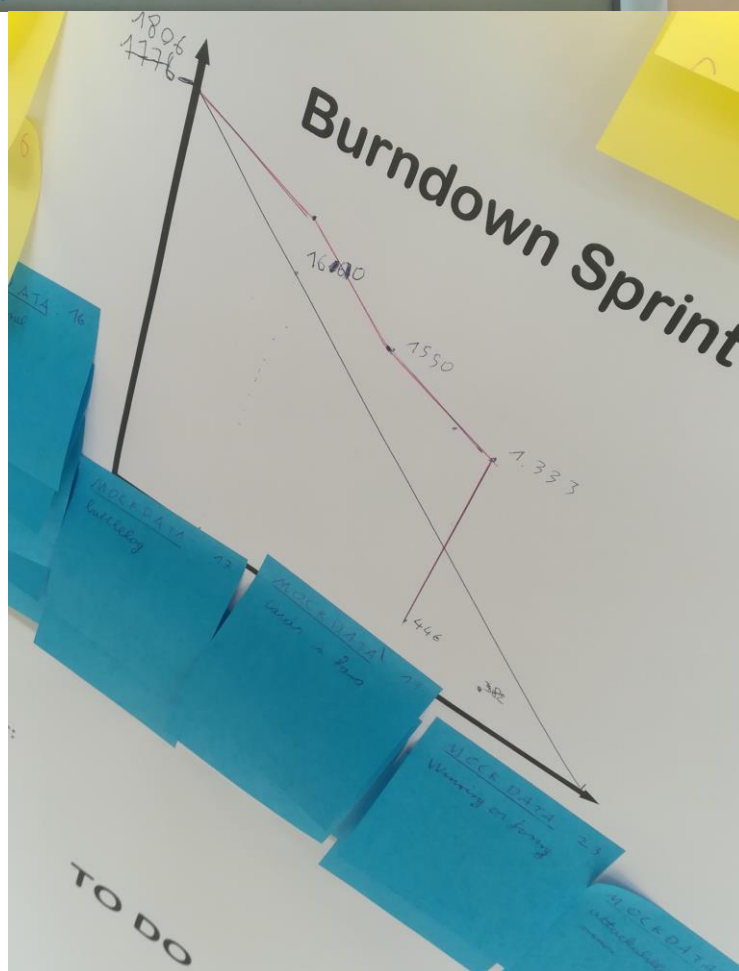
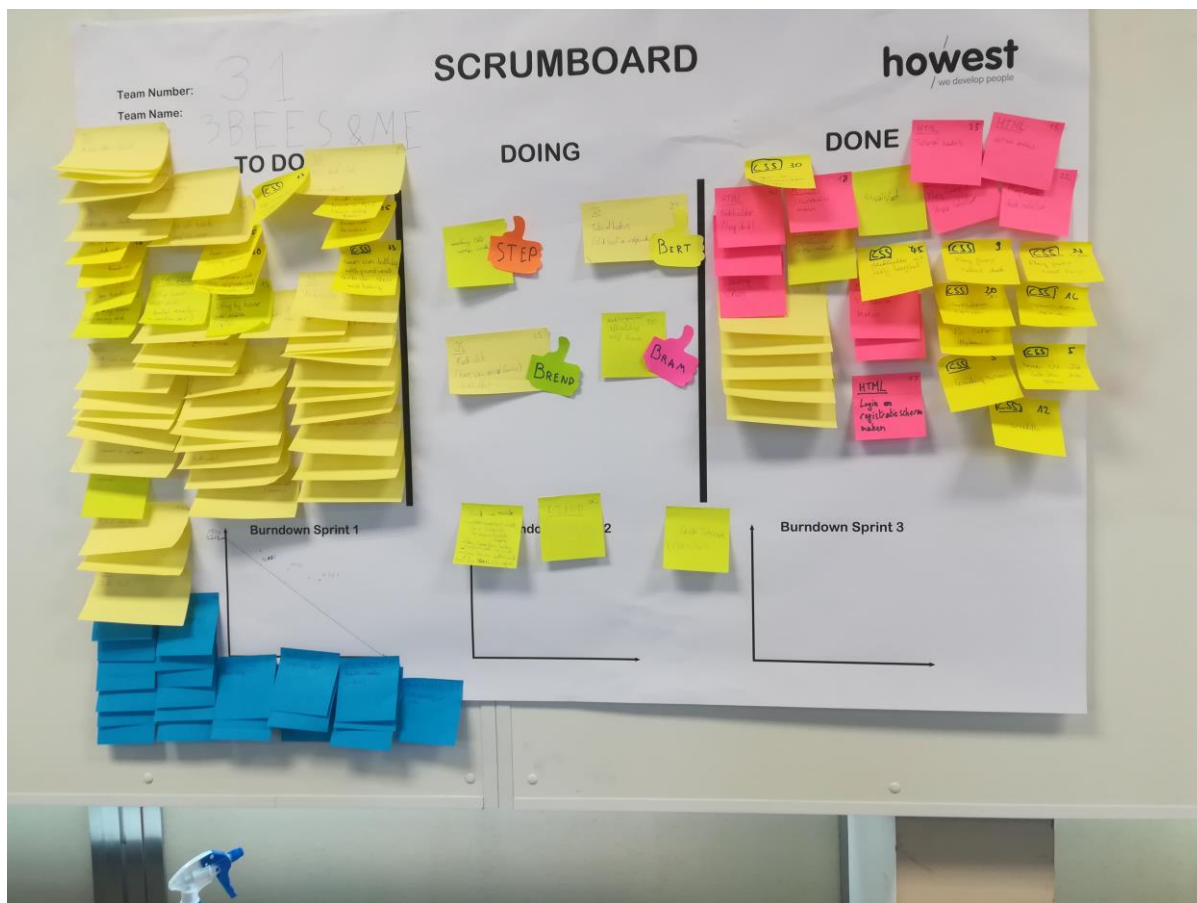
TOTAAL  
1776



