

Martin Arvidsson
UD 14
PhP kompletteringsuppsats
2015-11-19

Inledning:

Det här är min kompletteringsuppsats i PHP, Jag kommer att gå igenom följande punkter i uppsatsen.

- Vad mitt projekt är, förklara funktionaliteten med vad jag har skapat.
- Kraven som är ställda på projektet, kommer att gå igenom varje punkt och förklara relativt kortfattat hur jag lyckas uppfylla just det kravet.
- Mitt klassdiagram, varför det ser ut som det gör samt förklara hur applikationen fungerar rent kodmässigt.
- Gå igenom mina testfall och min testmetod
- Gå igenom mina användningsfall för applikationen
- Gå igenom lite om säkerhetsaspekterna i min applikation
- Förklara kortfattat om hur jag uppfyller den kodkvaliten som jag eftersträvade.

Projektets funktionalitet:

Mitt projekt som jag har skapat är en webbaserad lokal version av luffarshack, där man kan spela i två olika lägen, Antingen först till 3 vinster på ett 3 x 3 bräde, eller först till 5 vinster på ett 5x5 bräde, Spelarna turas om att trycka på varsin bricka för att göra ett drag, detta går fram och tillbaka tills någon antingen har vunnit eller hela brädet har fyllts utan att en vinnare har hittats.

När en vinnare för en match väl har hittats får man se det vinnande draget samt en knapp för att gå till nästa match i serien, när alla matcher väl är spelade tas man tillbaka till huvudmenyn där antal vinster för del spelaren som vann har ökat i det gamemodet som spelades, detta scoresystem är mest för att man ska se lite statistik över vem som vinner flest gånger.

Utöver detta har man också en möjlighet att avbryta matchen då den pågår genom att trycka på cancelknappen, detta avbryter spelserien som pågår och man tas tillbaka till startmenyn igen.

Kraven ställda på projektet:

Kraven som ställdes på projektet var dom på nästa sida, Jag kommer under varje krav beskriva hur jag anser att jag uppfyller det kravet.

Minimum krav:

- Uppgiften är inlämnad innan deadline.
Uppgiften lämnades in innan deadline, Nu är detta en kompletteringsuppsats som förhoppningsvis hinner in i tid.
- koden ska finnas tillgänglig på ett publikt repo.
koden är tillgänglig i en release här:
<https://github.com/MartinArvidsson/PHP-project-3x3-5x5/releases/tag/v1>.
- Applikationen ska finnas tillgänglig på en publik webbsida.
Mitt projekt är tillgängligt här:
<http://martin-viktor-arvidsson.se/PHP-project-3x3-5x5-master/Project/Index.php>
- Uppgiften är gjord individuellt.
Jag har jobbat ensam på projektet , brainstormat med andra kursmedlemmar om olika lösningar men annars ingenting.
- Ingen kod är kopierad, man har arbetat ensam
Se ovanstående
- Inga bibliotek och utomstående kod har använts
Ingen extern kod har använts.
- Ingen javascript har använts
Ingen javascript har använts.

- HTML följer webbstandarden

Försökt att följa standarden så bra som möjligt, validerade sidan och fick bort dom flesta felen som påpekades, en del fel kvarstod som jag inte lyckades lösa.

- Applikationen har en sida(Index.php)

Använder samma princip som på laborationerna, med att jag har en index sida som genererar HTML med hjälp av en layoutview, och det som står i layoutviewn bestämmer mastercontrollern beroende på vilken URL man är på.

Krav på projektet

- Kraven är beskrivna i visionen samt i användningsfallen

Mina testfall samt användningsfall är relativt kortfattat beskrivna i mitt repo på github i readme filen,

Användningsfallen som jag har beskriver dom olika funktionerna i applikationen, ex man ska navigera till sidan och kunna starta en 3x3 match, eller man ska kunna avbryta en match, eller att man ska kunna vinna på alla olika mönster.

- Projektet har tester och lyckas uppfylla testkraven

När det gäller testfallen är dom uppbyggda på relativt samma sätt, jag bygger vidare på testfallen och säger att man ska t.ex. inte kunna placera en bricka där någon annan redan har gjort det, testmetoden som jag har använt mig av under hela projektets gång har bara varit manuella tester då jag inte haft något bra alternativ, utan för att testa spellogiken har jag fått vinna mot mig själv på alla dom 8 olika mönster(3 lodrätt, 3 horisontellt, 2 diagonalt) som fanns i spelet.

- Applikationen är väl testad utöver testfallen

Jag har suttit och testat varje funktion manuellt allt eftersom jag implementerade dom, sedan gjordes en del testkörningar på hela applikationen innan den lades upp på den publika servern.

- Säkerhetsaspekten är igenomtänk och väl hanterad

Det finns inte jättemycket säkerhetsaspekter att kolla på i applikationen då jag inte har några inputfields, den enda säkerhetsaspekten som jag tror ev. skulle kunna manipuleras är dom olika sessionerna som håller koll på personens highscore, om man på något sätt skulle få den att automatiskt vinna en serie när den startas om det faktiskt går, jag har inte testat för detta.

