DESIGNDOKUMENT

D.1. UPPGIFTEN
D.1.1 Syfte 2
D.1.2 Avgränsningar 2
D.1.3 Kravspecifikation 2
D.1.4 Kvalitetskrav 3

D.2 HÅRDVARA OCH MJUKVARA
D.2.1 Designspråk och databassystem 4
D.2.2 Datormiljö 4

D.3 FUNKTIONER
D.3.1 Funktionslista för klassen Loader 5
D.3.2 Programkod för laddaren av figurvalet 6
D.3.3 Funktionslista för klassen Karta 7
D.3.4 Programkod för klassen Karta 13
D.3.5 Funktionslista för klassen Hero 48
D.3.6 Programkod för klassen Hero 48
D.3.7 Funktionslista för klassen ControlPanel 59
D.3.8 Programkod för klassen ControlPanel 60

D.4 ANVÄNDARGRÄNSSNITT 66

Kommentarer:

- Designdokumentet visar konkret programkoden av utvalda centrala funktioner i spelet Stundars tidsresa.
- Spelet finns tillgängligt på internet. Adressen är http://stundars.img.fi/main.html

D.1 UPPGIFTEN

D.1.1 Syfte

Avsikten med detta designdokument är att visa hur utvalda centrala funktioner i spelet Stundars tidsresa är programmerade i praktiken. Den mest centrala instansen av spelet är klassen karta vars hela programkod finns att läsa i detta designdokument. Klassen karta består av spelets exteriörer (gårdar) där spelaren rör sig, rollfigurer (interaktiva spelfigurer), kvarnen, frågesporten, avslutning, film och audio. Klasserna hero (spelfiguren), ControlPanel (spelmenyn), och Figloader (laddare) beskrivs också eftersom de spelar centrala roller i spelets funktionalitet.

D.1.2 Avgränsningar

Eftersom programkoden är massiv och består av mera än 11000 kodrader avgränsas detta designdokument till spelets viktigaste och mest centrala klasser på begäran av avdelningsföreståndare Kenneth Norrgård och handledare Börje Harju.

Programkoden för Flash kommunikation med XML-filerna som skapats via PHP från spelets MySQL databas lämnas bort eftersom databasens roll och användning i spelet är minimal. MySQL och PHP programmeringen gjordes av en medarbetare inom projektet.

D.1.3 Kravspecifikation

1. Syftet med den virtuella tidsresan är att på ett lustfyllt sätt lära sig om livet på landsbygden i svenska Österbotten under början av 1900-talet. Den virtuella tidsresan lämpar sig både som stöd till undervisningen i skolan och som fritidsaktivitet. Målsättningen är att ge en större medvetenhet, kunskap och förståelse för den kulturmiljö barnen lever i. Målsättningen är en professionell produkt, både när det gäller det pedagogiska innehållet samt de uppgifter som skall utföras.

   Stor vikt läggs även på produktens gränssnitt och illustrationer som skall inspirera barnen att återkomma till tidsresan. Systemet skall vara tvåspråkigt (finska/svenska) och man skall kunna byta språk från huvudmenyn. Tidsresan riktar sig framför allt till barn i åldern 6-12 år. Illustrationerna görs trevliga i typisk spelstil/”seriefigurs stil” som tilltalar barn. Miljöerna görs trevliga genom att saker rör sig och något smått hela tiden händer.

   Det finns möjlighet att besöka: Exteriören för A. Smedens stuga, B. Lanthandeln / Skolan, C. Bondgården, D. Kvarnen E. Kloka gummans stuga. Interiören för A. Lanthandeln, B. Bondgården och C. Smedens stuga.

2. Det skall finnas funktioner/dialogfönster för följande: uppdatering av medlemmar, utskrift av etiketter, utskrift av övriga rapporter, uppföljning av betalningar, sök medlem...

3. Spelet har inga kopplingar till andra system, men det hämtar text ur en simpel databas vilken innehåller texten för föremål i spelet och för information om livet förr.

4. Spelet körs i en webläsare med Flashplayer. Eftersom Flash player körs via en webläsare är spelet tillgängligt för Windows, Mac och Linux.

D.1.4 Kvalitetskrav

Nr	**Krav**	**Att ta ställning till...**
1	Användbarhet	Stor vikt läggs på gränssnittet.
2	Säkerhet	Flash och ActionScripts inbyggda säkerhet.
3	Effektivitet	Spelet fungerar på en några år gammal laptop ämnad för internetsurfning.
4	Korrekthet	Kvalitet prioriteras utöver kvantitet.
5	Tillförlitlighet	Spelet fungerar då användaren har normala webläsar inställningar.
6	Lättunderhållet	Spelet kan relativt lätt utvecklas och utökas.
7	Flexibilitet	Tack vare Flash är det mycket enkelt att modifiera spelet.
8	Begriplighet	Det bör vara lätt att förstå systemet eftersom det är ämnat för barn.
9	Återanvändbarhet	Det är önskvärt att klasser kan återanvändas.
10	Portabilitet	Spelet körs i en webläsare och är alltså cross-platform.
11	Interoperabilitet	Uppkoppling till databas för att kunna uppdatera vissa texter via ett anpassat gränssnitt för en webläsare.

D.2 HÅRDVARA OCH MJUKVARA

D.2.1 Designspråk och utvecklingsmiljö

Stundars tidsresa är utvecklat i Flash CS4 och programmerat med hjälp av språket ActionScript 3.0.

D.2.2 Datormiljö

Flash player är tillgänglig för alla moderna operativsystem eftersom den körs i en webläsare. Spelet kan alltså köras i Windows, Mac och Linux.

D.3 FUNKTIONER

D.3.1 Funktionslista för klassen Figloader

Detta är ett exempel på en av många nästan identiska laddare som användes i spelet vid inladdning av olika spelinstanser och vid kommunikation av variabler mellan olika spelinstanser.

Nr	Funktion	Komplexitet	typ
F1	figloaded	Normal	Initialisering
	Initialisering av inladdad instans		
F2	figloader	Normal	Laddning
	Inladdning av instans		
F3	killLoadedClip	Enkel	Minnestömning
	Ta bort instans ur minnet		
F4	onAllItemsLoaded	Enkel	Laddning
	Initialisering av inladdad instans		
F5	onProgressHandler	Enkel	Laddning
	Laddningsgrafik		
F6	rollfig	Enkel	Kommunikation
	Skickar variabler mellan instanser		

F1 Skickar variabler till figurvalet (child). Vilket språk som spelaren valt och en variabel för att kontrtollera att laddaren laddat allting klart och placerat den inladdade instansen på scen.

F6 Hämtar från figurvalet (child) kön variabeln beroende på vilken spelfigur spelaren väljer. Laddarens parent (startmenyn) kan kalla på denna funktion för att få veta vilken spelfigur spelaren valde. Detta måste göras på detta sätt för i ActionScript 3 går det inte att kalla på variabler två steg upp i hierarkin då instanser laddas in.

Laddningen var för mig den överlägset svåraste utmaningen i skapandet av Stundars tidsresa. Jag äger många böcker angående ActionScript 3 men ingen av dessa tar upp laddningen och interaktionen mellan olika swf filer på et konkret sätt så jag var helt tvungen att förlita mig till internet och jag hittade till sist några programmeringsbloggar som tog upp saken och praktiskt visade hur vissa saker bör göras. Jag hittade också biblioteket BulkLoader som förenklar in- och urladdningen. Bibliotekets metoder fungerar likt Flash egna, skillnaden är att den grundläggande laddningskonstruktionen är färdigt programmerad. Efter en del försök och misstag bemästrade jag slutligen in och urladdningen med hjälp av BulkLoader.

Jag visste från början att spelet skulle ladda in en huvudinstans dvs. exteriören i spelet, och därifrån skulle mindre spel (swf filer) laddas in och tömmas ur minnet vartefter spelaren rör sig mellan de olika delarna av spelet. Eftersom ActionScript 3 inte direkt stöder globala variabler mellan olika inladdningsinstanser bör olika globala variabler skickas från exteriören till laddaren som sedan skickar variabeln till den inladdade instansen. Om någon av dessa variabler sedan ändras i den inladdade instansen, som t.ex. att en viss uppgift utförs, bör variablerna sedan skickas tillbaka via laddaren när instansen avslutas innan den laddade instansen och laddaren töms ur minnet. Jag studerade först enklare alternativ till detta men de visade sig vara opålitliga och det är ändå allmänt sett bättre programmeringssed att skicka variabler mellan funktioner än att använda globala variabler. Denna laddning och

variabelstafett visade sig dock inte helt oförväntat vara en av de långsammaste och betungande delarna att programmera.

D.3.2 Programkod för klassen Figloader

```
/****   Klassen Figloader              ****/

package {
        import flash.display.*;
        import flash.events.*;

        import br.com.stimuli.loading.BulkLoader;
        import br.com.stimuli.loading.BulkProgressEvent;
        import flash.system.LoaderContext;
        import flash.system.ApplicationDomain;
        import flash.net.*;

        public class Figloader extends MovieClip {

                // Variabler.
                public var loaderFig : BulkLoader = new BulkLoader("figurval");
                public var LoaderContextFig:LoaderContext = new LoaderContext(false, Figdomain);
                public var Figdomain:ApplicationDomain = new ApplicationDomain();
                public var loadReady:Boolean = false;
                public var figval:MovieClip;
                private var kon:String;

                public function figloader() {

                        loaderFig.add("Figurval.swf");
                        loaderFig.addEventListener(BulkLoader.COMPLETE, onAllItemsLoaded);
                        loaderFig.addEventListener(BulkLoader.PROGRESS, onAllItemsProgress);
                        loaderFig.addEventListener(BulkLoader.PROGRESS, onProgressHandler);

                        // Starta laddningen.
                        loaderFig.start();

                }
                public function onAllItemsLoaded(evt : Event) : void{
            figloaded();

                }

                public function figloaded(){
                        trace("every thing is loaded!");
                        figval = loaderFig.getMovieClip ("Figurval.swf");
                        addChild(figval);

                        var passvar:int;
                        var passloaded:Boolean;
                        passvar = MovieClip(parent).whatLanguage;
                        passloaded = MovieClip(parent).loadedOnce;
                        //Skickar variabler till figurvalet. Vilket språk som spelaren valt och en
                        variabel för att kontrtollera att laddaren laddat allting klart och placerat
                        den inladdade instansen på scen.
                        figval.varReceiver(passvar, passloaded);

                        loadReady = true;
                        MovieClip(parent).lbar.x = 2222;
                        MovieClip(parent).loaderkvarn.x = 2222;
                        MovieClip(parent).loadervinge.x = 2222;
                        MovieClip(parent).loadingmaask.x = 2222;
                        MovieClip(parent).loadingmaask.alpha = 0;

                }


                function onProgressHandler(e:BulkProgressEvent):void {
```

```
                MovieClip(parent).loadervinge.rotation += 4;
            }

            public function killLoadedClip():void {

                loaderFig.remove("figurval");
                loaderFig.clear();
                removeChild(figval);
                MovieClip(parent).rollfigscheck(kon);
                MovieClip(parent).loadParaihop();

            }
            //Hämtar från figurvalet (child) kön variabeln beroende på vilken spelfigur
            spelaren väljer. Laddarens parent (startmenyn) kan kalla på denna funktion för att
            få veta vilken spelfigur spelaren valde.
            public function rollfig(passvar){
                kon = passvar;
                trace("kon "+kon);
            }

        }

}
```

D.3.3 Funktionslista för klassen Karta

I denna klass finns även funktionerna för exteriörerna, gränssnittet, rollfigurerna, skolfilmen, kvarnen, frågesporten, avslutningen och audio. Klassen är också tätt associerad med klassen Hero, vilket är själva figuren spelaren styr.

Avgränsning. Ett antal knappfunktioner är bortlämnade från funktionslistan p.g.a att de är så gott som identiska.

Nr	Funktion	Komplexitet	typ
F1	karta	Enkel	Initialisering
	Initialisering		
F2	kartaLoop	Normal	Initialisering
	Main loop		
F3	popAlla	Normal	Pop-up
	Pop-up alla uppdrag utförda		
F4	startParty	Normal	Spel-logik
	Spelets avslutningsfest		
F5	clicksmedbror	Normal	Spelfigur
	Interaktion med spelfiguren		
F6	partyFade	Normal	Grafikeffekter
	Grafikeffekter i spelavslutningen		
F7	popEnd	Normal	Pop-up
	Frågar om spelaren vill avsluta spelet		
F8	clickNo	Normal	Knapp
	Spelaren vill inte avsluta		
F9	clickYes	Normal	Knapp
	Spelaren vill avsluta och gå till startmenyn		
F10	endMalaren	Enkel	Avslutning
	Avsluta kvarnfilmen		
F11	endKarta	Normal	Avsluta instans
	Avslutar denna instans av klassen karta		
F12	initExterior	Normal	Initialisering
	Initialisering		

F13	popupVisste	Normal	Pop-up
	Pop-up ”visste du att”		
F14	moveVisste	Normal	Pop-up
	Visa ”visste du att” interfacet		
F15	hideVisste	Normal	Pop-up
	Gömm ”visste du att” interfacet		
F16	inituppdrag	Normal	Initialisering
	Initialisering av uppdragslistan		
F17	showUpBockar	Enkel	Grafik
	Visa uppdragsbokens bockar		
F18	checkuppdrag	Normal	Kontroll
	Kontrollerar vilka uppgifter som är utförda		
F19	showHelp	Enkel	Meny
	Visa spelinstruktionerna		
F20	clickbtn_backskol	Normal	Film
	Avslutar skolans interiör		
F21	pekabtn_backskol	Enkel	Knapp
	Pekar på skoldörren (over)		
F23	dissabtn_backskol	Enkel	Knapp
	Markören ej mera på knappen (out)		
F24	clickKvarndorr	Normal	Spelfigur
	Spelfiguren mjölnaren		
F25	onFLVComplete	Normal	Film
	Spela upp kvarnfilmen		
F26	removeFLVComplete	Normal	Film
	Ta bort lyssnaren		
F27	pekaGunmask	Normal	Knapp
	Markören på slangbellan (over)		
F28	dissaGunmask	Enkel	Knapp
	Markören ej mera på knappen (out)		
F29	clickGunmask	Normal	Knapp
	Spelaren klickar på slangbellan		
F30	layers	Enkel	Kommunikation
	objektens z ordning kan kallas från laddare		
F31	Mjoltimer	Normal	Timer
	Initialisering av ny timer		
F32	mjolTick	Enkel	Timer
	Animering av mjölnaren.		
F33	pekaKvarndorr	Normal	Knapp
	Knapp (over)		
F34	dissaKvarndorr	Normal	Knapp
	Knapp (out)		
F35	clickSmeddorr	Normal	Knapp
	Spelaren klickar på smedens dörr		
F36	pekaSmeddorr	Normal	Knapp
	Markören pekar på smedens dörr.		
F37	dissaSmeddorr	Normal	Knapp
	Markören pekar ej mera smedens dörr.		
F38	clickbtn_handeldorr	Normal	Knapp
	Spelaren klickar på handelns dörr.		