### Foto's scrumbord







## *Sprint 01 retrospective*

Wat ging er goed?

We pasten de scrum methode consequent toe en hielden elke morgen een daily standup bij ons scrumbord. Het scrumboard hebben we aan de muur gehangen dicht bij onze werkplek in het D-gebouw. We werkten goed samen, door te pair programmeren en een vraag in de groep te gooien werden problemen snel opgelost en kon iedereen doordoen.

Wat kan er beter?

We hadden een moeizame start i.v.m. het gebruik van git. Dit was tegen de helft van de week opgelost door afspraken te maken en een extra development laag toe te voegen op de master.

Een tweede probleem was dat er te veel tijd besteed werd aan niet essentiële features zoals het perfectioneren van de css. Dit en het onderschatten van de hoeveelheid werk zorgden ervoor dat we helemaal niet klaar waren met alle user stories.

Wat gaan we anders doen?

In sprint 2 zullen we proberen alle front-end af te werken en zeker minder user stories op te nemen in de backlog. Daarnaast moeten we zo weinig mogelijk tijd steken in extra zaken en ons houden met wat moet gedaan worden.

## **Sprint 02**

### *Sprint backlog*

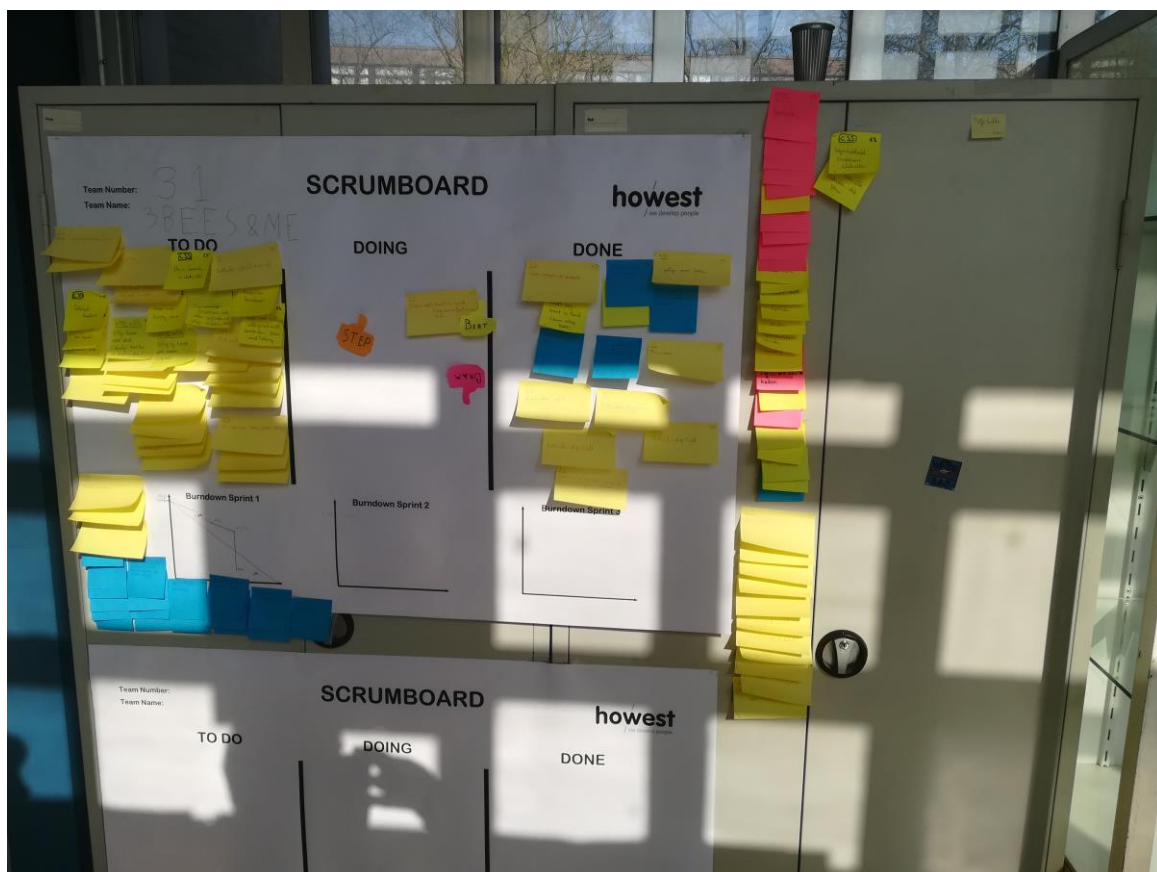
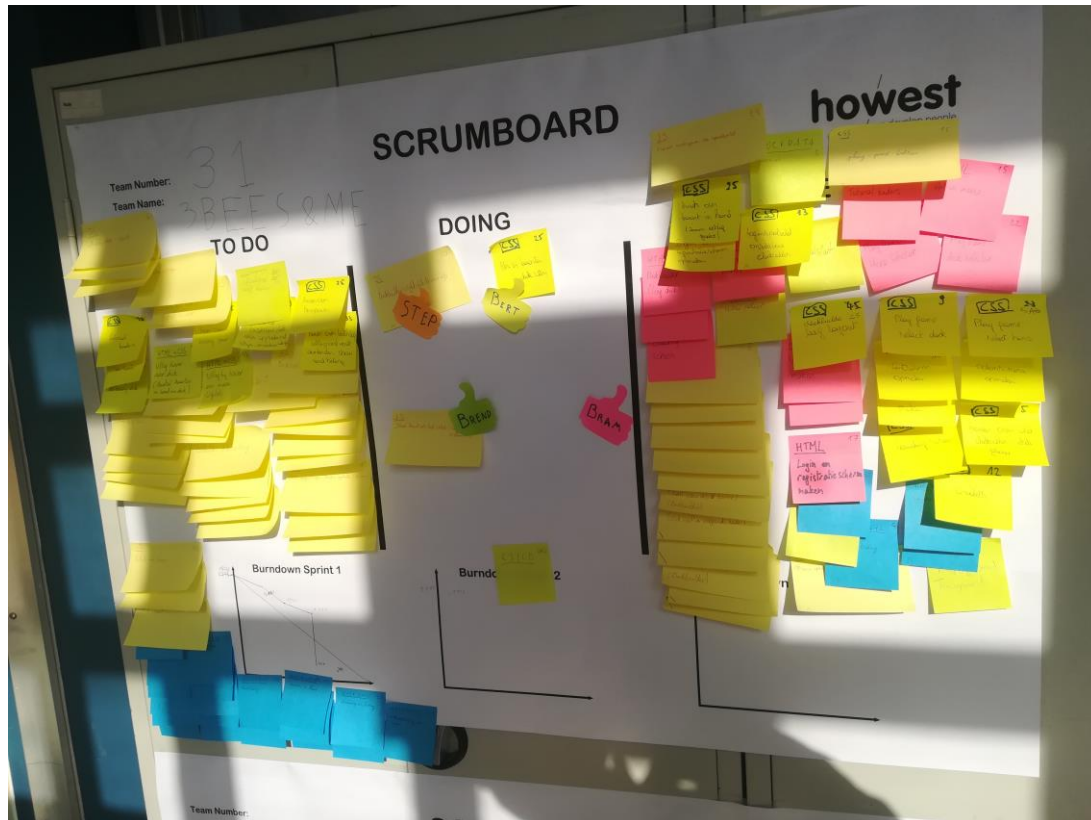
Omdat in sprint 01 maar iets meer dan de helft gedaan is, wordt de rest van de use cases overgeplaatst naar sprint 02. Daarnaast werden de user stories van de backend toegevoegd:

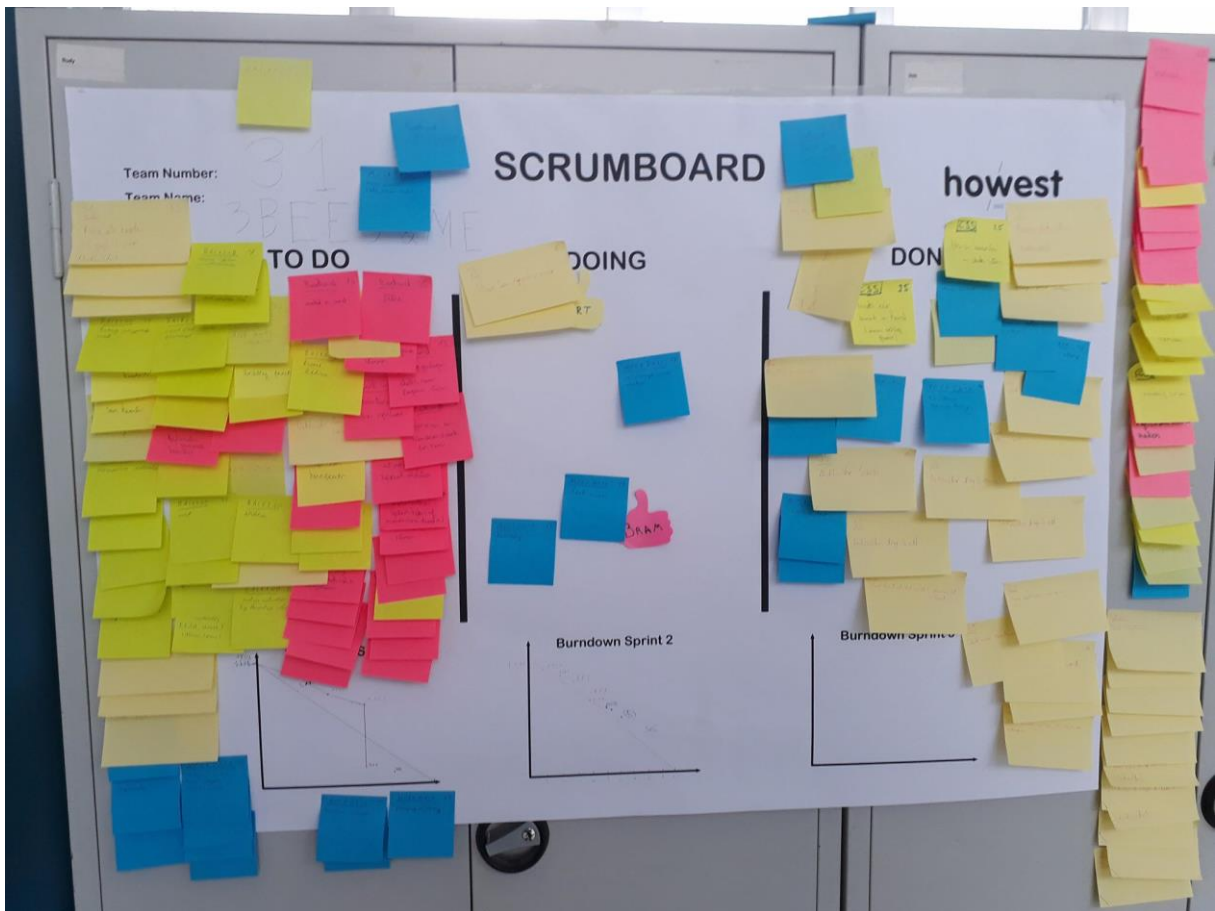
USER STORIES	GEWICHT
BACKEND: Javalin server klaarzetten	10
BACKEND: Alle heroes sturen naar pagina	17
BACKEND: Geselecteerde hero opslaan	10
BACKEND: Alle decks ophalen van MySQL database	26
BACKEND: Alle opgeslagen decks naar pagina sturen	15
BACKEND: Geselecteerd deck opslaan	10
BACKEND: Hero en Enemy naar pagina sturen	14
BACKEND: autoplay AI op End turn (request of time up)	45
BACKEND: Resterende tijd beurt naar pagina sturen	35
BACKEND: Val minion of hero aan + check voor TAUNTS (erna kijken of minion/ Hero dood is)	25
BACKEND: Random damage implementeren (tijdens kijken of minion/Hero dood is)	28
BACKEND: Random generate beginner	5
BACKEND: Uitvoeren van special abilities	33
BACKEND: Abilities	20
BACKEND: Battlecry	23
BACKEND: Charge	22

