

Opdracht – Chess Island

Titel	Opdracht – Chess Island		
Vak	Programmeren Basis		
OPO Code	MGP27a	Auteur(s)	Johan Roosen
Versie	2.0	Review	Dries Decuyper
Aangemaakt op	25/06/2024	Laatste wijziging	03/07/2024



1 Introductie

Dit onderdeel telt mee voor de evaluatie. Dit is een **INDIVIDUELE** opdracht.

Onrechtmatig gebruik, of kopiëren, van code van andere niet toegelaten bronnen, zoals vermeld tijdens de lessen, wordt aanzien als fraude. Gebruik van AI tools is NIET toegelaten. Opgelet! Ook je eigen code (of stukjes ervan) doorgeven aan medestudenten wordt aanzien als fraude.

Het onderwerp van deze opdracht is het spel **Chess Island**.

2 Basis spelregels Chess Island

- Chess Island wordt gespeeld door twee spelers op een variatie van een schaakbord, waarbij het doel is om zoveel mogelijk lege vakjes op het bord te bezoeken met twee beschikbare schaakstukken (paard en koning).
- Elke speler beschikt over zowel een paard als een koning.
- Het bord bestaat standaard uit rijen en kolommen. Sommige vakjes zijn ingenomen door een rots en kunnen niet bezocht worden door de spelers.
- De schaakstukken van beide spelers beginnen op willekeurige posities op het bord.
- Spelers spelen om de beurt.
- Bij elke beurt, kiest een speler ofwel het paard ofwel de koning om te verplaatsen volgens de geldende regels van het klassieke schaakspel.
- Een stuk kan enkel verplaatst worden naar een leeg vakje. Een leeg vakje is een vakje waar geen ander schaakstuk, vlag of rots op staat.
- Bij het verlaten van een vak, wordt een vlag achtergelaten op het vak dat de speler verlaat. Hierdoor kan het vakje niet meer bezocht worden door beide spelers.
- Als zowel het paard en de koning van een speler niet meer kan bewegen, eindigt het spel voor die speler.
- Het spel gaat door tot beide spelers geen geldige zetten meer hebben voor hun stukken. De speler die aan het einde van het spel de meeste vlaggen heeft geplaatst, wint.

3 Opdracht

3.1 Doel

In deze opdracht zal je een C# console toepassing maken waarbij 2 spelers het spel tegen elkaar kunnen spelen. De functionele vereisten zijn opgesplitst in fase 1, 2 en 3.

- Fase 1 – Basis functionaliteit
- Fase 2 – Gameplay
- Fase 3 – Uitbreidingen en spelvariaties

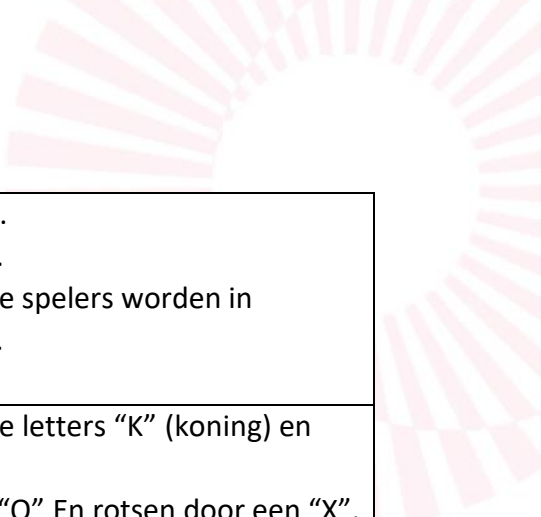
Het is de bedoeling om alle vereisten te programmeren volgens de 'best practices', van commentaar te voorzien en uitgebreid te testen. Te beginnen bij de vereisten van fase 1 en vervolgens fase 2 en 3.

3.2 Technische vereisten

- Solution- en projectnaam = **PRB99.ASN.ChessIsland**.
- Het spel moet volledig volgens een Object-Oriented Programming aanpak geprogrammeerd worden. De blauwdruk van het class diagram zit achteraan dit document.
- Enkel technieken die we in de cursus Programmeren Basis (theorie of oefeningen) gezien hebben zijn toegelaten. Andere technieken kunnen gebruikt worden op voorwaarde dat:
 - De student deze 100% begrijpt, correct toepast en kan uitleggen.
 - De student de bron vermeldt, waar deze techniek werd gevonden.
- Maak gebruik van voldoende commentaar om je code uit te leggen. Voor functies is het noodzakelijk om het gangbare commentaar formaat te gebruiken.
- Voorzie bovenaan in de Main() functie ook informatie in commentaar met je naam, datum en korte beschrijving van de applicatie.
- Volg strikt ALLE richtlijnen en *best practices* zoals aangeleerd tijdens de lessen (naamgevingconventies, taalkeuze, ...).
- Doe aan duidelijke bronvermelding (commentaar) als je iets op internet opgezocht en gevonden hebt.