F39	didLoadReady	Normal	Initialisering
	Initialisering av ”visste du att”		
F40	pekabtn_handeldorr	Normal	Knapp
	Markören pekar på handelns dörr		
F41	dissabtn_handeldorr	Enkel	Knapp
	Markören pekar ej mera på knappen		
F42	clickbtn_klokdorr	Normal	Spelfigur
	Spelaren klickar på spelfiguren kloka gumman (knapp)		
F43	pekabtn_klokdorr	Normal	Knapp
	Markören pekar på kloka gumman.		
F44	dissabtn_klokdorr	Enkel	Knapp
	Markören pekar ej mera på knappen		
F45	clickpil_N_E	Normal	Knapp
	Spelaren klickar på navigeringspilen		
F46	pekapil_N_E	Normal	Knapp
	Spelaren pekar på navigeringspilen		
F47	dissapil_N_E	Enkel	Knapp
	Markören pekar ej mera på knappen		
F48	clickuppg_sack	Normal	Knapp
	//Spelaren klickar på mjölsäcken		
F49	popUtford	Normal	Pop-up
	pop-up fönstret "uppgift utförd"		
F50	clickBonddorr	Normal	Knapp
	Spelaren klickar på bondgårdsdörren		
F51	pekaBonddorr	Normal	Knapp
	Markören pekar på bondgårdsdörren		
F52	dissaBonddorr	Enkel	Knapp
	Markören pekar ej mera på knappen		
F53	clickbtnxskolpop	Normal	Knapp
	klickar på aktiv skoldörr		
F54	clickSkoldorr	Normal	Knapp
	klickar på inaktiv skoldörr		
F55	pekaSkoldorr	Normal	Knapp
	Markören pekar på skoldörren		
F56	dissaSkoldorr	Enkel	Knapp
	Markören pekar ej mera på skoldörren		
F57	clickbtn_griff	Normal	Knapp
	Spelaren klickar på knappen för griffeltavlan		
F58	pekabtn_griff	Enkel	Knapp
	Markören pekar på knappen		
F59	dissabtn_griff	Enkel	Knapp
	Markören pekar ej på knappen		
F60	clickbtn_film	Normal	Knapp
	Spelaren klickar på knappen för skolfilmen		
F61	pekabtn_film	Enkel	Knapp
	Spelaren pekar på knappen		
F62	dissabtn_film	Enkel	Knapp
	Markören pekar ej på knappen		
F63	clickChick	Normal	Knapp

	Spelaren klickar på hönan		
F64	pekabtn_hus1	Normal	Knapp
	Markören pekar på lillstugan		
F65	pekabtn_hus2	Normal	Knapp
	Markören pekar på bondstugan		
F66	pekabtn_hus3	Normal	Knapp
	Markören pekar på loftsraden		
F67	pekabtn_hus4	Normal	Knapp
	Markören pekar på fähuset		
F68	dissabtn_hus1	Normal	Knapp
	Markören pekar ej på någon av husknapparna		
F69	keyDownFunction	Normal	Tangentbord
	Tangentbordets knappar (knapp tryckt)		
F70	keyUpFunction	Normal	Tangentbord
	Tangentbordsknapp ej mera nertryckt.		
F71	exteriorChange	Normal	Timer
	Start av exteriörändring		
F72	realchange	Normal	Logik
	Ändringen av exteriör		
F73	changeBg	Normal	Logik
	Spelfiguren går varifrån vart		
F74	fadeIn	Enkel	Grafik
	Fade in för hela scenen		
F75	fadeOut	Enkel	Grafik
	Fade out för hela scenen		
F76	stangLjud	Enkel	Audio
	Stoppa audion		
F77	ambientljud	Enkel	Audio
	Starta ljud		
F78	sfxAmbientCompleteHandler	Enkel	Audio
	När ljudet spelats, starta det pånytt		
F79	endambientljud	Enkel	Audio
	Stoppa ljudet		
F80	fagelljud	Enkel	Audio
	Starta ljud		
F81	sfxBirdsCompleteHandler	Enkel	Audio
	När ljudet spelats, starta det pånytt		
F82	endfagelljud	Enkel	Audio
	Stoppa ljudet		
F83	dorrljud	Enkel	Audio
	Starta ljudeffekt		
F84	sfx_doorCompleteHandler	Enkel	Audio
	Ljudet spelat till slut		
F85	uppgutfljud	Enkel	Audio
	Starta ljudeffekt		
F86	sfx_uppgutfCompleteHandler	Enkel	Audio
	Ljudet spelat till slut		
F87	ljudfanfare	Enkel	Audio
	Starta ljudeffekt		
F88	sfx_fanfareCompleteHandler	Normal	Audio
	Ljudet spelat till slut		

F89	deleteKarta	Normal	Minnestöming
	Ta bort denna instans (karta)		
F90	stopHero	Normal	I/O
	Stoppa spelfiguren		
F91	startHero	Normal	I/O
	Aktivera spelfiguren		
F92	ordnaUppgift	Normal	Grafik
	Menyobjekts z kordinater		
F93	ordnaFjaril	Normal	Musknapp
	z-kordinater		
F94	ordnaTomte	Normal	Kolission
	z-kordinater		
F95	endSkolGrif	Normal	Funktionalitet
	Uppdrag utfört		
F96	endUppgKloka	Normal	Funktionalitet
	Uppdrag utfört		
F97	endUppgBond	Normal	Funktionalitet
	Uppdrag utfört		
F98	handlaren	Normal	Funktionalitet
	Uppdrag utfört		
F99	handlarenUpp1	Normal	Funktionalitet
	Uppdrag utfört		
100	handlarenUpp2	Normal	Funktionalitet
	Uppdrag utfört		
101	smedenUpp1grot	Normal	Funktionalitet
	Uppdrag utfört		
102	smedenUpp2nagg	Normal	Funktionalitet
	Uppdrag utfört		
103	memoryUtford	Normal	Funktionalitet
	Uppdrag utfört		
104	varReceiver	Normal	Kommunikation
	Mottagande av variabler		
105	loadGriffel	Normal	Laddare
	Laddare (av griffeltavlan)		
106	loadShoot	Normal	Laddare
	Laddare (skjutspelet)		
107	loadBond	Normal	Laddare
	Laddare (bondstugans interiör)		
108	loadKlok	Normal	Laddare
	Laddare (memoryspelet)		
109	loadMus	Normal	Laddare
	Laddare (handelns interiör och musspelet)		
110	loadSmed	Normal	Laddare
	Laddare (smedens interiör)		
111	doNotWalk	Enkel	I/O
	Inaktivera spelfiguren tillfälligt		
112	ReallydoNotWalk	Normal	Timer
	Inaktivera spelfiguren tillfälligt		
113	walkTo	Normal	Funktionalitet
	Flytta inte på spelfiguren i vissa sammanhang		

F2 Main loop, handhar främst ordningen av objektens z-kordinater (vilka objekt som syns ovanför andra objekt).

F71 Funktion för start av ändring av exteriör (t.ex. spelaren går från smedens exteriör till handelns exteriör). Timern görs enkelt med tweener och en dummygrafik flyttas utanför spelrutan i en sekund och efter det kallas den riktiga funktionen för ändring realchange().

Huvudarean i spelet består av stugornas exteriörer, alltså själva området Stundars. Här kan spelaren gå mellan de olika husen och rollfigurerna och söka efter spel och uppgifter. Ur ren programmeringssynpunkt är exteriören den centrala delen av spelet som knyter samman de andra småspelen till en helhet. Filen växte enligt sin centrala roll under programmeringens gång och blev den största filen med den längsta koden i hela spelet.

Beroende på vilken exteriör spelaren befinner sig i spelas olika omgivningsljudeffekter upp, som t.ex. kraxande kråkor, susande vind och kacklande hönor. När spelaren pekar på en dörr dit spelaren kan gå in, öppnas dörren och en ljudeffekt spelas upp. I vissa fall då spelaren pekar på en dörr dycker även en pop-up knapp upp, vilken leder till lärorik information om livet förr.

De olika rollfigurerna i spelet så som Smedis Maria, kloka gumman, handlaren och mjölnaren ville jag programmera så att de alla skulle bygga på samma basklass för att standardisera konversationer och interaktioner mellan rollfigurerna och spelfiguren. Detta visade sig dock vara väldigt invecklat och lyckades endast delvis eftersom de olika spelsituationerna krävde att vissa funktioner skräddarsyddes. Det visade sig också att i vissa situationer var det lättare att sköta konversationerna via Flash tidslinje och i andra situationer helt och hållet via ActionScript. Spelets användargränssnitt tas upp i kapitel 6 i detta designdokument.

Bild 1. Kvarnens exteriör, spelarens rollfigur och mjölnaren.