BACKEND: Stealth	23
BACKEND: Devine Shield	21
BACKEND: Deathrattle	24
BACKEND: Freeze	22
BACKEND: Taunt	21
BACKEND: Enrage	24
BACKEND: Windfury	21
BACKEND: Acties uitvoeren bij kaart op veld	45
BACKEND: Weapons implementeren	28
BACKEND: Heropowers implementeren	27
BACKEND: Battlefield implementeren (bijhouden gespeelde kaarten)	30
BACKEND: Mana systeem implementeren	18
BACKEND: Handkaarten bijhouden	14
BACKEND: Deckkaarten (krijg dit al van voor het spel start)	12
BACKEND: battlelog naar pagina sturen	15
BACKEND: Enemy autogenerate deck	24
BACKEND: Enemy: Update aantal kaarten in hand	13
BACKEND: Enemy: Update: hover over kaart, Speel kaart/...	50
BACKEND: Battlelog updaten	20
BACKEND: Alle kaarten aanpassen aan geselecteerde hero	35
BACKEND: Switch hero or deck: Stuur een warning wnr deck niet is opgeslaan	25
BACKEND: Save deck wanneer requested	30
BACKEND: add a card	13
BACKEND: deleting a card	13
BACKEND: filters	35
BACKEND: sort	35
BACKEND: search	30
BACKEND: give all cards	50
BACKEND: Genereer een standaarddeck per hero	34



Foto's scrumbord











## *Sprint 02 retrospective*

Wat ging er goed?

We hebben in lichte onderbreking een paar keer samengekomen om te kunnen werken op dezelfde manier als op school. In deze sprint maken we meer gebruik van gitlab om samen te werken zonder het scrumbord. Dankzij de cumulatieve inzet van het team kan er nu begonnen worden aan de back-end.

Wat kan er beter?

Tijdens deze sprint zijn er veel extra zaken toegevoegd aan de backlog die we voordien niet ingeplant hadden. Daardoor zijn we minder opgeschoten dan we gedacht hadden waardoor sprint 03 nu een heel hoge werkdruk krijgt.

Wat gaan we anders doen?