4 Fase 1 – Basis functionaliteiten

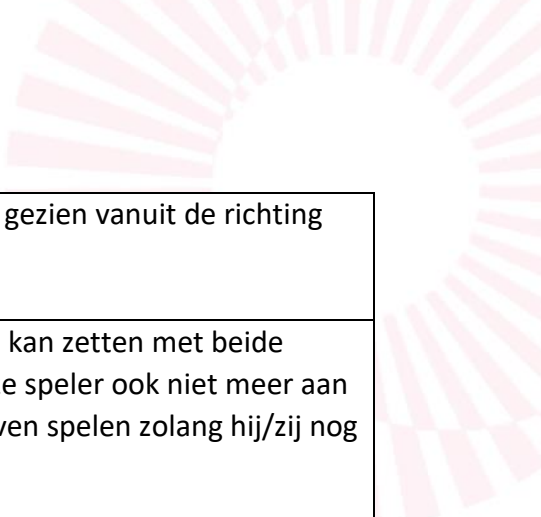
ID	Onderdeel	Functionele vereiste
1.1.1	Opstarten	De toepassing wordt opgestart als Console applicatie, via de “Run” optie vanuit het Visual Studio project.
1.1.2	Opstarten	De voornaam van beide spelers wordt opgevraagd. De eerste speler begint het spel.
1.2.1	Opzet	Bij de opstart toont de toepassing een lijst van alle beschikbare *.map bestanden uit de applicatie folder. Na het kiezen van een map, wordt deze map ingeladen.
1.2.2	Opzet	De positie van de rotsen op het bord worden ingeladen vanuit een tekstbestand “IslandMap.map”. De rotsen worden op de juiste locatie getekend. Het formaat waarin de info in dit tekstbestand wordt opgeslagen is als volgt: <ul style="list-style-type: none">- Elke rij van het bord wordt voorgesteld door één rij in het bestand.- Een lege cel wordt aangeduid met een “.”, een rots met een “X”. Voorbeeld:X....XX.....X..X
1.2.3	Opzet	Het spelbord (ingeladen uit de map) wordt getekend.
1.2.4	Opzet	Een map kan een willekeurige hoogte en breedte hebben. <ul style="list-style-type: none">- Maximum hoogte = 15- Maximum breedte = 30 Mappen van een te groot formaat resulteren in een foutboodschap.
1.2.5	Opzet	De koning en het paard van beide spelers worden op een willekeurige positie (vrij vak) op het bord geplaatst.
1.2.6	Opzet	Het scorebord wordt gevisualiseerd met daarop de volgende info: <ul style="list-style-type: none">- Welke speler is aan de beurt.- Hoeveel vlaggen elke speler reeds heeft geplant.



1.3.1	GUI	<p>Het spel gebruikt verschillende kleuren.</p> <ul style="list-style-type: none"> - Het bord en rotsen in één kleur. - De stukken en vlaggen van beide spelers worden in verschillende kleuren getekend. -
1.3.2	GUI	<p>De stukken worden voorgesteld door de letters “K” (koning) en “P” (paard).</p> <p>Vlaggen worden voorgesteld door een “O” En rotsen door een “X”.</p>
1.3.3	GUI	De volledige interface is in het Nederlands.
1.3.4	GUI	De coördinaten van het bord worden horizontaal aangeduid met een letter (A, B, C, ...) en verticaal met een cijfer (1, 2, 3, 4, ...).