D.3.4 Programkod för klassen Karta

```
/****   Klassen Karta          ****/

package {

        import flash.display.*;
        import flash.events.*;
        import flash.utils.Timer;
        import caurina.transitions.Tweener;

        import br.com.stimuli.loading.BulkLoader;
    import br.com.stimuli.loading.BulkProgressEvent;
        import flash.system.LoaderContext;
        import flash.system.ApplicationDomain;
        import flash.events.ProgressEvent;
        import flash.net.*;
        import fl.video.VideoEvent;
        import flash.media.Sound;
        import flash.media.SoundChannel;

        public class Karta extends MovieClip {

                static const borderBottom = 436;

                public var griffelloadern:Griffelloader = new Griffelloader();
                public var shootloadern:Shootloader = new Shootloader();
                public var bondloadern:Bondloader = new Bondloader();
                public var klokloadern:Klokloader = new Klokloader();
                public var musloadern:Musloader = new Musloader();
                public var smedloadern:Smedloader = new Smedloader();
                public var herostartX:int = 600;
                public var herostartY:int = 324+66;

                public var hero:Hero;
                public var maria:Smedismaria = new Smedismaria();
                public var chiq:Chicken = new Chicken();
                public var cannotMove:Boolean = false;
                public var bg:BgSwitcher;
                public var cpanel:ControlPanel;
                public var leftArrow, rightArrow, upArrow, downArrow, keyR:Boolean;

                //1 = smeden, 2 = handeln/(skolan), 3  = Bondgården, 4 = Kvarnen, 5 = Kloka gummans
                stuga.
                public var partofMap:int = 1;

                private var arrowSignLeft:Arrowsign;
                public var mouseOnstage:Boolean;
                private var startHeroOnce:Boolean;
                private var xuppdragIsHit:Boolean;

                // Tabell av objekten som utgör gränserna var spelfiguren kan röra sig i varje
                exteriör.
                public var boundaryArray:Array = new Array();

                public var buttonEsc:Boolean;
                public var popupskola:PopupSkola = new PopupSkola();
                private var klickSkoldorr:Boolean = false;
                private var onetimekvarn:Boolean = false;
                private var loadedOnceGriffel = false;
                private var loadedOnceShoot = false;
                private var loadedOnceBond  = false;
                private var loadedOnceKlok  = false;
                private var loadedOnceMus  = false;
                private var loadedOnceSmed  = false;
                public var removekvarndorrsMask:Boolean = false;
                private var fadeinonce:Boolean; //Fade in kontroll, (endast en gång).
```

```
public var sfx_doorComplete:Boolean = true;
private var sfx_door:Sfx_door = new Sfx_door();
private var sfx_doorChannel:SoundChannel;

public var sfx_uppgutfComplete:Boolean = true;
private var sfx_uppgutf:Sfx_uppdragutfort = new Sfx_uppdragutfort();
private var sfx_uppgutfChannel:SoundChannel;

public var sfx_fanfareComplete:Boolean = true;
private var sfx_fanfare:Sfx_fanfare = new Sfx_fanfare();
private var sfx_fanfareChannel:SoundChannel;

public var sfxAmbientComplete:Boolean = true;
private var sfxAmbient:SfxAmbient = new SfxAmbient();
private var sfxAmbientChannel:SoundChannel;

public var sfxBirdsComplete:Boolean = true;
private var sfxBirds:SfxBirds = new SfxBirds();
private var sfxBirdsChannel:SoundChannel;

private var kvarnfilmActive:Boolean = false;

//Aktivera z kordinat-ordnandet.
public var lagaLager = true

public var howmanyUppdrag:int = 7;
public var tblUppdrag:Array = new Array(); //För att kolla uppdragen.
public var tblUppbockar:Array = new Array();

public var vissteduatt:Didyouknow = new Didyouknow;
public var showSack:Boolean = false;
private var uppg_sackTaken:Boolean = false;
public var donotwalk:Boolean = false;
private var clkickedWalkOnce:Boolean = false;

public var allaUppUtforda:Boolean = false;
private var allaUppOnce:Boolean = true;

//Variabler för att hålla reda på vad spelaren gjort inom samma spelomgång.
public var harTalatHandlare:Boolean = false;
public var harkokatgrot:Boolean = false;
public var harnaggatbrod:Boolean = false;
private var harGriflat:Boolean = false;
private var harSkolfilm:Boolean = false;
private var harSkoldagOnce:Boolean = true;
private var harKloka:Boolean = false;
private var harKlokaOnce:Boolean = true;

public var harBond:Boolean = false;
private var harBondOnce:Boolean = true;

private var kvarnmode:int = 9;
private var onceSoundKD:Boolean = true;

//"Flicka", "pojke" eller "null".
public var rollfigurK:String;
// 1 = sve 2 = fin.
public var whatLanguage:int;

private var smedbrorState:int = 0;

public var popupOnceHandel:Boolean = true;
public var popupOnceSkola:Boolean = true;
public var popupOnceBond:Boolean = true;
public var popupOnceKvarnen:Boolean = true;
public var popupOnceKloka:Boolean = true;

private var showSackOnce:Boolean = false;
public var ljudAv:Boolean = false; //Ljud av eller på.
public var delAvSpel:String = "kartan";
```

```
private var mjoltimer:Timer;
private var switchPrat:Boolean = false;
public var pil_N_E_Bool:Boolean;
public var pekahus:Boolean = false;

//Initialisering och start av Main loop.
public function karta() {

        //Fade in grafiken.
        fadeIn();

        //Initialisera.
        initExterior();

        // Main Loop = kartaLoop.
        addEventListener(Event.ENTER_FRAME,kartaLoop);

}

//Main loop, handhar främst ordningen av objektens z-kordinater (vilka objekt som
syns ovanför andra objekt).
public function kartaLoop(event:Event){

        //Stoppa eller aktivera spelfiguren.
        if(cannotMove == true){
        hero.stopHero();
        stopHero();
        }
        if(cannotMove == false){
        hero.moveHero();
        hero.animateHero();
        }

        // Förhindra spelfiguren att röra på sig förrän spelaren pratat färdigt med
        Maria.
        if(maria.harPratatMaria == true && startHeroOnce == false){
                startHero();
                startHeroOnce = true;
        }

        // Ordna lager alltså objekt efter deras z-kordinat.
        if(cpanel.uppdragActive == false && cpanel.mapActive == false &&
        cpanel.vissteActive == false && cpanel.helpActive == false){
                switch(partofMap){
                        case 1: //Smeden.
                        if(lagaLager == true){

                                if(hero.y >= maria.y + 78+66){
                                        setChildIndex(hero, numChildren-1);
                                }else if(hero.y < maria.y + 78+66){
                                        setChildIndex(maria, numChildren-1);
                                }
                                setChildIndex(gunmask, numChildren-1);
                                setChildIndex(pil_N_E, numChildren-1);

                                setChildIndex(arrowSignLeft, numChildren-1);
                        }
                        break;
                        case 5: //Kloka gumman.
                        //Höna.
                        if(lagaLager == true){
                                setChildIndex(tomten, numChildren-1);

                                if(tomten.activeTomte == true){

                                }else tomteknut.x = 2222;

                                setChildIndex(fjarilen, numChildren-1);
                                if(hero.y >= btn_klokdorr.y + 48+66){
```

```
                                setChildIndex(btn_klokdorr, numChildren-1);
                                setChildIndex(hero, numChildren-1);
                        }else if(hero.y < btn_klokdorr.y + 48+66){
                                setChildIndex(hero, numChildren-1);
                                setChildIndex(btn_klokdorr, numChildren-1);
                        }
                        if(hero.y + 32 >= chiq.y){
                                setChildIndex(hero, numChildren-1);
                        }else if(hero.y + 32 < chiq.y){
                                setChildIndex(chiq, numChildren-1);
                        }
                        setChildIndex(pil_N_E, numChildren-1);

                        setChildIndex(arrowSignLeft, numChildren-1);
                }
                break;

        }

}

if(partofMap == 2){
        skoldorr.y = 243;
        btn_handeldorr.y = 232;

        if(cpanel.uppdragActive == false && cpanel.mapActive == false &&
        cpanel.vissteActive == false && cpanel.helpActive == false){
                if(lagaLager == true){
                        setChildIndex(pil_N_E, numChildren-1);
                        setChildIndex(arrowSignLeft, numChildren-1);
                }
        }

}else{
        skoldorr.y = -555;
        btn_handeldorr.y = -555;
}

if(partofMap == 3 && cpanel.uppdragActive == false && cpanel.mapActive ==
false && cpanel.vissteActive == false && cpanel.helpActive == false &&
lagaLager == true){
        setChildIndex(pil_N_E, numChildren-1);
        setChildIndex(brunnen, numChildren-1);
        setChildIndex(arrowSignLeft, numChildren-1);

}
//Reset hönan om den ska synnas (exteriör 1).
if(partofMap == 5 && chiq.chickenActive == false){
        chiq.x = chiq.startx;
        chiq.y = chiq.starty;
        chiq.chickenActive = true;
        chiq.chickCounter = 0;
        btn_klokdorr.x = 98+45;
        setChildIndex(btn_klokdorr, numChildren-1);

        if(cpanel.uppdragActive == false && cpanel.mapActive == false &&
        cpanel.vissteActive == false && cpanel.helpActive == false){
                if(lagaLager == true){
                        setChildIndex(pil_N_E, numChildren-1);
                        pil_N_E.x = 750;
                        pil_N_E.y = 240;
                        setChildIndex(arrowSignLeft, numChildren-1);
                }
        }

//Annars göm hönan.
}else if(partofMap != 5){
        chiq.x = 2222;
        chiq.chickenActive = false;
```

```
                btn_klokdorr.x = 2222;
                tomteknut.x = -333;
        }

        if(partofMap == 4){
                kvarnvingarna.visible = true;
                kvarnvingarna.rotation += 3;

                if(kvarnfilmActive     == true){
                        malarvingar.visible = true;
                        malarvingar.x = 460;
                        malarvingar.y = -300;
                        malarvingar.rotation += 3;
                        setChildIndex(malarvingar, numChildren-1);
                }else{
                        malarvingar.y = -2222;
                        malarvingar.visible = false;
                }

                if(onetimekvarn == false){
                        setChildIndex(kvarnvingarna, numChildren-2);
                        onetimekvarn = true;
                }

                if(uppg_sackTaken == false && showSack == true && showSackOnce ==
                false){
                        uppg_sack.x = 434;
                        bubvisste.sfxbubpling();
                        showSackOnce = true;
                }
                if(cpanel.uppdragActive == false && cpanel.vissteActive == false &&
                cpanel.mapActive == false && cpanel.onOffActive == false &&
                cpanel.helpActive == false && lagaLager == true){

                        if(tomten.activeTomte == true){
                        setChildIndex(tomten, numChildren-1);

                        }else tomteknut.x = 2222;

                        setChildIndex(fjarilen, numChildren-1);
                        setChildIndex(uppg_sack, numChildren-1);
                        setChildIndex(hero, numChildren-1);
                }
                if(sfxBirdsComplete == true)fagelljud();

                if(cpanel.uppdragActive == false && cpanel.vissteActive == false &&
                cpanel.mapActive == false && cpanel.onOffActive == false &&
                cpanel.helpActive == false && removekvarndorrsMask == false){
                        btn_kvarndorrMask.x = 383;
                        btn_kvarndorrMask.y = 293;
                        btn_kvarndorr.x = 383;
                        setChildIndex(btn_kvarndorrMask, numChildren-1);

                }else{
                        btn_kvarndorrMask.x = 2222;
                        btn_kvarndorr.x = 2222;
                }

                if(cpanel.uppdragActive == false && cpanel.mapActive == false &&
                cpanel.vissteActive == false && cpanel.helpActive == false){

                        setChildIndex(arrowSignLeft, numChildren-1);
                }

        }else{
                kvarnvingarna.visible = false;
                uppg_sack.x = 2222;
                btn_kvarndorrMask.x = 2222;
                malarvingar.y = -2222;
```

```
            }

            if(partofMap != 5 && partofMap != 4){
                    tomteknut.x = -333;
            }

            if(cpanel.xuppdrag.hitTestPoint(mouseX, mouseY, false)){
                    xuppdragIsHit = true;
            }else xuppdragIsHit = false;


            if(buttonEsc == true){
                    if(maria.harPratatMaria == true){
                            stopHero();
                    }
                    popEnd();

            }

            if(allaUppUtforda == true && allaUppOnce == true){

                    allaUppOnce = false;

                    maria.maraRemove();
                    maria.gotoAndStop(3);

                    if(whatLanguage == 2){
                            popalla.gotoAndStop(2);
                    }else{
                            popalla.gotoAndStop(1);
                    }

                    popAlla();

            }

            if(lagaLager == false){
                    fjarilen.activeFjaril = false;
                    tomten.activeTomte = false;

            }

    }

    private function popAlla(){

            setChildIndex(popalla, numChildren-1);

            //Pop-up uppgift utförd.
            if(sfx_uppgutfComplete == true && ljudAv == false){
                    uppgutfljud();
            }

            Tweener.addTween(popalla, {y:50, time:0.4, transition:"easeOutSine",
            onComplete:function() {
            popstay();
            }});

            function popstay(){
            Tweener.addTween(tweendummy2, {y:128, time:2, transition:"easeOutSine",
            onComplete:function() {
                    popback();
                    }});
            }
            function popback(){
                    Tweener.addTween(popalla, {y:747, time:0.4,
            transition:"easeOutSine"});
            }
            }
```

```
//Starta avslutningen.
public function startParty(){

        Tweener.addTween(tweendummy, {x:2222, y:0, time:3, transition:"linear",
        onComplete:function() {
        party();
        }});

        function party(){

                cpanel.hideKnappar();

                maria.gotoAndStop(1);
                stopHero();
                lagaLager = false;

                partybild2.alpha = 0;
                partybild3.alpha = 0;

                partybild.alpha = 0;
                setChildIndex(partybild, numChildren-1);
                partybild.x = 0;
                Tweener.addTween(partybild, {alpha:1, time:0.6,
                transition:"linear"});

                setChildIndex(partybild2, numChildren-1);
                setChildIndex(partybild3, numChildren-1);

                setChildIndex(smedbror, numChildren-1);
                smedbror.x = partybild.x + 484;

                arrowSignLeft.walkout = "no";
                doNotWalk();

                if(sfxAmbientComplete == false)endambientljud();

                Tweener.addTween(tweendummy, {x:2222, y:0, time:3,
                transition:"linear", onComplete:function() {
                partyknappar();
                }});

        }
}

//Funktion för Rollfiguren Smedis Marias bror.
private function clicksmedbror(evt:MouseEvent):void{
        switch(smedbrorState){
                case 0:
                if(whatLanguage == 2){
                        smedbror.gotoAndStop(5);
                }else{
                        smedbror.gotoAndStop(2);
                }
                smedbrorState = 1;
                break;
                case 1:
                if(whatLanguage == 2){
                        smedbror.gotoAndStop(6);
                }else{
                        smedbror.gotoAndStop(3);
                }
                smedbrorState = 2;
                ljudfanfare();

                break;
                case 2:
                if(whatLanguage == 2){
                        smedbror.gotoAndStop(7);
                }else{
```

```
                        smedbror.gotoAndStop(4);
                }
                smedbror.buttonMode = false;
                partyFade();
                break;
        }
}

//Funktion för fade out effekter i slutet av avslutningsfesten.
private function partyFade(){
        partybild2.x = 0;
        partybild3.x = 0;

        Tweener.addTween(tweendummy, {y:200, time:0.4, transition:"easeOutSine",
        onComplete:function() {
        feastfade1();
        }});

        function feastfade1(){
        Tweener.addTween(partybild2, {alpha:1, time:2, transition:"linear",
        onComplete:function() {
                feastfade2();
                }});
        }
        function feastfade2(){
        Tweener.addTween(tweendummy, {y:200, time:1, transition:"easeOutSine",
        onComplete:function() {
                feastfade3();
                }});
        }
        function feastfade3(){
        Tweener.addTween(partybild3, {alpha:1, time:2, transition:"linear",
        onComplete:function() {
                        feastfade4();
                }});
        }
        function feastfade4(){
        Tweener.addTween(tweendummy, {y:300, time:4, transition:"easeOutSine",
        onComplete:function() {

                        fragesport.x = 0;
                        setChildIndex(fragesport, numChildren-1);
                        fragesport.fragesport();

                }});
        }

}

//Pop-up "Vill du avsluta spelet?"
private function popEnd(){
        buttonEsc = false;
        lagaLager = false;
        loadingmaask.x = 0;
        setChildIndex(loadingmaask, numChildren-1);

        setChildIndex(avslutaBg, numChildren-1);

        setChildIndex(btn_yes, numChildren-1);
        btn_yes.addEventListener(MouseEvent.CLICK,clickYes);
        btn_yes.buttonMode = true;
        btn_yes.gotoAndStop(2);

        setChildIndex(btn_no, numChildren-1);
        btn_no.addEventListener(MouseEvent.CLICK,clickNo);
        btn_no.buttonMode = true;
        btn_no.gotoAndStop(1);

        btn_yes.x = 400;
```

```
            btn_no.x = btn_yes.x + 47;
            avslutaBg.x = btn_yes.x - 10;

            setChildIndex(txt_avsluta, numChildren-1);
            txt_avsluta.x = 275;

            if(whatLanguage == 2){
                    txt_avsluta.text = "Tahdotko lopettaa?";
            }else{
                    txt_avsluta.text = "Vill du avsluta spelet?";
            }
    }
    //Vill inte avsluta.
    private function clickNo(evt:MouseEvent):void{

            arrowSignLeft.walkout = "no";
            doNotWalk();
            avslutaBg.x = 2222;
            btn_yes.x = 2222;