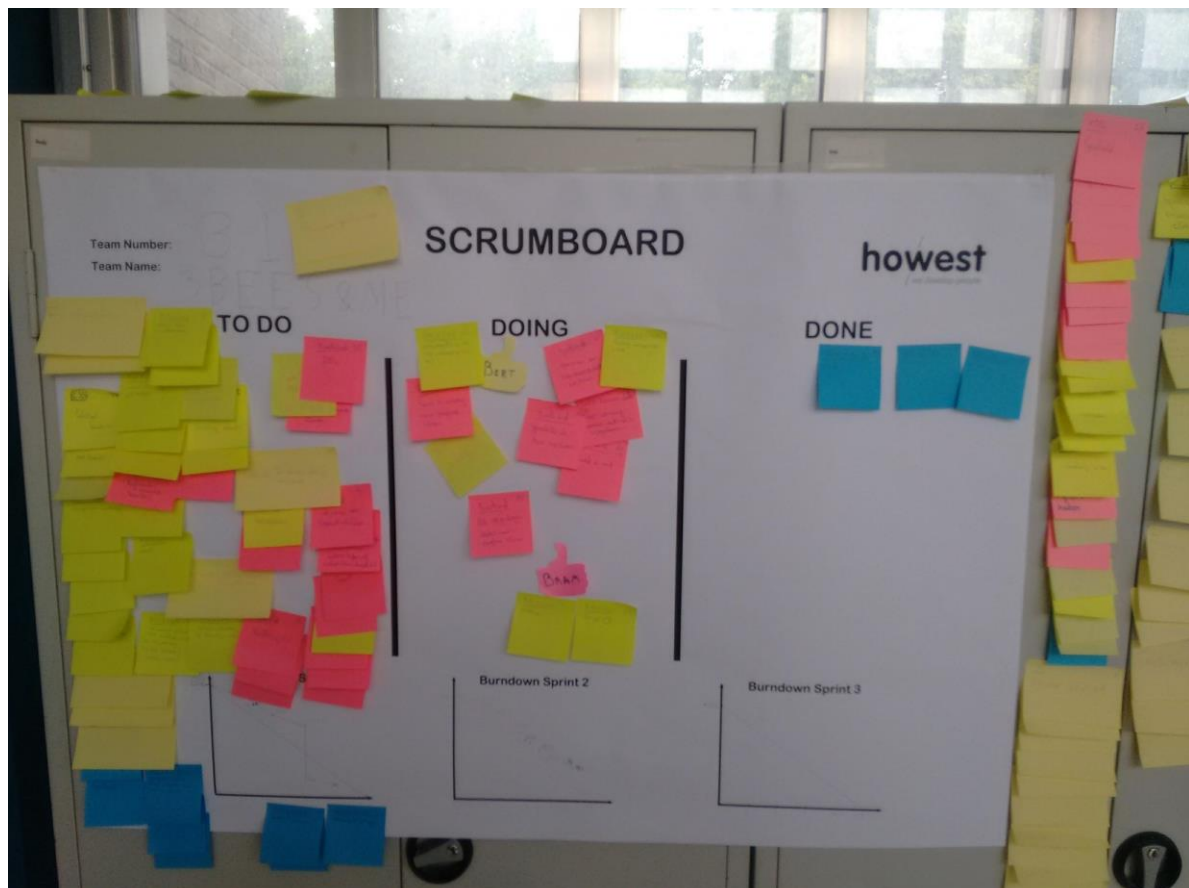
In sprint 03 zullen we ons concentreren op het laten werken van het spel want we willen het kunnen spelen, hoe simpel het ook is. We verwachten geen mirakel meer maar we willen zo veel mogelijk doen.