5 Fase 2 – Gameplay

ID	Onderdeel	Functionele vereiste
2.3.1	GUI	Alle mogelijke acties worden aangeboden via een eenvoudig menu systeem, waarbij elke speler met een minimaal aantal toetsaanslagen zijn/haar keuze kan maken.
2.3.2	GUI	Als de beurt wisselt van de ene speler naar de andere, wordt het bord telkens opnieuw getekend op een LEGE console. De console mag dus niet verder scrollen naarmate er steeds meer inhoud op de console verschijnt. Let ook op dat boodschappen op het scherm niet te snel verdwijnen als gevolg hiervan.
2.4.1	Spelverloop	De eerste speler begint het spel. Spelers krijgen één zet per beurt waarna de beurt naar de andere speler gaat.
2.4.2	Spelverloop	De speler die aan de beurt is wordt gevraagd om eerst te kiezen welk stuk hij/zij wil verplaatsen. En vervolgens naar welk vak. De opties waarnaar het stuk zich kan verplaatsen, worden in een keuzemenu gegeven. De speler kan er hieruit één kiezen.
2.4.3	Spelverloop	Een speler kan enkel een beweging maken naar een vak dat aangeboden wordt uit het keuzemenu. Andere opties worden niet aangeboden via het menu.
2.4.4	Spelverloop	Als een speler een vak moet kiezen, maar een “ongeldige” keuze maakt, wordt de keuze van het stuk zelf, opnieuw ongedaan gemaakt. Dit is een mogelijkheid voor de speler om, indien hij/zij zich realiseert toch het andere stuk te willen gebruiken, terug te keren.
2.4.5	Spelverloop	Bij het verlaten van een vak, plant de speler een vlag op het vak dat hij/zij verlaat. De vlag wordt dus pas zichtbaar van zodra de speler het vak verlaat.
2.5.5	Spelverloop	Voor mogelijke bewegingen worden de regels uit het klassieke schaakspel gebruikt. <ul style="list-style-type: none">- Koning kan één vakje bewegen in elke richting (horizontaal, verticaal, diagonaal).- Paard kan een sprong maken: één stap voorwaarts (horizontaal of verticaal) en vervolgens één stap diagonaal



		voorwaarts naar links of rechts, gezien vanuit de richting vanwaar hij kwam.
2.5.6	Spelverloop	Indien een speler geen geldige stappen kan zetten met beide stukken, wordt dit gemeld en komt deze speler ook niet meer aan de beurt. De tegenstander kan dan blijven spelen zolang hij/zij nog geldige zetten heeft.
2.5.7	Spelverloop	Een stuk kan niet buiten de randen van het bord springen.
2.6.1	Einde	Als beide spelers geen geldige zet meer kunnen plaatsen, eindigt het spel.
2.6.2	Einde	Als het spel beëindigd wordt, wordt de naam van de winnaar in een opvallende kader getoond met als tekst "WINNAAR = ... ". Waarbij op de "..." de naam van de winnaar getoond wordt.
2.6.3	Einde	Indien er geen winnaar is, wordt de tekst getoond "GELIJK SPEL".

6 Fase 3 - Uitbreidingen en spelvarianties

Er zijn tal van mogelijke uitbreidingen en spelvarianties mogelijk voor dit spel.

Zowel op vlak van GUI als spelverloop/spelregels.

Voorbeelden:

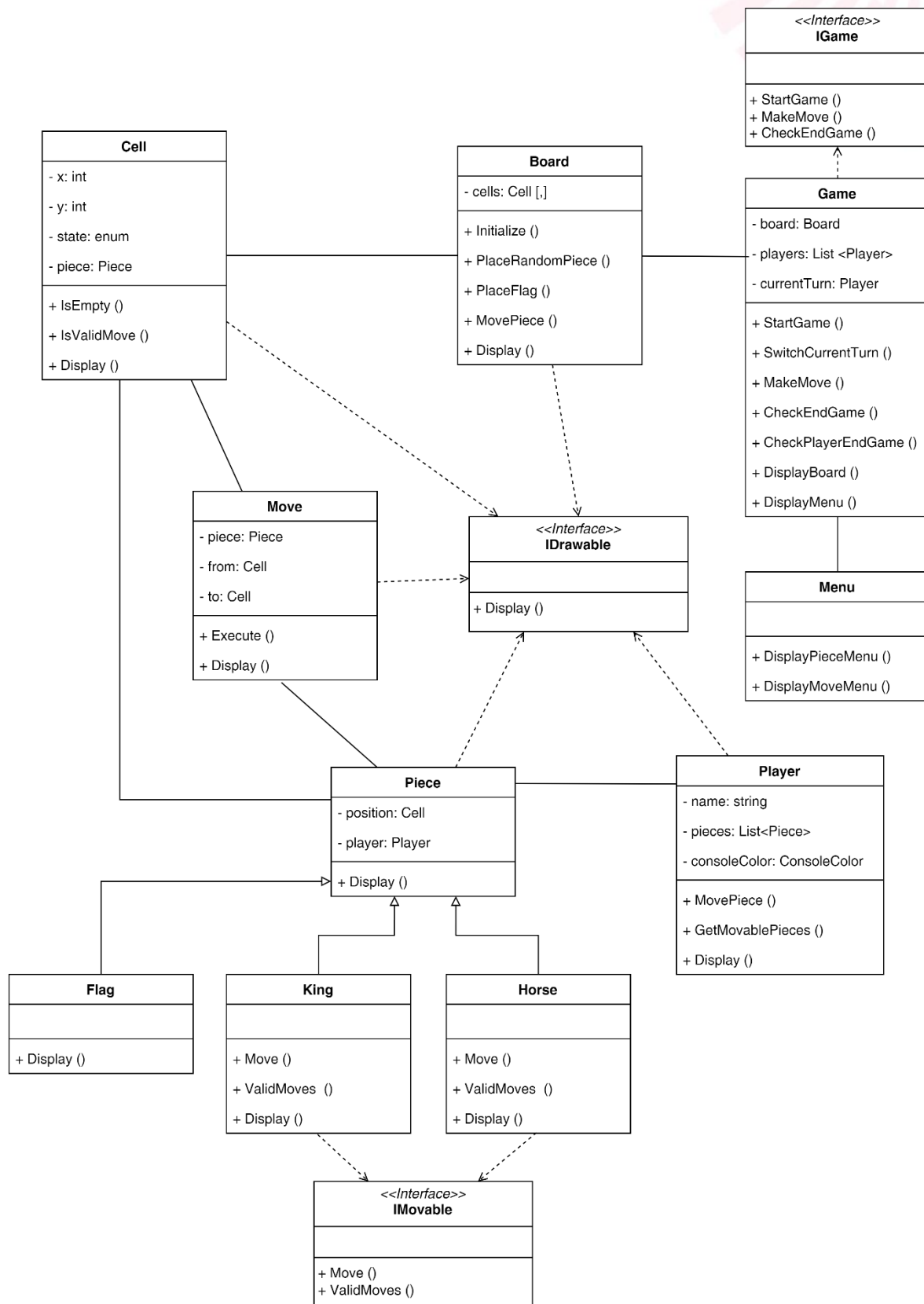
Onderdeel	Functionele vereiste
GUI	De symbolen van de koning en paard worden voorgesteld door een afbeelding van een schaakstuk.
Spelverloop	Bij de opstart, kan er gekozen worden om de schaakstukken willekeurig te plaatsen ofwel dat elke speler om beurt zijn/haar stukken zelf mag plaatsen op de gevisualiseerde map volgens het volgende stappenplan: <ul style="list-style-type: none">- Speler 1 plaatst zijn/haar koning- Speler 2 plaatst zijn/haar koning- Speler 1 plaatst zijn/haar paard- Speler 2 plaatst zijn/haar paard
Spelregel	Elke speler heeft één keer de mogelijkheid om op een vak te springen waar een vlag staat van de tegenpartij. De vlag van de tegenpartij wordt hierbij verwijderd en vervangen. Of de speler deze optie al gebruikt heeft of niet, wordt aangeduid in het scorebord.