            btn_no.x = 2222;
            loadingmaask.x = 2222;
            txt_avsluta.x = 2222;
            trace("no");
            btn_yes.removeEventListener(MouseEvent.CLICK,clickYes);
            btn_no.removeEventListener(MouseEvent.CLICK,clickNo);
            lagaLager = true;
            if(maria.harPratatMaria == true){
                            startHero();
                            ReallydoNotWalk();
            }

    }
    //Vill avsluta.
    private function clickYes(evt:MouseEvent):void{
            arrowSignLeft.walkout = "no";
            doNotWalk();
            trace("yes");
            btn_yes.removeEventListener(MouseEvent.CLICK,clickYes);
            btn_no.removeEventListener(MouseEvent.CLICK,clickNo);
            endKarta();

    }

    //Avsluta kvarnfilmen.
    public function endMalaren(){
            kvarnfilmActive = false;
    }

    //Avsluta klassen karta, ta bort laddare och EventListeners för att frigöra minne.
    public function endKarta(){
            stopHero();
            removeEventListener(Event.ENTER_FRAME,kartaLoop);

            griffelloadern.loaderGriffel.clear(); //Ta bort griffeltavlans laddare,
            (annars säger BulkLoader "has already been created" om spelaren öppnar
            griffeltavlan mer än en gång).

            //Radera denna inladdade instans från själva laddaren (parent).
            MovieClip(parent).killLoadedClip();

            skoldorr.removeEventListener(MouseEvent.CLICK,clickSkoldorr);
            skoldorr.removeEventListener(MouseEvent.MOUSE_OVER,pekaSkoldorr);
            skoldorr.removeEventListener(MouseEvent.MOUSE_OUT,dissaSkoldorr);
            popupskola.btn_film.removeEventListener(MouseEvent.CLICK,clickbtn_film);
            popupskola.btn_film.removeEventListener(MouseEvent.MOUSE_OVER,pekabtn_film);
            popupskola.btn_film.removeEventListener(MouseEvent.MOUSE_OUT,dissabtn_film);
            popupskola.btn_griff.removeEventListener(MouseEvent.CLICK,clickbtn_griff);

popupskola.btn_griff.removeEventListener(MouseEvent.MOUSE_OVER,pekabtn_griff);
```

```
popupskola.btn_griff.removeEventListener(MouseEvent.MOUSE_OUT,dissabtn_griff);

                BulkLoader.removeAllLoaders();

        }

        //Funktion för initialisering.
        public function initExterior(){

                addChild(vissteduatt);
                vissteduatt.x = 2222;
                vissteduatt.y = 0;
                vissteduatt.howManyActive = 2;
                vissteduatt.didyouknow();

                txtLoading.x = 2222;
                lbar.x = 2222;

                if(sfxAmbientComplete == true)ambientljud();

                partofMap = 1;

                menyKvarn.menyKvarn();

                uppg_sack.buttonMode = true;
                uppg_sack.addEventListener(MouseEvent.CLICK,clickuppg_sack);

                gunmask.x = 132;
                setChildIndex(gunmask, numChildren-1);
                gunmask.addEventListener(MouseEvent.CLICK,clickGunmask);
                gunmask.addEventListener (MouseEvent.MOUSE_OVER,pekaGunmask);
                gunmask.addEventListener (MouseEvent.MOUSE_OUT,dissaGunmask);
                gunmask.buttonMode = true;
                gunmask.gotoAndStop(1);

                btn_kvarndorrMask.addEventListener(MouseEvent.CLICK,clickKvarndorr);
                btn_kvarndorrMask.addEventListener (MouseEvent.MOUSE_OVER,pekaKvarndorr);
                btn_kvarndorrMask.addEventListener (MouseEvent.MOUSE_OUT,dissaKvarndorr);
                btn_kvarndorrMask.buttonMode = true;
                btn_kvarndorr.buttonMode = true;
                btn_kvarndorr.mouseChildren = false;
                btn_kvarndorr.gotoAndStop(1);
                btn_kvarndorrMask.gotoAndStop(1);
                btn_kvarndorrMask.mouseChildren = false;

                btn_smeddorr.addEventListener(MouseEvent.CLICK,clickSmeddorr);
                btn_smeddorr.addEventListener (MouseEvent.MOUSE_OVER,pekaSmeddorr);
                btn_smeddorr.addEventListener (MouseEvent.MOUSE_OUT,dissaSmeddorr);
                btn_smeddorr.buttonMode = true;
                btn_smeddorr.gotoAndStop(1);

                btn_handeldorr.addEventListener(MouseEvent.CLICK,clickbtn_handeldorr);
                btn_handeldorr.addEventListener (MouseEvent.MOUSE_OVER,pekabtn_handeldorr);
                btn_handeldorr.addEventListener (MouseEvent.MOUSE_OUT,dissabtn_handeldorr);
                btn_handeldorr.buttonMode = true;
                btn_handeldorr.gotoAndStop(1);

                btn_bonddorr.addEventListener(MouseEvent.CLICK,clickBonddorr);
                btn_bonddorr.addEventListener (MouseEvent.MOUSE_OVER,pekaBonddorr);
                btn_bonddorr.addEventListener (MouseEvent.MOUSE_OUT,dissaBonddorr);
                btn_bonddorr.buttonMode = true;
                btn_bonddorr.gotoAndStop(1);

                btn_klokdorr.addEventListener(MouseEvent.CLICK,clickbtn_klokdorr);
                btn_klokdorr.addEventListener (MouseEvent.MOUSE_OVER,pekabtn_klokdorr);
                btn_klokdorr.addEventListener (MouseEvent.MOUSE_OUT,dissabtn_klokdorr);
                btn_klokdorr.buttonMode = true;
                btn_klokdorr.gotoAndStop(1);
```

```
skoldorr.addEventListener(MouseEvent.CLICK,clickSkoldorr);
skoldorr.addEventListener (MouseEvent.MOUSE_OVER,pekaSkoldorr);
skoldorr.addEventListener (MouseEvent.MOUSE_OUT,dissaSkoldorr);
skoldorr.buttonMode = true;
skoldorr.gotoAndStop(1);
skoldorr.x = 2222;

popupskola.btn_film.addEventListener(MouseEvent.CLICK,clickbtn_film);
popupskola.btn_film.addEventListener (MouseEvent.MOUSE_OVER,pekabtn_film);
popupskola.btn_film.addEventListener (MouseEvent.MOUSE_OUT,dissabtn_film);
popupskola.btn_film.buttonMode = true;

popupskola.btn_griff.addEventListener(MouseEvent.CLICK,clickbtn_griff);
popupskola.btn_griff.addEventListener (MouseEvent.MOUSE_OVER,pekabtn_griff);
popupskola.btn_griff.addEventListener (MouseEvent.MOUSE_OUT,dissabtn_griff);
popupskola.btn_griff.buttonMode = true;

bgskolfilm.btnbackskol.addEventListener(MouseEvent.CLICK,clickbtn_backskol);
bgskolfilm.btnbackskol.addEventListener
(MouseEvent.MOUSE_OVER,pekabtn_backskol);
bgskolfilm.btnbackskol.addEventListener
(MouseEvent.MOUSE_OUT,dissabtn_backskol);
bgskolfilm.btnbackskol.buttonMode = true;

pil_N_E.addEventListener(MouseEvent.CLICK,clickpil_N_E);
pil_N_E.addEventListener (MouseEvent.MOUSE_OVER,pekapil_N_E);
pil_N_E.addEventListener (MouseEvent.MOUSE_OUT,dissapil_N_E);
pil_N_E.buttonMode = true;
pil_N_E.gotoAndStop(1);

popupskola.btnxskolpop.addEventListener(MouseEvent.CLICK,clickbtnxskolpop);
popupskola.btnxskolpop.buttonMode = true;

smedbror.addEventListener(MouseEvent.CLICK,clicksmedbror);
smedbror.buttonMode = true;
smedbror.gotoAndStop(1);

popupskola.x = 810;
popupskola.y = -555;
addChild(popupskola);

popupskola.btn_film.gotoAndStop(1);
popupskola.btn_griff.gotoAndStop(1);
bgskolfilm.btnbackskol.gotoAndStop(1);

// Skapa bakgrundsbytare.
bg = new BgSwitcher();
addChild(bg);
bg.gotoAndStop(1);
bg.x = -180;
bg.y = 0;

hus1.alpha =  0;
hus2.alpha =  0;
hus3.alpha =  0;
hus4.alpha =  0;

hus1.addEventListener (MouseEvent.MOUSE_OVER,pekabtn_hus1);
hus2.addEventListener (MouseEvent.MOUSE_OVER,pekabtn_hus2);
hus3.addEventListener (MouseEvent.MOUSE_OVER,pekabtn_hus3);
hus4.addEventListener (MouseEvent.MOUSE_OVER,pekabtn_hus4);

hus1.addEventListener (MouseEvent.MOUSE_OUT,dissabtn_hus1);
hus2.addEventListener (MouseEvent.MOUSE_OUT,dissabtn_hus1);
hus3.addEventListener (MouseEvent.MOUSE_OUT,dissabtn_hus1);
hus4.addEventListener (MouseEvent.MOUSE_OUT,dissabtn_hus1);

hus1.buttonMode = true;
```

```
hus2.buttonMode = true;
hus3.buttonMode = true;
hus4.buttonMode = true;

setChildIndex(boundary_handeln, numChildren-1);
setChildIndex(boundary_bondgarden, numChildren-1);
setChildIndex(boundary_smeden, numChildren-1);

//Gränser var spelfiguren kan röra sig.
boundary_handeln.alpha = 0;
boundary_handeln.y = -555;
boundaryArray.push(boundary_handeln);
boundary_bondgarden.alpha = 0;
boundary_bondgarden.y = -555;
boundaryArray.push(boundary_bondgarden);
boundary_smeden.alpha = 0;
boundary_smeden.y = 0;
boundaryArray.push(boundary_smeden);

instruktioner.instruktioner();
instruktioner.helpText();

//Skapa spelfigur och placera på scenen.
hero = new Hero();
addChild(hero);
hero.gotoAndStop(1);
hero.x = herostartX;
hero.y = herostartY;
hero.boundaries();

//Skapa spelfiguren Smedis Maria och placera på scenen.
addChild(maria);
maria.gotoAndStop(1);
maria.smedismaria();
maria.x = 610;
maria.y = 258;

addChild(chiq);
chiq.gotoAndStop(1);
chiq.chicken();
chiq.startx = 700;
chiq.starty = 420;
chiq.endx = 500;
chiq.endy = 440;
chiq.x = chiq.startx;
chiq.y = chiq.starty ;
chiq.addEventListener(MouseEvent.CLICK, clickChick);

//Skapa spelmenyn och placera på scenen.
cpanel = new ControlPanel();
addChild(cpanel);
cpanel.x = 0;
cpanel.y = 500;
cpanel.Knapparna();
cpanel.controlPanel();

//Skapa navigeringspilarna och placera på scenen.
arrowSignLeft = new Arrowsign();
addChild(arrowSignLeft);
arrowSignLeft.arrowsign();

btn_smeddorr.x = 833;
setChildIndex(btn_smeddorr, numChildren-1);

inituppdrag();

bubvisste.sprak = whatLanguage;
bubvisste.bubvislanguage();
bubvisste.bubvisste();
```

```
        if(whatLanguage == 2){

                poputford.txt_poputford.text = "Tehtävä suoritettu!";
        }else{
                poputford.txt_poputford.text = "Uppdrag utfört!";
        }

        fjarilen.fjaril();
        tomten.tomte();

}

//Funktion för Pop-up fönster "visste du att" (information om livet förr).
public function popupVisste(){

        Tweener.addTween(bubvisste, {y:23, time:0.4, transition:"easeOutSine"});
        bubvisste.sfxbubpling(); //Spela pling ljudeffekt.
        setChildIndex(bubvisste, numChildren-1);

        Tweener.addTween(tweendummy, {x:2222, y:0, time:5, transition:"linear",
        onComplete:function() {
        hidePop();
        }});

        function hidePop(){
        Tweener.addTween(bubvisste, {y:647, time:0.4, transition:"easeOutSine"});
        }

}

//Visa själva "visste du att" interfacet.
public function moveVisste(){

        lagaLager = false; //Sluta placera objekt överst i z lagren.
        stopHero(); //Stoppa spelfiguren.

        cpanel.vissteActive = true; //Aktivera "visste du att" interfacet.

        vissteduatt.x = 0; //Placera det så det syns på scenen.
        setChildIndex(vissteduatt, numChildren-1); //Lägg interfacet överst i z
        ordningen.
        bubvisste.y = 748;

}

//Göm själva "visste du att" interfacet.
public function hideVisste(){

        lagaLager = true;
        if(maria.harPratatMaria == true)startHero();
        cpanel.vissteActive = false;
        Tweener.addTween(tweendummy, {x:2222, y:0, time:1, transition:"linear",
        onComplete:function() {
        walkagain();
        }});

        function walkagain(){
                donotwalk = false;
        }

}

//Initialisering av uppdragslistan "vilka uppdrag som är utförda" och bockarna
(kryssena).
private function inituppdrag() {
        for(var i = 0; i < howmanyUppdrag; i++){
                tblUppdrag[i] = false;
        }
```

```
        for(var j = 0; j < howmanyUppdrag; j++){
                var nyBock:UppBock = new UppBock;
                addChild(nyBock);
                tblUppbockar.push(nyBock);

                tblUppbockar[j].x = 2222;
                tblUppbockar[j].y = 150 + (j * 40);
                tblUppbockar[j].visible = false;
        }


}

//Visa bockarna när uppdragsboken visas.
public function showUpBockar(){
        for(var n = 0; n < howmanyUppdrag; n++){
                tblUppbockar[n].x = 540;
                setChildIndex(tblUppbockar[n], numChildren-1);
                checkuppdrag();
        }
}

//För att kolla vilka uppdrag som är utförda och visa bockar därefter.
public function checkuppdrag(){

        var truers:int = 0;

        for(var k = 0; k < howmanyUppdrag; k++){
                if(tblUppdrag[k] == true) truers++;
        }

        if(truers >= howmanyUppdrag){
                trace("Alla uppdrag utförda");
                allaUppUtforda = true;
        }

        for(var l = 0; l < howmanyUppdrag; l++){
                if(tblUppdrag[l] == true){
                        tblUppbockar[l].visible = true;
                }
        }

}
//Visa spelinstruktionerna.
public function showHelp(){

        lagaLager = false;
        instruktioner.x = 0;
        setChildIndex(instruktioner, numChildren-1);

}
//Funktion för när skolan avslutas och spelaren tas tillbaka till exteriörerna.
private function clickbtn_backskol(evt:MouseEvent):void{
        // back to karta
        skolfilmen.autoRewind = true;
        skolfilmen.stop();
        skolfilmen.x = 2222;
        bgskolfilm.x = 2222;
        startHero();
        endSkolGrif();

}

//Markören pekar på knappen.
private function pekabtn_backskol(evt:MouseEvent):void{
        bgskolfilm.btnbackskol.gotoAndStop(2);
}

//Markören pekar ej mera på knappen.
private function dissabtn_backskol(evt:MouseEvent):void{
```

```
		bgskolfilm.btnbackskol.gotoAndStop(1);
	}

	//Funktion för kvarndörren/spelfiguren mjölnaren (knapp).
	private function clickKvarndorr(evt:MouseEvent):void{
		if(kvarnmode == 9){
			kvarnmode = 0;

			setChildIndex(hero, numChildren-1);

				if(onceSoundKD == true){
					onceSoundKD = false;

					Mjoltimer();

				}

				if(popupOnceKvarnen == true){
					popupOnceKvarnen = false;
					popupVisste();

					vissteduatt.howManyActive = 7;
					vissteduatt.showOnlyActiveMenu();
					vissteduatt.thing_id = vissteduatt.xmlTableStart + 6;
					vissteduatt.vilkenText();

					btn_kvarndorr.gotoAndStop(3);
					btn_kvarndorrMask.gotoAndStop(7);

					if(whatLanguage == 2){
						btn_kvarndorrMask.mjolprat.htmlText = "<b>Hei!
					Tule katsomaan miten mylly toimii sillä aikaa kun
					jauhaan sinulle säkillisen jauhoja.</b>";
					}else{
						btn_kvarndorrMask.mjolprat.htmlText = "<b>Hej!
					Kom in och se hur kvarnen fungerar medan jag malar en
					säck mjöl till dig.</b>";
					}

				}

		}else{
			lagaLager = false;
			if(kvarnmode == 0){
				kvarnmode = 1;

				if(whatLanguage == 2){
					btn_kvarndorrMask.mjolprat.htmlText = "<b>Ole hyvä,
				tässä on jauhosäkkisi. Tervetuloa uudelleen!</b>";
				}else{
					btn_kvarndorrMask.mjolprat.htmlText = "<b>Varsågod
				här är din mjölsäck. Välkommen åter!</b>";
				}
				btn_kvarndorr.gotoAndStop(5);
			}

			arrowSignLeft.walkout = "no";
			doNotWalk();

			lagaLager = false;
			removekvarndorrsMask = true;
			btn_kvarndorrMask.x = 2222;
			kvarnfilm.source =
			"http://stundars.img.fi/video/KvarnenMal_NoVings.flv";
			kvarnfilm.play();
			bgKvarnskar.x = 0;
			bgKvarnskar.y = 0;
			kvarnfilm.x = 0;
			kvarnfilm.y = 0;
```

```
                kvarnfilmActive = true;
                setChildIndex(bgKvarnskar, numChildren-1);
                setChildIndex(kvarnfilm, numChildren-1);
                setChildIndex(menyKvarn, numChildren-1);
                setChildIndex(btnbackkvarn, numChildren-1);
                stopHero();
                menyKvarn.placeramenykvarn();

                if(sfxBirdsComplete == false)endfagelljud();
                kvarnfilm.addEventListener(VideoEvent.COMPLETE, onFLVComplete);
        }
}

private function onFLVComplete(event:VideoEvent):void {
        event.target.play();
}
public function removeFLVComplete(){
        kvarnfilm.removeEventListener(VideoEvent.COMPLETE, onFLVComplete);
}

//*** *** *** knappar *** *** ***

//Markören pekar på knappen = (over).
//Markören pekar ej mera på knappen = (out).

//Markören pekar på den osynliga maskningen på slangbellan (knapp).
private function pekaGunmask(evt:MouseEvent):void{
        if(maria.harPratatMaria == true && arrowSignLeft.walkout == "no"){
                if(whatLanguage == 2){
                        gunmask.gotoAndStop(3);
                }else {
                        gunmask.gotoAndStop(2);
                }
        }
}

//Markören pekar ej mera på slangbellan.
private function dissaGunmask(evt:MouseEvent):void{
        gunmask.gotoAndStop(1);
}

//Spelaren klickar på slangbellan.
private function clickGunmask(evt:MouseEvent):void{

        if(maria.harPratatMaria == true && arrowSignLeft.walkout == "no"){

                lagaLager = false; // Så inte figurerna ploppar upp i skjutspelet.
                stopHero(); // Så spelaren inte kan gå omkring med figuren under
                skjutspelet.
                loadShoot();  //Ladda in skjutspelet.

        }
}

//Funktion för att aktivera ordnandet av objektens z ordning från andra klasser.
public function layers(){
        lagaLager = true;
}

//Initialisering av ny timer för att animera mjölnaren.
private function Mjoltimer(){
        //Var tusende millisekund besök funktionen mjolTick.
        mjoltimer = new Timer(500, 0);
        mjoltimer.addEventListener(TimerEvent.TIMER, mjolTick);
        mjoltimer.start();
}

//Animering av mjölnaren.
private function mjolTick(event:TimerEvent){
```

```
            if(kvarnmode == 0){

                    if(switchPrat == false){
                            switchPrat = true;
                            btn_kvarndorr.gotoAndStop(3);
                    }else{
                            switchPrat = false;;
                            btn_kvarndorr.gotoAndStop(4);
                    }
            }else if(kvarnmode == 1){
                    if(switchPrat == false){
                            switchPrat = true;
                            btn_kvarndorr.gotoAndStop(5);
                    }else{
                            switchPrat = false;;
                            btn_kvarndorr.gotoAndStop(6);
                    }

            }else{

            }

    }

    private var forstTiKvarn:Boolean = true;

    //Markören pekar på kvarndörren (knapp).
    private function pekaKvarndorr(evt:MouseEvent):void{

            switch(kvarnmode){
                    case 0:


                    break;
                    case 1:
                            btn_kvarndorr.gotoAndStop(5);

                    break;
                    case 2:
                            if(sfx_doorComplete == true)dorrljud();
                            btn_kvarndorr.gotoAndStop(2);
                            btn_kvarndorrMask.gotoAndStop(1);
                    break;
                    case 9:
                            if(forstTiKvarn == true){
                            forstTiKvarn = false;
                            if(sfx_doorComplete == true)dorrljud();
                            }
                            btn_kvarndorr.gotoAndStop(11);
                    break;
            }

    }

    //Markören pekar ej mera på kvarndörren.
    private function dissaKvarndorr(evt:MouseEvent):void{
            switch(kvarnmode){
                    case 0:
                            btn_kvarndorr.gotoAndStop(3);
                            btn_kvarndorrMask.gotoAndStop(7);
                    break;
                    case 1:
                            btn_kvarndorr.gotoAndStop(5);

                    break;
                    case 2:
                            btn_kvarndorr.gotoAndStop(1);
                            btn_kvarndorrMask.gotoAndStop(1);
                    break;
```

```
        }
}

//Spelaren klickar på smedens dörr (knapp).
private function clickSmeddorr(evt:MouseEvent):void{
        if(maria.harPratatMaria == true){
                arrowSignLeft.walkout = "no";
                doNotWalk();