Code Quality

- Kodens MÅSTE vara objekt orienterad.

Jag tror att jag har följt stukturen som vi har lärt oss i OOAD kursen (Object oriented analysis and design) relativt bra,
- Kodens ser bra ut
 - Bra förklarande namngivning

Försöker namnge mina funktioner , variabler m.m. på ett sådant sätt så att dom beskriver vad dom gör ex. "PlayerMovesMade" eller en funktion som heter "DoesUserWantToMove".
 - Kodens är väl kommenterad / Dokumenterad

överlag försöker jag att ev. göra en kommentar innan en funktion / variabel som inte är självklar vad den gör eller relativt komplicerad, målet med kommentarerna är att man ska förstå programmets funktionalitet enbart genom att man läser kommentarerna brevid funktionerna
 - Undvik arrayer när man arbetar med något som är en klass.

Tror inte jag använder arrayer vid andra tillfällen än när jag skriver ut spelplanen vilket görs genom en multidimensionell array för att lösa problemet med att göra kommentarsystemet
 - Kodens har ordentlig indentering

Den ska vara väl indelad i grupper om lagom storlek, en funktion i taget och indenteringen ska följas.
- Felmeddelanden hanteras på ett bra sätt (Undantag etc.)

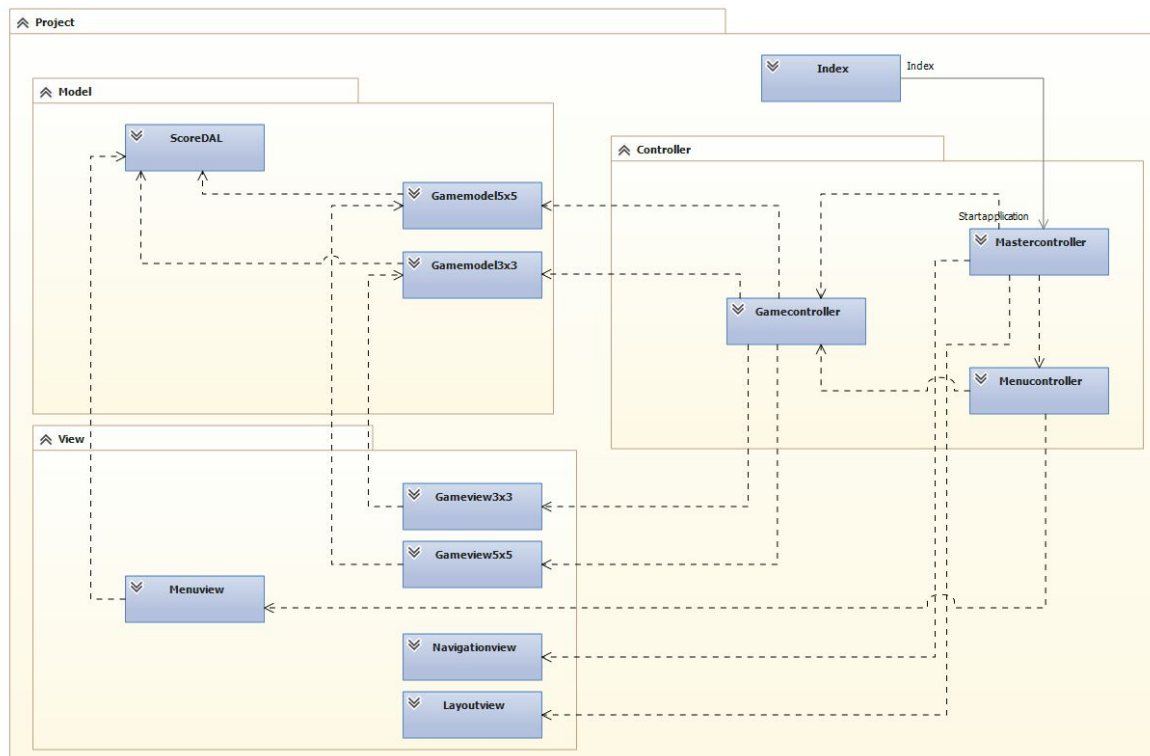
Det finns inga sätt att kasta undantag på då det inte finns någonstans i programmet där det ska kunna kastas undantag så som jag ser i dagsläget.

- Kodens repositorie är väl presenterat på git (eg. README.md)
 - Ja, Min readme förklarar vad applikationen är, vilka usecases jag haft samt vilka testcases som jag har gjort.
 - Beroenden är beskrivna och förklarade
 - Programmet har inga utomstående beroenden
 - Installationsinstruktioner
 - Ingen utomstående installation krävs heller.
- Beroenden är inkapslade eller hanterade
 - Inga strängberoenden, använd strategier för att undvika dom
 - Tror inte jag har några stringdependencies mellan dom olika filerna, dom sessionerna som finns är bara i den filen/klassen.
- Undvik onödig komplexitet
 - Tror inte jag har någon komplexitet, hade lite innan den här kompletteringen bl.a. kollade jag brädet efter vinnande möster rad för rad vilket gav 8 rader, loopar igenom nu istället vilket minskade till 3 rader istället.