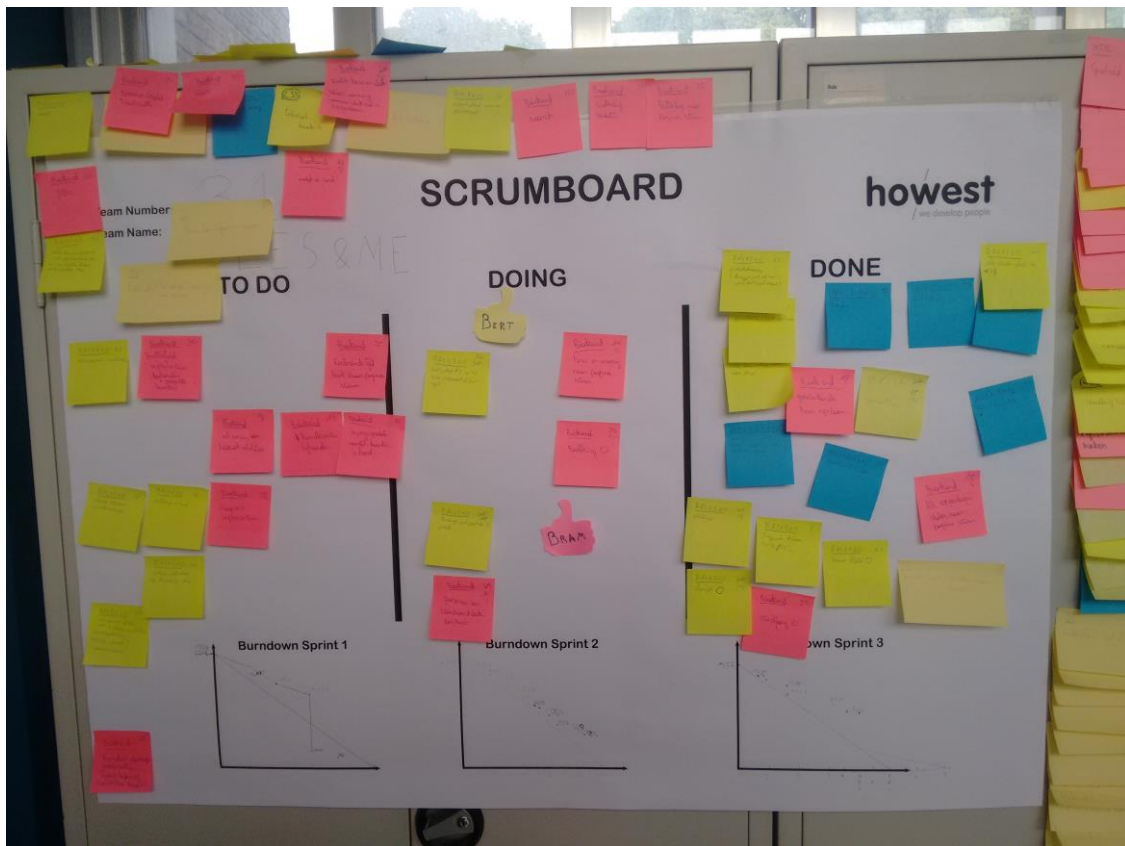
## **Sprint 03:**

### *Sprint backlog*

In sprint 03 werkten we verder met de user stories die nog niet af waren in sprint 02 en sprint 01.

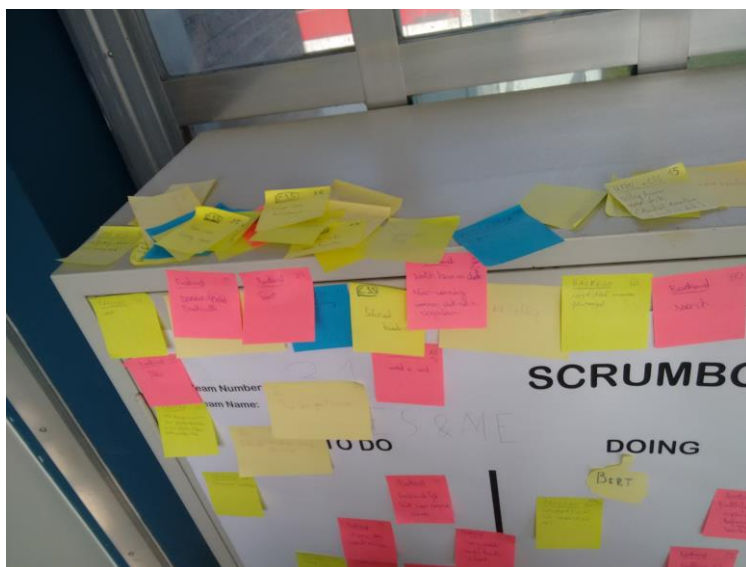
Foto's scrumbord











## *Sprint 03 retrospective*

Wat ging er goed?

Na een weekend van zware arbeid voor het vak Project Communication vlogen we er meteen terug in met dit project. Het spel werkte meer en meer en dit werkte heel motiverend. De omzet van het team was gigantisch in de laatste sprint en dit zorgde voor een grote opluchting op het eind van de laatste dag.

Wat kan er beter?

Deze week merkten we dat tijdens sprint een en twee niet iedereen evenveel aandacht besteed had aan de vooraf opgestelde vorm- en kwaliteitsvereisten voor de code. Dit allemaal controleren koste meer tijd dan we verwacht hadden.

Wat gaan we anders doen?

Een volgend project zullen we starten met een stuk extra ervaring. Met deze ervaring zullen we bepaalde zaken anders doen. De grootste verbeteringen die we denken te kunnen maken zijn een betere product backlog en de focus meer op de functionaliteit van de backend leggen. Het is belangrijker om een basisproduct af te hebben en er daarna extra features aan toe te voegen dan te ambitieus te zijn.