                //Ladda smedens interiör.
                if(sfxAmbientComplete == false)endambientljud();
                lagaLager = false;
                stopHero();
                loadSmed();
        }
}

//Markören pekar på smedens dörr.
private function pekaSmeddorr(evt:MouseEvent):void{
        btn_smeddorr.gotoAndStop(2);
        if(cpanel.mapActive == false){setChildIndex(hero, numChildren-1);}
        if(sfx_doorComplete == true)dorrljud();

}

//Markören pekar ej mera smedens dörr.
private function dissaSmeddorr(evt:MouseEvent):void{
        btn_smeddorr.gotoAndStop(1);
}

// Spelaren klickar på handelns dörr.
private function clickbtn_handeldorr(evt:MouseEvent):void{

        arrowSignLeft.walkout = "no";
        doNotWalk();
        if(sfxBirdsComplete == false)endfagelljud();
        lagaLager = false;
        stopHero();
        loadMus();

}

//När "visste du att" laddats in.
public function didLoadReady(){
        trace("didLoadReady");
        //Två aktiva från början.
        vissteduatt.howManyActive = 2;
        //Visa endast de som aktiverats.
        vissteduatt.showOnlyActiveMenu();
        vissteduatt.thing_id = vissteduatt.xmlTableStart + 1;
        //Kalla funktionen som går igenom XML data
        vissteduatt.vilkenText();
}

//Markören pekar på handelns dörr (knapp).
private function pekabtn_handeldorr(evt:MouseEvent):void{
        if(whatLanguage == 2){
                btn_handeldorr.gotoAndStop(3);
        }else{
                btn_handeldorr.gotoAndStop(2);
        }
        if(cpanel.mapActive == false){setChildIndex(hero, numChildren-1);}
        if(sfx_doorComplete == true)dorrljud();

        if(popupOnceHandel == true){
                popupOnceHandel = false;
                popupVisste();

                vissteduatt.howManyActive = 3;
                vissteduatt.showOnlyActiveMenu();
                vissteduatt.thing_id = vissteduatt.xmlTableStart + 2;
```

```
                visseduatt.vilkenText();

        }

}

//Markören pekar ej mera på handelns dörr.
private function dissabtn_handeldorr(evt:MouseEvent):void{
        btn_handeldorr.gotoAndStop(1);
}

//Spelaren klickar på spelfiguren kloka gumman (knapp).
private function clickbtn_klokdorr(evt:MouseEvent):void{
        lagaLager = false;
        arrowSignLeft.walkout = "no";
        doNotWalk();
        if(sfxBirdsComplete == false)endfagelljud();
        lagaLager = false;
        stopHero();
        loadKlok();
}

//Markören pekar på kloka gumman.
private function pekabtn_klokdorr(evt:MouseEvent):void{
        btn_klokdorr.gotoAndStop(2);
        if(cpanel.mapActive == false){setChildIndex(hero, numChildren-1);}

        if(whatLanguage == 2){
                btn_klokdorr.kloktext.htmlText = "<b>Pelaa kanssani ja hanki tietoa
                lääkkeistäni.</b>"
        }else{
                btn_klokdorr.kloktext.htmlText = "<b>Spela ett spel med mig och
                erhåll kunskap om mina läkemedel.</b>"
        }

        if(popupOnceKloka == true){
                popupOnceKloka = false;
                popupVisste();

                visseduatt.howManyActive = 8;
                visseduatt.showOnlyActiveMenu();
                visseduatt.thing_id = visseduatt.xmlTableStart + 7;
                visseduatt.vilkenText();

        }

}
//Markören pekar ej mera på kloka gumman.
private function dissabtn_klokdorr(evt:MouseEvent):void{
        btn_klokdorr.gotoAndStop(1);
}

//Spelaren klickar på navigeringspilen (knapp).
private function clickpil_N_E(evt:MouseEvent):void{
        if(maria.bubbla.bubbelnummer != 7){
                hero.leftscreen = "button";
                exteriorChange();
        }
}

//Spelaren pekar på navigeringspilen.
private function pekapil_N_E(evt:MouseEvent):void{
        if(maria.harPratatMaria == true){
        pil_N_E.gotoAndStop(2);
        pil_N_E_Bool = true;

        if(whatLanguage == 2){
                //switch för att visa till vilken exteriör du kommer till från
                vilken
```

```
                        //1 = Smeden, 2 = Handeln och Skolan, 3  = Bondgården, 4 = Kvarnen,
                        5 = Kloka gummans stuga
                        switch(partofMap){
                                case 1:
                                arrowSignLeft.whichExterior = "Kyläkauppa ja koulu";
                                break;
                                case 2:
                                arrowSignLeft.whichExterior = "Maalaistalo";
                                break;
                                case 3:
                                arrowSignLeft.whichExterior = "Mylly";
                                break;
                                case 4:
                                arrowSignLeft.whichExterior = " ";
                                break;
                                case 5:
                                arrowSignLeft.whichExterior = "Seppä";
                                break;
                        }
                }else{

                        switch(partofMap){
                                case 1:
                                arrowSignLeft.whichExterior = "Handeln och Skolan";
                                break;
                                case 2:
                                arrowSignLeft.whichExterior = "Bondgården";
                                break;
                                case 3:
                                arrowSignLeft.whichExterior = "Kvarnen";
                                break;
                                case 4:
                                arrowSignLeft.whichExterior = " ";
                                break;
                                case 5:
                                arrowSignLeft.whichExterior = "Smeden";
                                break;
                        }

                }

                cpanel.txtNavigator.htmlText = "<b>"+arrowSignLeft.whichExterior+"</b>";

                }

        }

        //Markören pekar ej mera på navigeringspilen.
        private function dissapil_N_E(evt:MouseEvent):void{
                pil_N_E.gotoAndStop(1);
                arrowSignLeft.whichExterior = " ";
                pil_N_E_Bool = false;
                cpanel.txtNavigator.htmlText = "<b>"+arrowSignLeft.whichExterior+"</b>";

        }

        //Spelaren klickar på mjölsäcken.
        private function clickuppg_sack(evt:MouseEvent):void{

                arrowSignLeft.walkout = "no";
                doNotWalk();
                tblUppdrag[3] = true; //uppdrag 4 mjölsäken (tabellen börjar från 0 därför
                tblUppdrag[3])
                uppg_sackTaken = true;
                uppg_sack.x = 2222;
                popUtford();

                if(kvarnmode == 1){
                        kvarnmode = 2;
```

```
                mjoltimer.removeEventListener(TimerEvent.TIMER, mjolTick);
                btn_kvarndorr.gotoAndStop(1);
                btn_kvarndorrMask.gotoAndStop(1);
        }

}

//Funktion för pop-up fönstret "uppgift utförd".
private function popUtford(){

        setChildIndex(poputford, numChildren-1);

        //Pop-up uppgift utförd
        if(sfx_uppgutfComplete == true && ljudAv == false){
                uppgutfljud();
        }

        Tweener.addTween(poputford, {y:200, time:0.4, transition:"easeOutSine",
        onComplete:function() {
        popstay();
        }});

        function popstay(){
        Tweener.addTween(tweendummy, {y:128, time:2, transition:"easeOutSine",
        onComplete:function() {
                popback();
                }});
        }
        function popback(){
        Tweener.addTween(poputford, {y:747, time:0.4, transition:"easeOutSine"});
        }


}

//Spelaren klickar på bondgårdsdörren (knapp).
private function clickBonddorr(evt:MouseEvent):void{

        arrowSignLeft.walkout = "no";
        doNotWalk();
        if(sfxAmbientComplete == false)endambientljud();
        lagaLager = false;
        stopHero();
        loadBond();
}

//Markören pekar på bondgårdsdörren.
private function pekaBonddorr(evt:MouseEvent):void{
        if(whatLanguage == 2){
                btn_bonddorr.gotoAndStop(3);
        }else{
                btn_bonddorr.gotoAndStop(2);
        }

        if(cpanel.mapActive == false){setChildIndex(hero, numChildren-1);}
        if(sfx_doorComplete == true)dorrljud();

        if(popupOnceBond == true){
                popupOnceBond = false;
                popupVisste();

                vissteduatt.howManyActive = 5;
                vissteduatt.showOnlyActiveMenu();
                vissteduatt.thing_id = vissteduatt.xmlTableStart + 4;
                vissteduatt.vilkenText();

        }

}
```

```
//Markören pekar ej mera på bondgårdsdörren.
private function dissaBonddorr(evt:MouseEvent):void{
        btn_bonddorr.gotoAndStop(1);
}

//Spelaren klickar på skoldörren då den redan är aktiv.
private function clickbtnxskolpop(evt:MouseEvent):void{
        if(klickSkoldorr == true){
                popupskola.y = -555;
                klickSkoldorr = false;
                ReallydoNotWalk(); //För inte dit spelfiguren.
        }
}

//Spelaren klickar på skoldörren då den inte är aktiv.
private function clickSkoldorr(evt:MouseEvent):void{
        arrowSignLeft.walkout = "no";
        doNotWalk();

        if(klickSkoldorr == false){
                popupskola.y = 88;
                setChildIndex(popupskola, numChildren-1);
                klickSkoldorr = true;
        }else{
                popupskola.y = -555;
                klickSkoldorr = false;
        }
}

//Markören pekar på skoldörren.
private function pekaSkoldorr(evt:MouseEvent):void{
        skoldorr.gotoAndStop(2);
        if(cpanel.mapActive == false){setChildIndex(hero, numChildren-1);}
        if(sfx_doorComplete == true)dorrljud();

        if(popupOnceSkola == true){
                popupOnceSkola = false;
                popupVisste();

                vissteduatt.howManyActive = 4;
                vissteduatt.showOnlyActiveMenu();
                vissteduatt.thing_id = vissteduatt.xmlTableStart + 3;
                vissteduatt.vilkenText();

        }

}

//Markören pekar ej mera på skoldörren.
private function dissaSkoldorr(evt:MouseEvent):void{
        skoldorr.gotoAndStop(1);
}

//Spelaren klickar på knappen för griffeltavlan.
private function clickbtn_griff(evt:MouseEvent):void{
        lagaLager = false;
        harGriflat = true;
        arrowSignLeft.walkout = "no";
        doNotWalk();

        if(sfxBirdsComplete == false)endfagelljud();
        stopHero();
        loadGriffel();
}
//Spelaren pekar på knappen för griffeltavlan.
private function pekabtn_griff(evt:MouseEvent):void{
        popupskola.btn_griff.gotoAndStop(2);
}
//Spelaren pekar ej mer på knappen för griffeltavlan.
```

```
private function dissabtn_griff(evt:MouseEvent):void{
        popupskola.btn_griff.gotoAndStop(1);
}
//Spelaren klickar på knappen för skolfilmen.
private function clickbtn_film(evt:MouseEvent):void{
        lagaLager = false;
        harSkolfilm = true;
        arrowSignLeft.walkout = "no";
        doNotWalk();
        if(sfxBirdsComplete == false)endfagelljud();
        skolfilmen.source = "http://stundars.img.fi/video/skolfilmen.flv";
        skolfilmen.play();
        skolfilmen.x = 210;
        skolfilmen.y = 56;
        bgskolfilm.x = 0;
        bgskolfilm.y = 0;

        setChildIndex(bgskolfilm, numChildren-1);
        setChildIndex(skolfilmen, numChildren-1);
        stopHero();
}
//Spelaren pekar på knappen för skolfilmen.
private function pekabtn_film(evt:MouseEvent):void{
        popupskola.btn_film.gotoAndStop(2);
}
//Spelaren pekar ej mer på knappen för skolfilmen.
private function dissabtn_film(evt:MouseEvent):void{
        popupskola.btn_film.gotoAndStop(1);
}
//Spelaren klickar på hönan (knapp).
private function clickChick(evt:MouseEvent):void{
        arrowSignLeft.walkout = "no";
        doNotWalk();
        if(chiq.kackelComplete == true && ljudAv == false){
        chiq.kacklaren();
        }
}
//Husknapparna är osynliga masker.
//Markören pekar på lillstugan (knapp).
private function pekabtn_hus1(evt:MouseEvent):void{
        pekahus = true;
        if(whatLanguage == 2){
                cpanel.txtNavigator.htmlText = "<b>Pikkutupa</b>";
        }else{
                cpanel.txtNavigator.htmlText = "<b>Lillstugan</b>";
        }
}
//Markören pekar på bondstugan (knapp).
private function pekabtn_hus2(evt:MouseEvent):void{
        pekahus = true;
        if(whatLanguage == 2){
                cpanel.txtNavigator.htmlText = "<b>Maalaistalo</b>";
        }else{
                cpanel.txtNavigator.htmlText = "<b>Bondstugan</b>";
        }
}
//Markören pekar på loftsraden (knapp).
private function pekabtn_hus3(evt:MouseEvent):void{
        pekahus = true;
        if(whatLanguage == 2){
                cpanel.txtNavigator.htmlText = "<b>Luhti</b>";
        }else{
                cpanel.txtNavigator.htmlText = "<b>Loftsrad</b>";
        }
}
//Markören pekar på fähuset (knapp).
private function pekabtn_hus4(evt:MouseEvent):void{
        pekahus = true;
        if(whatLanguage == 2){
```

```
                cpanel.txtNavigator.htmlText = "<b>Navetta</b>";
        }else{
                cpanel.txtNavigator.htmlText = "<b>Fähus</b>";
        }
}
//Markören pekar ej mera på någon av husknapparna.
private function dissabtn_hus1(evt:MouseEvent):void{
        pekahus = false;
}

// Tangentbordets knappar (knapp tryckt).
public function keyDownFunction(event:KeyboardEvent){

                if (event.keyCode == 37) //Vänsterpil
                {
                leftArrow = true;
                } else if (event.keyCode == 39) //Högerpil
                {
                rightArrow = true;
                } else if (event.keyCode == 38) //Pil uppåt
                {
                upArrow = true;
                } else if (event.keyCode == 40) //Pil neråt
                {
                downArrow = true;
                }else if (event.keyCode == 27) //Escape
                {
                        buttonEsc = true;
                }else if (event.keyCode == 82) //R för testning av olika saker.
                {
                        keyR = true;
                        // Testfunktioner eller variabler här.

                }
}

// Tangentbordsknapp ej mera nertryckt.
public function keyUpFunction(event:KeyboardEvent){

                if (event.keyCode == 37)
                {
                        leftArrow = false;
                } else if (event.keyCode == 39)
                {
                        rightArrow = false;
                } else if (event.keyCode == 38)
                {
                        upArrow = false;
                } else if (event.keyCode == 40)
                {
                        downArrow = false;
                }else if (event.keyCode == 27)
                {
                        buttonEsc = false;
                }else if (event.keyCode == 82)
                {
                        keyR = false;
                }
}

private var changeOnce:Boolean = true;

//Funktion för start av ändring av exteriör (t.ex. spelaren går från smedens
exteriör till handelns exteriör). Timern görs enkelt med tweener och en dummygrafik
flyttas utanför spelrutan i en sekund och då kallas den riktiga funktionen för
ändring realchange().
public function exteriorChange(){

        if(changeOnce == true){
```

```
                changeOnce = false;
                trace("exteriorChange");
                stopHero();
                lagaLager = false;
                setChildIndex(loadingmaask, numChildren-1);
                loadingmaask.x = 0;
                fjarilen.activeFjaril = false;
                tomten.activeTomte = false;

                Tweener.addTween(this, {alpha:0, time:1});

                Tweener.addTween(tweendummy2, {x:2222, y:0, time:1,
                transition:"linear", onComplete:function() {
                trace("To change");
                change();
                }});

                function change(){
                trace("To realchange");
                        realchange();
                }
        }
}

//Den riktiga ändringen av exteriör efter en kort timer (en sekund).
private function realchange(){

        trace("realchange");
        //Reset spelfigurens position.
        herostartX = 348;
        herostartY = 400;

        //Stoppa ljudeffekter.
        if(sfxAmbientComplete == false)endambientljud();
        if(sfxBirdsComplete == false)endfagelljud();

        //Ändra bakgrund
        changeBg();

        pil_N_E.x = 2222;

        //***** Reset av de olika exteriörerna *****//
        if(partofMap == 1){
                boundary_smeden.y = 0;
                maria.x = 610;
                btn_smeddorr.x = 833;
                setChildIndex(btn_smeddorr, numChildren-1);
                setChildIndex(tomten, numChildren-1);
                setChildIndex(fjarilen, numChildren-1);
                gunmask.x = 132;
                setChildIndex(gunmask, numChildren-1);
                setChildIndex(hero, numChildren-1);
                if(sfxAmbientComplete == true)ambientljud();
                setChildIndex(pil_N_E, numChildren-1);
                pil_N_E.x = 700;
        }else{
                boundary_smeden.y = -555;
                maria.x = 2222; //Göm maria om exteriören inte är 1 (smedens stuga).
                btn_smeddorr.x = 2222;
                gunmask.x = 2222;

        }

        if(partofMap == 2){
                setChildIndex(tomten, numChildren-1);
                setChildIndex(fjarilen, numChildren-1);
                boundary_handeln.y = 0;
                skoldorr.y = 243;
                setChildIndex(skoldorr, numChildren-1);
```

```
            setChildIndex(btn_handeldorr, numChildren-1);
            setChildIndex(hero, numChildren-1);
            if(sfxBirdsComplete == true)fagelljud();
    }else {
            boundary_handeln.y = -555;
            skoldorr.y = -555;
            popupskola.y = -555;
    }

    if(partofMap == 3){
            setChildIndex(tomten, numChildren-1);
            setChildIndex(fjarilen, numChildren-1);
            boundary_bondgarden.y = 199;
            btn_bonddorr.x = 482;
            setChildIndex(btn_bonddorr, numChildren-1);
            if(sfxAmbientComplete == true){ambientljud();}

            hus1.x = 2;
            hus1.y = 5;
            hus2.x = 163;
            hus2.y = 65;
            hus3.x = 681;
            hus3.y = 122;
            hus4.x = 1378;
            hus4.y = 83;
            setChildIndex(hus1, numChildren-1);
            setChildIndex(hus2, numChildren-1);
            setChildIndex(hus3, numChildren-1);
            setChildIndex(hus4, numChildren-1);

            setChildIndex(hero, numChildren-1);
            brunnen.x = 1456;
            setChildIndex(brunnen, numChildren-1);
    }else {
            boundary_bondgarden.y = -555;
            btn_bonddorr.x = 2222;
            brunnen.x = 2222;
            hus1.y = 888;
            hus2.y = 888;
            hus3.y = 888;
            hus4.y = 888;
    }

    if(partofMap == 4){
            btn_kvarndorr.x = 383;
            btn_kvarndorr.y = 293;
            btn_kvarndorrMask.x = 383;
            btn_kvarndorrMask.y = 293;

            if(uppg_sackTaken == false && showSack == true){
                    uppg_sack.x = 434;

            }
            if(sfxBirdsComplete == true)fagelljud();
