

PSPy-11

projektbeskrivning

Villkor och tider

Start:	måndag 15:e maj
Projektdefinition:	<i>senast</i> fredag 19:e maj
Projektredovisning:	26: maj
Kodvillkor:	Python, git, ett "nytt" Python-bibliotek.
Genomförande:	Individuellt, eller grupparbete två personer OM: <ol style="list-style-type: none">1. ni befinner er tillgängliga på nätet under projekttiden2. ni genomför en daily scrum under normala veckoarbetsdagar, och3. ni delar upp arbetet så att ni blir ansvariga för två separata delar Ni får byta idéer med varandra helt fritt, och hämta kod från nätet, men ni ska ange källan.
Tidsanvändning:	v20: förstudier och projektdefinition, experimentell programmering v21: programmering och projektredovisning
Betygsnivåer:	G ni har lämnat in en länk till <i>något</i> , som funkar i huvudsak, men saknar viktiga saker som git-historik, projektdefinition, inte har rätt referenser och så vidare, vilket gör det svårt för läraren att förstå sig på ditt utvecklingsarbete; VG ni har lämnat in git-länk <i>med historik</i> , har referenser så att läraren kan se vilken kod ni har utvecklat ifrån, ligger på ungefär svårighetsgrad 3, har en slideshow, har en video <i>eller</i> bilder som demonstrerar utvecklingsprocessen;
Redovisning:	projektdefinition, git-repo, muntlig redovisning 26:e maj.

Hur projektet genomförs

Vecka 20 – projektdefinition

I projektdefinitionen skall ni bestämma er för vad ni vill göra under projektet, och ni **ska** göra det i diskussion med läraren för att kontrollera att ni inte tar er vatten över huvudet. Ni skall boka tid på doodle med läraren i vecka 20! Doodle-länken kommer ut måndag den 15:e maj!

Nivå: ni skall sätta er in i och använda **ett** (1) Python-bibliotek på svårighetsgrad 3, och ni ska planera att göra ett program i detta bibliotek.

Exempel på svårighetsgrader:

bibliotek	svårighetsgrad 1	svårighetsgrad 2	svårighetsgrad 3
math	areaberäkning	formelanvändning i fysik	många formelanvändningar i fysik, eller matematiska algoritmer för reella beräkningsproblem, t.ex. Newtons rothittarmetod

bibliotek	svårighetsgrad 1	svårighetsgrad 2	svårighetsgrad 3
datetime	tidsdifferenser	kalenderberäkning	årskalendrar med tidsmedvetenhet veckonummering, månadsupplägg, helger, samt händelser associerade till dagarna
tkinter		exempelkod, t.ex. texteditorn	en utvidgad texteditor med monospace i editorfönstret, och en proportionell font på övriga GUI:t, öppna, spara, sparasom, återställ, about-dialog
pygame		ett spelprogram som du hittar från nätet med egna ikoner och färgsättningar	ett spelprogram med nya tangenttryckningar, nya knapptryckningar, extra animationer, nya ljud
colorama	färger i consolen	en flashig meny-layout med clear screen	ett interaktivt spel med hjälp av colorama och windows-curses , eller gör en egen klon av colorama med utökad ANSI-funktionalitet
requests	tanka ner websidor	läs av ett API som du hittat på nätet och spara filen på JSON	läs av ett API och visa med matplotlib eller annan grafisk möjäng, alltså något mer utöver våra labbar den 9:e maj!
matplotlib	rita en kurva	rita flera kurvor som du räknar ut med hjälp av formler	ladda kurvor från fil och visa med hjälp av matplotlib
andra bibliotek	<i>förhandlingsfråga med läraren</i>	<i>förhandlingsfråga med läraren</i>	<i>förhandlingsfråga med läraren</i>

OBS: ni måste experimentera med biblioteket och någorlunda förstå koden, innan ni skriver specifikationen, annars riskerar ni att köra fast.

Specifikation: Din (er) projektspecifikation skall vara ett dokument (cirka ½ till 1 A4-sida) med:

1. ditt namn och ditt email,
2. en eller ett par meningar om vad för sorts program du planerar att göra,
3. namnet på det bibliotek som du tänker använda, och en länk till dokumentation på nätet,
4. om du bygger vidare på exempelkod från nätet (OBS: rekommenderas!), så behövs en länk till denna exempelkod, och en beskrivning om vad som skall vidareutvecklas, om inte så måste jag se på ditt git-repo hur du bygger upp koden från scratch
5. en länk till ett github-repo (*samt en inbjudan på slack till repot om ni absolut känner att ni måste krångla med privata repon!*) där projektet ligger!