MVC implementation

- Koderna är uppdelade i Model/VY/Kontrollerna
 - Ja, Jag har följt MVC strukturen under arbetets gång
- Inga beroenden från Model till Kontroller eller Vyn
 - Inga beroende, kontrollerna hämtar från vyn och skickar till modellen, vyn läser också från modellen.
- Inga beroenden från vyn till kontrollerna
 - Vyn har inga beroende till kontrollerna
- använd enbart \$_GET, \$_POST, \$_COOKIES etc i Vy-klasserna
 - Har inga get på andra ställen än i vyerna
- Ingen output generation i Model eller Kontroller
 - Vyn sköter all HTML generering.
- Ingen Vy responsibilitet i Model eller Kontroller
 - Ingen \$_GET, HTML generering eller input finns någon annanstans än i vyerna.
- Ingen Model responsibilitet i Vy eller Kontroller
 - ingen "komplicerad" validering av data eller nersparande av data finns någon annanstans än i vyn
- Ingen kontroller responsibilitet i Vyn eller Modellen
 - Kontrollerna dirigerar uppdrag och ingen annanstans.

Class Diagram



Har enbart en association och den är mellan Index och Mastercontrollern, det är en association för att den inte ligger i ett fält / funktion. Resterande relationer är anropade i funktioner, vilket leder till dependency då funktionen inte fungerar utan den relationen.

Index startar mastercontrollern, som i sin tur kollar vad är URL:en med hjälp av Navigationview, står det inte Game3x3 eller Game5x5 så gå till menucontroller, annars pågår ett spel och isåfall gå till gamecontroller som sköter spelet efter det.

Menucontrollern har två knappar för att spela 3x3 eller 5x5, detta skickas till Gamecontroller som sedan startar spelet med hjälp av att köra den motsvarande vyn och modellen tills en spelare får 3 eller 5 poäng beroende på vilken match man valde, detta skickas sedan till scoredal som uppdaterar highscoren för den matchen och spelaren med +1, när det är klart skrivs menuviewn ut igen och hela loopen börjar om.

Testfall/Testmetod

Mina testfall har varit tester som kollat funktionaliteten på sidan, att dom olika knapparna fungerar m.m. och gör det dom ska (3x3 knappen ska inte starta 5x5).

Testfallen som jag har är inte särskilt stora då dom enbart oftast testar en liten funktion vardera då jag saknar ställen för input i applikationen är varje test mer ett "Gör knappen det den ska? Ja/Nej"

Testmetoden som jag har använt mig av har varit manuella tester allt eftersom fler funktioner blev tillagda i applikationen.

ex. när jag lade till 5x5 spelplanen testade jag först att vinna diagonalt åt båda hållen, efter det började jag att testa horisontellt och efter det vertikalt.

samma princip använde jag på 3x3 spelplanen samt när poängen lades till, då testade jag det genom att vinna en först till 3 vinster serie som ex. spelare "O" och när jag kom tillbaka till start skulle poängen ha uppdaterats med 1 poäng och om det inträffade kunde jag bocka av det testet då funktionen gjorde det som den skulle.

Exempel på testfall:

Testcase	Requirement	Goal	Method of testing	Status
Datahandling	After winning a series the score on the startpage should reflect that a player won and update accordingly	Updated score on startpage	Start a game of 3x3, finish the entire game, win 3 times as player X or O	Completed, Score updated accordingly to winner of series

Användningsfall

Mina användningsfall har varit det som användaren kommer att göra i applikationen (Placera brickor, navigera runt etc..) Jag började med det som kommer först och arbetade mig igenom applikationen, Man ska kunna navigera till sidan, man ska kunna starta en match, man ska kunna placera bricka m.m

Exempel på Användningsfall:

1. Able to start a new game of tic-tac-toe 3x3 or 5x5
2. Able to play the version of tic-tac-toe locally between two players.

Säkerhetsaspekter i applikationen

Det finns inte särskilt mycket säkerhetsaspekter i min applikation att diskutera enligt mig, Jag skapade ett spel där det enbart finns knappar, det finns ingenstans där man kan mata in data så jag behöver inte oroa mig om att man ska ev. skicka in ex. HTML taggar i applikationen.

Det som ev. skulle kunna vara en "säkerhetsrisk" är att poängdatabasen går att nå och ladda ner genom att ändra i URL:en till sökvägen för databasen, men eftersom jag inte lagrar någon användardata är utan bara nummer ser jag inte det som en jättestor risk.

Kodkvalitet

Jag har försökt att hålla en hög standard på min kodkvalitet överlag, korrekt indentering samt att jag har skrivit kommentarer runt funktionerna i applikationen så att man förstår vad funktionen gör utan att behöva läsa igenom hela funktionen, min namngivning tror jag är relativt bra eftersom namnen är ganska bra beskrivande på vad funktionen gör eller vad variablen håller för slags information ex. (int PlayerOWins , function getwhowonmessage()..).
utöver detta ska det inte finnas död kod i applikationen från tidigare idéer och försök.