            setChildIndex(btn_kvarndorr, numChildren-1);
            setChildIndex(btn_kvarndorrMask, numChildren-1);
            setChildIndex(uppg_sack, numChildren-1);
    }else{
            malarvingar.y = -2222;
            btn_kvarndorr.x = 2222;
            btn_kvarndorrMask.x = 2222;
    }

    if(partofMap == 5){
            if(sfxBirdsComplete == true)fagelljud();

    }

    onetimekvarn = false;
```

```
klickSkoldorr = false; //Så skoldörrspop-uppen far ner om den e uppe.

// Placera spelaren till sin startposition.
hero.x = herostartX;
hero.y = herostartY;
hero.newx = herostartX;
hero.newy = herostartY;
hero.walktoX = herostartX;
hero.walktoY = herostartY;
cannotMove = true;
hero.boundaries();
bg.x = 0;

//Reset av navigeringspilarna.
arrowSignLeft.walkout = "no";
arrowSignLeft.turnArrow();
arrowSignLeft.goleftExterior();
arrowSignLeft.textarrowsign.text = " ";
setChildIndex(arrowSignLeft, numChildren-1);

//Reset av bakgrundsändraren.
bg.bgSwitcher();

//Spelaren kan styra spelfiguren igen
cannotMove = false;
startHero();
//Objektens z kordinater styr igen.
lagaLager = true;
loadingmaask.x = 2222;
Tweener.addTween(this, {alpha:1, time:1});

changeOnce = true;
hero.leftscreen = "false";

//Starta tomten och fjärillen.
fjarilen.startFjaril();
tomten.startTomte();

}

//Funktion för att veta till vilken exteriör du kommer till från en annan.
public function changeBg(){
	if(hero.leftscreen == "left"){

		//switch för att veta till vilken exteriör du kommer till från en
		annan.
		//1 = smeden, 2 = handeln/(skolan), 3  = Bondgården, 4 = Kvarnen, 5
		= Kloka gummans stuga
		switch(partofMap){
			case 1:
			partofMap = 5;
			break;
			case 2:
			//partofMap = 1;
			break;
			case 3:
			partofMap = 2;
			break;
			case 4:
			partofMap = 5;
			break;
			case 5:
			//partofMap = 4;
			break;
		}


		//1 = smeden, 2 = handeln/(skolan), 3  = Bondgården, 4 = Kvarnen, 5
		= Kloka gummans stuga
	}else if (hero.leftscreen == "right"){
```

```
                switch(partofMap){
                case 1:
                partofMap = 3;
                break;
                case 2:
                partofMap = 1;
                break;
                case 3:
                partofMap = 1;
                break;
                case 4:
                partofMap = 3;
                break;
                case 5:
                partofMap = 4;
                break;
                }

                //1 = smeden, 2 = handeln/(skolan), 3  = Bondgården, 4 = Kvarnen, 5
                = Kloka gummans stuga
        }else if (hero.leftscreen == "button"){

                switch(partofMap){
                case 1:
                partofMap = 2;
                break;
                case 2:
                partofMap = 3;
                break;
                case 3:
                partofMap = 4;
                break;
                case 4:
                //partofMap = 5;
                break;
                case 5:
                partofMap = 1;
                break;
                }

        }

        cpanel.youarehere.switchYouarehere();

}

//***** Fade *****//

//Fade in för hela rutan
public function fadeIn(){
        this.alpha = 0;
        Tweener.addTween(this, {alpha:1, time:1});
        Tweener.addTween(hero, {alpha:1, time:1});
}
//Fade out för hela rutan
public function fadeOut(){
        this.alpha = 1;
        Tweener.addTween(this, {alpha:0, time:1});
        hero.alpha = 0;
}

//***** Audio *****//

//Stoppa audion.
public function stangLjud(){

        if(sfxAmbientComplete == false)endambientljud();
        if(sfxBirdsComplete == false)endfagelljud();
```

```
}
//Starta ljud.
public function ambientljud(){

        if(ljudAv == false){
                sfxAmbientComplete = false;
                sfxAmbientChannel = sfxAmbient.play(0, 1);
                sfxAmbientChannel.addEventListener(Event.SOUND_COMPLETE,
                sfxAmbientCompleteHandler);
        }
}
//När ljudet spelats klart, starta det pånytt.
private function sfxAmbientCompleteHandler(e:Event):void {
 sfxAmbientComplete = true;
        ambientljud();
}
//Stoppa ljudet.
private function endambientljud(){
        sfxAmbientChannel.stop();
        sfxAmbientComplete = true;
}
//Starta ljud.
public function fagelljud(){

        if(ljudAv == false){
                sfxBirdsComplete = false;
                sfxBirdsChannel = sfxBirds.play(0, 1);
                sfxBirdsChannel.addEventListener(Event.SOUND_COMPLETE,
                sfxBirdsCompleteHandler);
        }
}
//När ljudet spelats klart, starta det pånytt.
private function sfxBirdsCompleteHandler(e:Event):void {
        sfxBirdsComplete = true;
        fagelljud();
}
//Stoppa ljudet.
private function endfagelljud(){
        sfxBirdsChannel.stop();
        sfxBirdsComplete = true;
}
//Starta ljudeffekt.
public function dorrljud(){
        if(ljudAv == false){
                sfx_doorComplete = false;
                sfx_doorChannel = sfx_door.play(0, 1);
                sfx_doorChannel.addEventListener(Event.SOUND_COMPLETE,
                sfx_doorCompleteHandler);
        }
}
//Ljudet spelat till slut.
private function sfx_doorCompleteHandler(e:Event):void {
sfx_doorComplete = true;
 }
//Starta ljudeffekt.
public function uppgutfljud(){
        if(ljudAv == false){
                sfx_uppgutfComplete = false;
                sfx_uppgutfChannel = sfx_uppgutf.play(0, 1);
                sfx_uppgutfChannel.addEventListener(Event.SOUND_COMPLETE,
                sfx_uppgutfCompleteHandler);
        }
}
//Ljudet spelat till slut.
private function sfx_uppgutfCompleteHandler(e:Event):void {
sfx_uppgutfComplete = true;
}
//Starta ljudeffekt.
public function ljudfanfare(){
```

```
		if(ljudAv == false){
			sfx_fanfareComplete = false;
			sfx_fanfareChannel = sfx_fanfare.play(0, 1);
			sfx_fanfareChannel.addEventListener(Event.SOUND_COMPLETE,
			sfx_fanfareCompleteHandler);
		}
	}
	//Ljudet spelat till slut.
	private function sfx_fanfareCompleteHandler(e:Event):void {
		sfx_fanfareComplete = true;
	}
	//Ta bort denna instans (karta).
	public function deleteKarta() {
		removeEventListener(Event.ENTER_FRAME, kartaLoop);
		parent.removeChild(this);

	//************************************************//

	}
	//Förhindra spelaren från att röra på spelfiguren.
	public function stopHero(){
		downArrow = false;
		upArrow = false;
		rightArrow = false;
		leftArrow = false;
		stage.removeEventListener(KeyboardEvent.KEY_DOWN,keyDownFunction);
		stage.removeEventListener(KeyboardEvent.KEY_UP,keyUpFunction);
		stage.removeEventListener(MouseEvent.CLICK,walkTo);
	}
	//Återaktivera spelfiguren.
	public function startHero(){
		// lyssna efter tangentbord
		stage.addEventListener(KeyboardEvent.KEY_DOWN,keyDownFunction);
		stage.addEventListener(KeyboardEvent.KEY_UP,keyUpFunction);
		stage.addEventListener(MouseEvent.CLICK,walkTo);
	}

	//Ordning av objekts z-kordinater. Kan kallas från andra klasser.
	public function ordnaUppgift(){
		setChildIndex(cpanel.uppdrag, numChildren-1);
		setChildIndex(cpanel.nyMap, numChildren-1);
		setChildIndex(cpanel.youarehere, numChildren-1);
		setChildIndex(cpanel.xuppdrag, numChildren-1);

	}
	//Ordning av fjärillens z-kordinater. Kan kallas från andra klasser.
	public function ordnaFjaril(){
		setChildIndex(fjarilen, numChildren-1);
	}
	//Ordning av tomtens z-kordinater. Kan kallas från andra klasser.
	public function ordnaTomte(){
		setChildIndex(tomten, numChildren-1);
	}

	//När griffeltavlan avslutas utförs ett uppdrag.
	public function endSkolGrif(){
	lagaLager = true;
		if(harGriflat == true && harSkolfilm == true && harSkoldagOnce == true){
			harSkoldagOnce = false;

			popUtford();
			tblUppdrag[6] = true;
			checkuppdrag();
		}
	}
	//När memoryspelet avslutas utförs ett uppdrag.
	public function endUppgKloka(){

		ReallydoNotWalk();
```

```
		if(harKloka == true && harKlokaOnce == true){
			harKlokaOnce = false;

			popUtford();
			tblUppdrag[4] = true;
			checkuppdrag();
		}
	}
	//När bondgårdsspelet avslutas utförs ett uppdrag.
	public function endUppgBond(){

	harBond = true;
		if(harBond == true && harBondOnce == true){
			harBondOnce = false;

			popUtford();
			tblUppdrag[2] = true;

			checkuppdrag();

		}
	}
	//Spelaren har talat med handlaren.
	public function handlaren(){
		harTalatHandlare = true;
		trace("harTalatHandlare "+harTalatHandlare);
	}
	//Spelaren har talat med handlaren.
	public function handlarenUpp1(){
		tblUppdrag[0] = true; //uppdrag 1 ägg (tabellen börjar från 0 därför
		tblUppdrag[0])
		trace("tblUppdrag[0] "+tblUppdrag[0]);
		checkuppdrag();
	}
	//Spelaren har jagat möss.
	public function handlarenUpp2(){
		tblUppdrag[1] = true; //uppdrag 2 ris kaffe (tabellen börjar från 0 därför
		tblUppdrag[1])
		trace("tblUppdrag[1] "+tblUppdrag[1]);
		checkuppdrag();
	}
	//Spelaren har kokat gröt.
	public function smedenUpp1grot(){
		harkokatgrot = true;
		if(harkokatgrot == true && harnaggatbrod == true){
			tblUppdrag[5] = true;
			trace("tblUppdrag[5] "+tblUppdrag[5]);
			checkuppdrag();
		}
	}
	//Spelaren har naggat bröd
	public function smedenUpp2nagg(){
		harnaggatbrod = true;

		if(harkokatgrot == true && harnaggatbrod == true){
			tblUppdrag[5] = true;
			trace("tblUppdrag[5] "+tblUppdrag[5]);
			checkuppdrag();
		}
	}
	//Spelaren har spelat memory.
	public function memoryUtford(){

		harKloka = true;
		trace("harKloka "+harKloka);
	}

	//Funktion för mottagande av variabler från andra delar av spelet.
```

```
public function varReceiver(vartsu, vartsu2){
        trace("var receiver karta språket är" +vartsu);
        whatLanguage = vartsu;
        rollfigurK = vartsu2;
        trace("rollfigK "+rollfigurK);

        bubvisste.sprak = whatLanguage;
        bubvisste.bubvislanguage();

        cpanel.Uppdragsprak();
        instruktioner.helpText();

        if(whatLanguage == 2){
                poputford.txt_poputford.text = "Tehtävä suoritettu!";
        }else{
                poputford.txt_poputford.text = "Uppdrag utfört!";
        }

}

//Laddare (av griffeltavlan).
public function loadGriffel(){

        lbar.x = 0;
        txtLoading.x = 270;
        loadingmaask.x = 0;
        loadingmaask.alpha = 0;
        Tweener.addTween(loadingmaask, {alpha:0.6, time:0.8});

        loaderkvarn.x = 392;
        loadervinge.x = 450.20;
        setChildIndex(loadingmaask, numChildren-1);
        setChildIndex(loaderkvarn, numChildren-1);
        setChildIndex(loadervinge, numChildren-1);

        if(whatLanguage == 2){
        txtLoading.text = "Ladataan Liitutaulua";
        }else txtLoading.text = "Laddar Griffeltavla";

        if(loadedOnceGriffel == false){
                griffelloadern.griffelloader();
                addChild(griffelloadern);
                loadedOnceGriffel = true;
        }else{
                var griffelloadern2:Griffelloader = new Griffelloader();
                griffelloadern2.griffelloader();
                addChild(griffelloadern2);

        }
}

//Laddare (skjutspelet).
public function loadShoot(){

        loadingmaask.x = 0;
        loadingmaask.alpha = 0;
        Tweener.addTween(loadingmaask, {alpha:0.6, time:0.8});
        loaderkvarn.x = 392;
        loadervinge.x = 450.20;
        setChildIndex(loadingmaask, numChildren-1);
        setChildIndex(loaderkvarn, numChildren-1);
        setChildIndex(loadervinge, numChildren-1);

        if(loadedOnceShoot == false){
                shootloadern.shootloader();
                addChild(shootloadern);
                loadedOnceShoot = true;
        }else{
                var shootloadern2:Shootloader = new Shootloader();
```

```
                shootloadern2.shootloader();
                addChild(shootloadern2);
        }
}

//Laddare (bondstugans interiör).
public function loadBond(){

        loadingmaask.x = 0;
        loadingmaask.alpha = 0;
        Tweener.addTween(loadingmaask, {alpha:0.6, time:0.8});
        loaderkvarn.x = 392;
        loadervinge.x = 450.20;
        setChildIndex(loadingmaask, numChildren-1);
        setChildIndex(loaderkvarn, numChildren-1);
        setChildIndex(loadervinge, numChildren-1);

        if(loadedOnceBond == false){
                bondloadern.bondloader();
                addChild(bondloadern);
                loadedOnceBond = true;
        }else{
                var bondloadern2:Bondloader = new Bondloader();
                bondloadern2.bondloader();
                addChild(bondloadern2);

        }
}

//Laddare (griffeltavlan).
public function loadKlok(){

        loadingmaask.x = 0;
        loadingmaask.alpha = 0;
        Tweener.addTween(loadingmaask, {alpha:0.6, time:0.8});
        loaderkvarn.x = 392;
        loadervinge.x = 450.20;
        setChildIndex(loadingmaask, numChildren-1);
        setChildIndex(loaderkvarn, numChildren-1);
        setChildIndex(loadervinge, numChildren-1);

        if(loadedOnceKlok == false){
                klokloadern.klokloader();
                addChild(klokloadern);
                loadedOnceKlok = true;
        }else{
                var klokloadern2:Klokloader = new Klokloader();
                klokloadern2.klokloader();
                addChild(klokloadern2);
        }
}

//Laddare (handelns interiör och musspelet).
public function loadMus(){

        loadingmaask.x = 0;
        loadingmaask.alpha = 0;
        Tweener.addTween(loadingmaask, {alpha:0.6, time:0.8});
        loaderkvarn.x = 392;
        loadervinge.x = 450.20;
        setChildIndex(loadingmaask, numChildren-1);
        setChildIndex(loaderkvarn, numChildren-1);
        setChildIndex(loadervinge, numChildren-1);

        if(loadedOnceMus == false){
                musloadern.musloader();
                addChild(musloadern);
                loadedOnceMus = true;
        }else{
```

```
                        var musloadern2:Musloader = new Musloader();
                        musloadern2.musloader();
                        addChild(musloadern2);
                        if(harTalatHandlare == true){
                                musloadern2.sendtalked();
                        }
                }
        }
        //Laddare (smedens interiör).
        public function loadSmed(){

                loadingmaask.x = 0;
                loadingmaask.alpha = 0;
                Tweener.addTween(loadingmaask, {alpha:0.6, time:0.8});
                loaderkvarn.x = 392;
                loadervinge.x = 450.20;
                setChildIndex(loadingmaask, numChildren-1);
                setChildIndex(loaderkvarn, numChildren-1);
                setChildIndex(loadervinge, numChildren-1);

                if(loadedOnceSmed == false){
                        smedloadern.smedloader();
                        addChild(smedloadern);
                        loadedOnceSmed = true;
                }else{
                        var smedloadern2:Smedloader = new Smedloader();
                        smedloadern2.smedloader();
                        addChild(smedloadern2);
                        if(harTalatHandlare == true){
                                smedloadern2.sendtalked();
                        }
                }
        }

        //Tilffällig av-aktivering av spelfiguren
        public function doNotWalk(){
                hero.walktoX = hero.x;
                hero.walktoY = hero.y;
        }

        //Strängare tilffällig av-aktivering av spelfiguren med timer.
        public function ReallydoNotWalk(){
                hero.walktoX = hero.x;
                hero.walktoY = hero.y;
                stopHero();

                Tweener.addTween(tweendummy2, {x:2222, y:0, time:0.4, transition:"linear",
onComplete:function() {
                starth();
                }});

                function starth(){
                        startHero();
                }

        }

        //För att spelfiguren inte ska röra på sig då spelaren klickar på vissa objekt
        (annars rör sig spelfiguren dit p.g.a. klicket på vänstra musknappen, alltså p.g.a.
        spelfigurens styrning.
        public function walkTo(event:MouseEvent){

                        if(clkickedWalkOnce == true){
                                clkickedWalkOnce = false;
                                arrowSignLeft.walkout = "no";
                        }
```

```
                        if(maria.x != 1111 && mouseX > maria.x && mouseX < maria.x + maria.marWidth
                        && mouseY > maria.y && mouseY < maria.y + maria.marHeight){ //Så att
                        spelaren inte rör sig då man trycker på smedismaria
                        } else if(donotwalk == true){

                        } else if(maria.bubblaActive == true && mouseX > maria.bubbla.x && mouseX <
                        maria.bubbla.x + maria.bubbla.bubWidth && mouseY > maria.bubbla.y && mouseY
                        < maria.bubbla.y + maria.bubbla.bubHeight){
                                //kan inte röra på sig om man trycker på pratbubblan
                        } else if(xuppdragIsHit == true){
                                //kan inte röra på sig om man trycker på uppdragsrutans X /
                        Stängknapp
                        } else if(cpanel.uppdragActive == false){
                                        if (arrowSignLeft.walkout == "left" && partofMap == 3 ||
                        arrowSignLeft.walkout == "left" && partofMap == 2 || arrowSignLeft.walkout
                        == "left" && partofMap == 1 && clkickedWalkOnce == false){
                                                hero.walktoX = -2222;
                                                hero.walktoY = 460;
                                                clkickedWalkOnce = true;
                                        }else if (arrowSignLeft.walkout == "left" && clkickedWalkOnce
                                        == false){
                                                hero.walktoX = -1111;
                                                clkickedWalkOnce = true;
                                        }else if (arrowSignLeft.walkout == "right" && partofMap == 3
                                        || arrowSignLeft.walkout == "right" && partofMap == 2 ||
                                        arrowSignLeft.walkout == "right" && partofMap == 1 &&
                                        clkickedWalkOnce == false){
                                                hero.walktoX = 2222;
                                                hero.walktoY = 460;
                                                clkickedWalkOnce = true;
                                        }else if (arrowSignLeft.walkout == "right" &&
                                        clkickedWalkOnce == false){
                                                hero.walktoX = 1111;
                                                clkickedWalkOnce = true;
                                        }else{

                                                if(mouseY < borderBottom + 32){
                                                        hero.walktoX = mouseX;
                                                        hero.walktoY = mouseY;
                                                }

                                        }

                                        if (hero.walktoX < hero.x) hero.direction = -1;
                                        if (hero.walktoX > hero.x) hero.direction = 1;

                                }

                        }

                }

}
```