OBS: Skicka dokumentet med email till tomas.kindahl@molk.se!

git-repot: ni kan skicka mig ett tomt repo, det är helt OK. Ni kan inte vara klara innan ni har börjat! Men ni ska använda repot för utvecklingsarbetet i vecka 21.

Ni måste få ett GO! (🚦) av läraren på slack för att ni ska gå vidare med genomförandet! Jag måste hinna läsa igenom dokumentet och titta på biblioteket som ni har valt ut.

Vecka 21 – genomförande

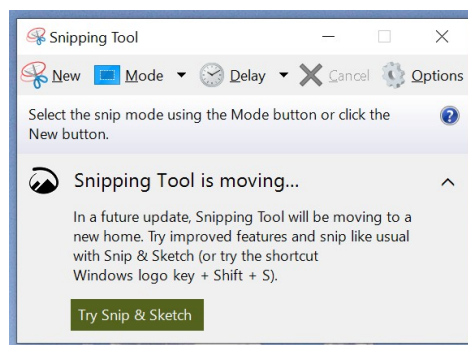
Koda dagtid i mellandagarna 8 timmar per dag, med raster! Gör git-pushes ofta och mycket! Så här granskar läraren er kodning:

1. ser redovisningen bra ut, finns det en kort och koncis berättelse om vad ni har gjort? demonstrerar ni programvaran genom att köra den?
2. går det att kлона er kod och köra den?
3. går det att se i git-historiken hur utvecklingsarbetet har fortgått? Läraren granskar git-historiken bland annat **för att kompensera** för den händelse att programmet inte går att köra vid redovisningen.

Använd git för att ni skall bli säkrare i er programmering, och för att ni skall ha en större chans att bli godkända.

Viktiga saker att tänka på:

1. För redovisningens skull är det en bra idé att göra dagliga anteckningar om svårigheter och framgångar. Gör sådana dagliga anteckningar! TomKi samlar dock inte på dem.
2. För redovisningens skull ta snapshots (t.ex. med hjälp av Snipping Tool) av era program så fort de börjar bli intressanta, gör också en demo-video,
3. Följ faserna i DTP! Om du kör fast, så beror det sannolikt på att du gör för många saker så att du tappar fokus, eller på att du kodat alldeles för mycket utan att testköra och nu sitter och letar efter felet. Använd git för att se vilka ändringar du gjorde när felet smög sig in! Sikta inte på någon stor buggrensning! Se till att få det att **funka någorlunda** bara!
4. Kontakta läraren **direkt** om det blir strul! Vi kan diskutera buggar och ändrade förutsättningar för projektet. Det är bättre att få ett problem utrett än att vänta till för sent och inte lyckas med redovisningen.



Vecka 21, fredag 26:e maj – redovisning

Den 25-26:e maj redovisar ni. Ni redovisar genom att

1. presentera era program för era klasskamrater, läraren skickar ut en doodle-länk där ni bokar (ungefärlig) tid, redovisningen skall gå till så här (ungefär):
 - a. hej jag heter X, jag har gjort ett Y-program (*robotspel, en väder-applikation, ett kalenderprogram, etc.*)
 - b. berätta litet mer om programmet, och vilket bibliotek det använder, att visa kod är inte nödvändigt,
 - c. visa programmet, eller en video på programmet, eller bilder på programmet,
 - d. berätta litet om svårigheter
 - e. berätta vad du lärde dig!
2. ladda upp saker på Moodle som läraren sedan kan sitta och gå igenom:
 - a. eventuell slideshow,
 - b. git-länk,
 - c. eventuella videos och bilder, allt som kan underlätta för läraren att godkänna.

Redovisning

Redovisning på zoom den 25-26:e maj. Läraren kommer att lägga ut en doodle-länk på slack den 24:e maj, samt en påminnelse om att förbereda redovisningen. Att tänka på inför den muntliga redovisningen:

1. glöm inte att presentera dig och säga vad ditt program gör,
2. var så kortfattad som möjligt i presentationen, gör inga långa tekniska utläggningar, säg bara hur du har gjort så snabbt som möjligt, och visa programmet om du har ett fungerande program,
3. förbered gärna en kort lista av saker du ska säga, gör en övning av presentationen om du är osäker, eller skriv ett manus som du läser ifrån när du redovisar!