Verzin zelf één spelvariatie naar keuze, om te implementeren.

Opgelet! Deze moet verschillend zijn van de voorbeelden die hierboven vermeld zijn. De voorbeelden dienen enkel als inspiratie.

Beschrijf je variatie duidelijk in commentaar.

7 Class diagram



Bijkomende tips class diagram:

- Het class diagram moet worden beschouwd als een **blauwdruk**. Dit wil zeggen dat indien gewenst nieuwe klassen of interfaces kunnen worden aangemaakt. Hoewel dit niet noodzakelijk is.
- De functies in het class diagram bevatten geen signatuur. Dat wil zeggen dat input parameters en resultaten van functies niet weergegeven zijn. Deze moeten (eventueel met de programmeertips hieronder) zelf uitgewerkt worden.
- Private functies en data members kunnen uiteraard worden toegevoegd om de code leesbaar en onderhoudsvriendelijk te maken. Dit is uiteraard wel aangewezen.

Programmeertips

- Voor het gebruik van kleuren in de GUI maak je best gebruik van een library genaamd Pastel. Hoe je deze kan gebruiken, vind je terug via <https://github.com/silkfire/Pastel>.
- In de Game klasse
 - De MakeMove() functionaliteit wordt uitgevoerd op basis van het door de speler gekozen schaakstuk, en een move van het desbetreffende schaakstuk.
 - De DisplayMenu() functionaliteit moet onderscheid kunnen maken tussen meerdere menu's, om de juiste weer te geven. Een menu kan ook gebaseerd zijn op de input van een speler.
- In de Board klasse
 - De PlaceFlag() functionaliteit moet niet enkel een vlag aanmaken met al zijn gegevens, maar ook de gegevens van de cel invullen en toekennen aan de stukken van de speler.
- In de Move klasse
 - Voor de implementatie van de Display() functionaliteit, kan de ASCII tabel nuttig zijn.
- In de Player klasse
 - De GetMovablePieces() geeft enkel een lijst terug van de schaakstukken (van de desbetreffende speler) die kunnen bewegen.

8 Evaluatie

De behaalde score wordt toegekend op basis van de evaluatie van meerdere criteria. Het correct werken van alle functionaliteiten is dus niet voldoende om een maximum score te behalen!

Zie ook de evaluatie criteria op Toledo.