D.3.5 Funktionslista för klassen Hero

Nr	Funktion	Komplexitet	typ
F1	Hero	Normal	Main
	Main loop		
F2	moveHero	Svår	Animering
	Tidsbaserad animering och förflyttning		
F3	moveheroKeyStatic	Svår	Animering
	Förflyttning i statisk exteriör (tangentbord)		
F4	moveheroMouseStatic	Svår	Animering
	Förflyttning i statisk exteriör (mus)		
F5	moveheroKeyScroll	Svår	Animering
	Förflyttning i scrollande exteriör (tangentbord)		
F6	moveheroMouseScroll	Svår	Animering
	Förflyttning i scrollande exteriör (mus)		
F7	boundaries	Normal	Funktionalitet
	Gränser för spelfiguren (statisk)		
F8	scrollBoundaries	Normal	Funktionalitet
	Gränser för spelfiguren (scrolling)		
F9	keyStopHero	Normal	Tangentbord
	Stoppar figuren när tangentbord trycks efter musklick		
F10	animateHero	Normal	Animering
	Animation (byter bilder i sekvens)		
F11	stopHero	Normal	Funktionalitet
	Tillfällig inaktivering av spelfiguren		
F12	deleteHero	Normal	Minnestömning
	Radera denna instans av spelfiguren		

Spelfiguren kan styras antingen med piltangenterna på tangentbordet eller med musen. Styrningen av spelfiguren sköts från spelfigurens egna klass. Eftersom det finns både statiska och scrollande exteriörer behövs det fyra skillda system för styrandet av spelfiguren. Styrning med mus i exteriör både med statisk eller rörlig bakgrund och styrning med tangentbord både vid statisk och rörlig bakgrund. Ett bekvämt styrande av spelfiguren är naturligtvis viktig och funktionen finslipades genom hela produktionen. Programmeringen av scrolling alltså rörlig bakgrund är relativt invecklat eftersom utöver bakgrundens förflyttning bör alla objekt på spelplan förflyttas och alla objekt bör kollisionskontrolleras mot spelfiguren så att spelaren t.ex. inte kan röra sig igenom hus. Dessutom bör spelfigurens z kordinat tas i beaktande i förhållande till x och y kordinaterna för att kontrollera om spelfiguren ska visas bakom eller framför andra objekt på spelplanen.

