**Schets-Editor Toelichting**

1. **CirkelTool**

Ik heb in de constructor-methode van “SchetsWin” de “CirkelTool” toegevoegd. Vervolgens heb ik een klasse “CirkelTool” en een “CirkelTool “toegevoegd die hetzelfde doen als een rechthoek maar dan maakt het cirkels in plaats van rechthoeken, dus met Ellipse in plaats van Rectangle. Tenslotte heb ik “Bezig” ge-overwrite zodat dit ook werkt en een icoontjes toegevoegd voor de cirkel tools.

1. Kleuren toegevoegd: "White", "Orange", "Moccasin", "Gray", "Pink", "Chocolate, "Gold", DarkGray", "Salmon", "GreenYellow", "Purple”, Turquoise","DodgerBlue" en "Thistle". Verder heb ik op de plek van het label: “Kleur:” een knop gemaakt die een ColorDialog laat zien waarmee je de kleur kan veranderen. In SchetsWin wordt de knop gemaakt en in SchetsControl staat de “KleurKiezer” methode die dan wordt aangeroepen door de knop en de ColorDialog opent.
2. Ik heb een variabele toegevoegd aan de klasse StartpuntTool genaamd “Lijndikte” die een standaard integer waarde van 3 heeft. Deze variabele wordt vervolgens gebruikt door alle Tools die een Lijndikte gebruiken, zoals de LijnTool of de RechthoekTool (het kader). Vervolgens heb ik in SchetsWin 2 knoppen toegevoegd, een +knop en een -knop, die dan VergrootDikte en VerkleinDikte in SchetsControl aanroepen. Deze methodes vergroten/ verkleinen de lijndikte met 1 en roepen Invalidate aan. VerkleinDikte doet dit alleen zolang de dikte groter is dan 0, want een kleinere Lijndikte is niet mogelijk en levert exceptions.
3. **GumTool**

Onder in de Tool klasse heb ik een klasse “Element” gemaakt die belangrijke variabelen van een element als parameter heeft, deze opslaat in een lokale variabele en vervolgens teruggeeft via een property zodat het makkelijk terug te krijgen is.

In de Schets-Klasse heb ik een lijst van Elementen gemaakt genaamd “elementenlijst”.

Alle Tools (behalve de TekstTool) hebben geen override “compleet”methode meer, en worden dus allemaal verwezen naar de “compleet”methode uit Tweepunt-tool. In deze methode wordt er een nieuwe Element gemaakt met als parameters, p1, p2, de penkleur, de tool (hiervoor wordt this gebruikt, zodat hij altijd de huidige tool opslaat), pendikte en een string waarin de tekst van de TekstTool kan worden opgeslagen (bij de compleet-methode is dit een lege string, omdat het hier nog niet van pas komt). Dit nieuwe Element wordt vervolgens toegevoegd aan de elementenlijst.

In de teken-methode van SchetsControl wordt vervolgens voor elk element uit de elementenlijst (met foreach) gekeken welke tool het element is (met elem.Tool, het property Tool die meegegeven is aan het element). Het tekent dan het Element op de manier van de Tool. Dus een RechthoekTool, wordt gemaakt met g.DrawRectangle(…). Aan het eind van de teken-methode wordt schets.Teken(pea.Graphics); gedaan zodat de Graphics op de bitmap getekend wordt.

In de klasse “GumTool” heb ik alle inhoud uit de methodes verwijderd en heb ik de MuisDrag en MuisLos methodes opnieuw ingevuld. Met foreach kijk in voor elk Element of het geklikte punt ook raak op het Element is, de “if” verschilt per Tool, omdat het per Tool verschilt wat “raak” is. Als het raak is en elem.Tool is ook het gewenste Tool kunnen we de bitmap leegmaken (door een wit vlak boven op de bitmap te tekenen), het element (dat op dat moment raak geklikt is) uit de elementenlijst verwijderen, Invalidate aanroepen zodat alle elementen opnieuw getekend worden en tenslotte break aanroepen zodat de foreach niet verder gaat met de andere Elementen zodra hij de juiste heeft. Er staat bovenin Muislos een for loop die elk element afgaat van het nieuwste element naar element op positie 0, zo gumt hij het voorste element eerst.

1. **Opslaan**

In de Element klasse heb ik een methode stringmaken gemaakt die van een element een string maakt, door alle elementen om te zetten naar strings en vervolgens achter elkaar te plakken met een ‘/’. Ik heb een klasse Uitstring gemaakt met een methode UitString die een string omzet in een element, door de string te splitten op het ‘/’ teken en vervolgens elk element variabele veranderen in de nieuwe waarde. Hiervan wordt vervolgens een Element gemaakt en toegevoegd aan de elementenlijst.

In SchetsWin is er een methode genaamd maakFileMenu. Hier heb ik “opslaan” aan toegevoegd zodat er een opslaan knop is. Deze knop staat in maakSaveknop, die vervolgens schetscontrol.opslaan aanroept. Deze methode roept vervolgens Bestand.SaveActies aan en geeft de ingetikte filenaam mee. SaveActies maakt dan van elk element uit de lijst een string met “stringmaken” en schrijft deze op de StreamWriter.

**Openen**

In Hoofdscherm wordt een “openen”knop gemaakt, zodat hij altijd in de menubalk tevoorschijn komt. Deze knop stuurt uiteindelijk de methode openen aan in SchetsControl. Deze methode roept voor elke regel in het bestand UitString aan met de string van de regel. Dit wordt vervolgens in UitString omgezet in een element en toegevoegd aan de lijst. De methode openen roept tenslotte invalidate aan zodat alle elementen getekend worden.