D.3.6 Programkod för klassen Hero

```
/****   Klassen Hero           ****/

package {
        import flash.display.*;
        import flash.events.*;
        import flash.utils.getTimer;
        import flash.utils.Timer;

        public class Hero extends MovieClip {
```

```
private var herowidth = 32;
private var heroheight = 32;
private var borderBottom = 436;
private var borderUp = 150;
private var borderLeft = 0;
private var borderRight = 900;
private var leaveLeftBond = 102;
private var leaveLeft = -6;
private var leaveRight = 888;
private var iswalking:String = "no"; //no, left eller right.
private var walkmode:String = "static"; //static eller scroll.
private var scrollBgKant:int; //Sätt pixelbredden på scrollande bakgrunden beroende
av exteriör.

private var ctrlPanelHeight = 80; //Så man kan stoppa spelfiguren vid kanten av
menyn.
static const speed:Number = 164.0;
private var lastTime:int; //Tid sedan senaste loop. För animationen av spelfiguren.
private var timePassed:int = 0;
private var animstep:Number = 1;
public var direction = -1; // 1 = höger, -1 = vänster, 2 = neråt,  -2 = uppåt.
public var verticalDir =  2; // 2 = neråt, -2 = uppåt.
public var horizontalDir = 1; // 1 = höger, -1 = vänster.
public var herow:int = herowidth;
public var heroh:int = heroheight;
public var heroIshit:Boolean = false;
public var onetimeTimerSwitch:Boolean = false; //Den tidsbaserade animationen
behöver en tillbakaflyttning av tiden den första gången för att fungera korrekt.
public var walktoX:int;
public var walktoY:int;
//Position denna loop.
public var newx:int;
public var newy:int;
public var bg_newx:int;
public var leftscreen:String;// Om spelfiguren lämnat skärmen till höger eller
vänster eller om den inte alls lämnat skärmen.
public var changedOnce:Boolean = false; // Så att inte exteriören byts många ggr i
rad p.g.a. spelfigurens position.
private var boundaryUpHit:Boolean; // För att kontrollera övre gränsen.
private var boundaryLeftHit:Boolean; // För att kontrollera vänstra gränsen.
private var boundaryRightHit:Boolean; // För att kontrollera högra gränsen.
private var boundaryDownHit:Boolean;
private var vilkenBoundary:int; //Gräns för scrolling beroende på exteriör 0 =
handelnskolan 1 = bondgården 2 = smeden osv.
private var animationspeed = 216;
private var animationspeedLR = 110;

//*** Klassens Main funktion. ***
public function Hero() {
// Ingenting här, spelfigurens rörelser och animation kallas från karta klassens
(som är heron klassens parent) Main loop.
        walktoX = 600;
        walktoY = 324+66;

}

public function moveHero() {

        // För att inte förflytta spelfiguren för mycket då denne aktiveras.
        if(onetimeTimerSwitch == false){
                lastTime = getTimer() - 20;
                onetimeTimerSwitch = true;
        }

        // Tidsskillnaden.
        timePassed = getTimer()-lastTime;
        lastTime += timePassed;

        //Position denna loop.
        newx = this.x;
```

```
newy = this.y;

bg_newx = MovieClip(parent).bg.x;

if(heroIshit == true){
//Stanna spelfiguren om det behövs.
        stopHero();
}

if(MovieClip(parent).cannotMove == true){
        stopHero();
}

boundaries(); // Gränser för var spelfiguren kan röra sig.

switch(walkmode){
                case "static":
        //Rörelse av spelfiguren med piltangenter i en statisk exteriör.
                moveheroKeyStatic();
        //Rörelse av spelfiguren med musklick i en statisk exteriör.
                moveheroMouseStatic();
                break;
                case "scroll":
                        switch(MovieClip(parent).partofMap){
                        case 1:
                        scrollBgKant = -345;
                        break;
                        case 2:
                        scrollBgKant = -1087;
                        break;
                        case 3:
                        scrollBgKant = -1028;
                        break;
                        }
        //Rörelse av spelfiguren med piltangenter i en scrollande exteriör.
                moveheroKeyScroll();
        //Rörelse av spelfiguren med musklick i en scrollande exteriör.
                moveheroMouseScroll();
                break;
                }


        //*************************************************************
        //***** Kontrollera gränser. ******************************
        //*************************************************************


        //Kloka gumman.
        if (newx <= leaveLeft && MovieClip(parent).partofMap == 5){
                newx = borderLeft;
        }else if (newx <= leaveLeftBond && MovieClip(parent).partofMap ==
        3){
                leftscreen = "left";

                if(changedOnce == false)MovieClip(parent).exteriorChange();
        }else if (newx + herowidth <= leaveLeft){
                leftscreen = "left";

                if(changedOnce == false)MovieClip(parent).exteriorChange();
        }else if (newx >= leaveRight){
                leftscreen = "right";

                if(changedOnce == false)MovieClip(parent).exteriorChange();
        }else if(leftscreen != "button"){
                leftscreen = "false";
        }

        if (newy <= borderUp) newy = borderUp;
        if (newy >= borderBottom) newy = borderBottom;
        if (walktoY <= borderUp) walktoY = borderUp;
```

```
if (walktoY >= borderBottom + 32) walktoY = borderBottom;


// Repositionera.
if (leftscreen == "false"){
this.x = newx;
this.y = newy;
MovieClip(parent).bg.x = bg_newx;


switch(MovieClip(parent).partofMap){
case 1: //Smeden.
MovieClip(parent).maria.x = bg_newx + 610;
MovieClip(parent).btn_smeddorr.x = bg_newx + 833;
MovieClip(parent).gunmask.x = bg_newx + 132;
MovieClip(parent).boundary_smeden.x = MovieClip(parent).bg.x;
MovieClip(parent).pil_N_E.x = bg_newx + 700;
MovieClip(parent).pil_N_E.y = 250;

                if(MovieClip(parent).tomten.activeTomte == true){
                        switch(MovieClip(parent).tomten.vilkenTomte){
                        case 1:
                        MovieClip(parent).tomten.x = bg_newx + 266;
                        break;
                        case 2:
                        MovieClip(parent).tomten.x = bg_newx + 266;
                        break;
                        case 3:
                        MovieClip(parent).tomten.x = bg_newx + 766;
                        break;
                        }
                }
break;
case 2: //Handeln/Skolan.
MovieClip(parent).skoldorr.x = bg_newx + 1044;
MovieClip(parent).btn_handeldorr.x = bg_newx + 307;
MovieClip(parent).popupskola.x = bg_newx + 864;
MovieClip(parent).boundary_handeln.x = MovieClip(parent).bg.x;
MovieClip(parent).pil_N_E.x = bg_newx + 1708;
MovieClip(parent).pil_N_E.y = 250;

                if(MovieClip(parent).tomten.activeTomte == true){
                switch(MovieClip(parent).tomten.vilkenTomte){
                        case 1:
                        MovieClip(parent).tomten.x = bg_newx + 187;
                        break;
                        case 2:
                        MovieClip(parent).tomten.x = bg_newx + 898;
                        break;
                        case 3:
                        MovieClip(parent).tomten.x = bg_newx + 1554;
                        break;
                        }
                }
break;
case 3: //Bondgården.
MovieClip(parent).boundary_bondgarden.x = MovieClip(parent).bg.x;
MovieClip(parent).btn_bonddorr.x = bg_newx + 482;
MovieClip(parent).brunnen.x = bg_newx + 1456;
MovieClip(parent).pil_N_E.x = bg_newx + 794;
MovieClip(parent).pil_N_E.y = 264;
MovieClip(parent).hus1.x = bg_newx + 2;
MovieClip(parent).hus2.x = bg_newx + 163;
MovieClip(parent).hus3.x = bg_newx + 681;
MovieClip(parent).hus4.x = bg_newx + 1378;

                if(MovieClip(parent).tomten.activeTomte == true){

                        switch(MovieClip(parent).tomten.vilkenTomte){
```

```
                                        case 1:
                                        MovieClip(parent).tomten.x = bg_newx + 91;
                                        break;
                                        case 2:
                                        MovieClip(parent).tomten.x = bg_newx + 671;
                                        break;
                                        case 3:
                                        MovieClip(parent).tomten.x = bg_newx + 1346;
                                        break;
                                        }
                                }

                break;
                case 5:

                break;
                }

        if(MovieClip(parent).bg.x < scrollBgKant)MovieClip(parent).bg.x = scrollBgKant;
        if(MovieClip(parent).bg.x > 0)MovieClip(parent).bg.x = 0;


                }

        }

        public function moveheroKeyStatic(){
                //***** Rörelse av spelfiguren med piltangenter i en statisk exteriör. *****

                ///////////////trace(timePassed);
                if(timePassed > 40) timePassed = 40; // För att undvika för stora
                hopp i spelfigurens rörelse.
                // Rör spelfiguren till vänster.
                if (MovieClip(parent).leftArrow && !MovieClip(parent).rightArrow) {
                        newx -= speed*timePassed/1000;
                        direction = -1;
                        horizontalDir = -1;
                        keyStopHero();
                }

                // Rör spelfiguren till höger.
                if (MovieClip(parent).rightArrow) {
                        newx += speed*timePassed/1000;
                        direction = 1;
                        horizontalDir = 1;
                        keyStopHero();
                }

                // Rör spelfiguren uppåt.
                if (MovieClip(parent).upArrow && !MovieClip(parent).downArrow) {
                        newy -= speed*timePassed/1000;
                        direction = -2;
                        verticalDir = -2;
                        keyStopHero();
                }

                // Rör spelfiguren neråt.
                if (MovieClip(parent).downArrow) {
                        newy += speed*timePassed/1000;
                        direction = 2;
                        verticalDir = 2;
                        keyStopHero();
                }
        }

        public function moveheroMouseStatic(){
        // ***** Rörelse av spelfiguren med musklick i en statisk exteriör. *****

                if (!MovieClip(parent).leftArrow && !MovieClip(parent).rightArrow &&
                !MovieClip(parent).upArrow && !MovieClip(parent).downArrow){
```

```
                // Rör spelfiguren till vänster.
                if (walktoX != this.x){
                        if (walktoX < this.x) {
                                newx -= speed*timePassed/1000;
                                direction = -1;
                                horizontalDir = -1;
                        }//Om spelfiguren rör sig för långt till vänster placera den
                        vid musklickets kordinater.
                        else if (direction == -1 && walktoX > this.x) newx = walktoX;
                        //Rör spelfiguren till höger
                        else if (walktoX > this.x + herow) {
                                newx += speed*timePassed/1000;
                                direction = 1;
                                horizontalDir = 1;
                        }
                }//Rör spelfiguren uppåt.
                if (walktoY != this.y){
                        if (walktoY < this.y) {
                                newy -= speed*timePassed/1000;
                                direction = -2;
                                verticalDir = -2;
                        }//Om spelfiguren rör sig för långt uppåt placera den vid
                        musklickets kordinater.
                        else if (direction == -2 && walktoY > this.y) newy = walktoY;
                        //Rör spelfiguren neråt.
                        else if (walktoY > this.y + heroh) {
                                newy += speed*timePassed/1000;
                                direction = 2;
                                verticalDir = 2;
                        }
                }
        }
}

public function moveheroKeyScroll(){
//***** Rörelse av spelfiguren med piltangenter i en scrollande exteriör *****
        if(timePassed > 40) timePassed = 40; // För att undvika för stora hopp i
        spelfigurens rörelse.
                // Rör spelfiguren till vänster.
                if (MovieClip(parent).leftArrow && !MovieClip(parent).rightArrow &&
        boundaryLeftHit == false) {
                        if(this.x <= 348 && MovieClip(parent).bg.x < 0){
                                bg_newx += speed*timePassed/1000;
                                direction = -1;
                                horizontalDir = -1;
                                keyStopHero();
                        }else newx -= speed*timePassed/1000;
                        direction = -1;
                        horizontalDir = -1;
                        keyStopHero();

                }

                //Rör spelfiguren till höger.
                if (MovieClip(parent).rightArrow && boundaryRightHit == false) {
                        if(this.x >= 548 && MovieClip(parent).bg.x > scrollBgKant){
                                bg_newx -= speed*timePassed/1000;
                                direction = 1;
                                horizontalDir = 1;
                                keyStopHero();
                        }else newx += speed*timePassed/1000;
                        direction = 1;
                        horizontalDir = 1;
                        keyStopHero();

                }

                //Rör spelfiguren uppåt.
```

```
                        if (MovieClip(parent).upArrow && !MovieClip(parent).downArrow &&
                        boundaryUpHit == false) {

                                newy -= speed*timePassed/1000;
                                direction = -2;
                                verticalDir = -2;
                                keyStopHero();

                        }

                        //Rör spelfiguren neråt.
                        if (MovieClip(parent).downArrow && boundaryDownHit == false) {
                                newy += speed*timePassed/1000;
                                direction = 2;
                                verticalDir = 2;
                                keyStopHero();

                        }
                }
                public function moveheroMouseScroll(){
                // ***** Rörelse av spelfiguren med musklick i en scrollande exteriör. *****

                        if (!MovieClip(parent).leftArrow && !MovieClip(parent).rightArrow &&
                !MovieClip(parent).upArrow && !MovieClip(parent).downArrow){
                        // Rör spelfiguren till vänster.
                        if (walktoX != this.x ){
                                if (walktoX < this.x && this.x <= 348 &&
                                MovieClip(parent).bg.x < 0 && boundaryLeftHit == false) {
                                        bg_newx += speed*timePassed/1000;
                                        walktoX += speed*timePassed/1000;
                                        direction = -1;
                                        horizontalDir = -1;
                                }else if (walktoX < this.x && boundaryLeftHit == false){
                                        newx -= speed*timePassed/1000;
                                        direction = -1;
                                        horizontalDir = -1;
                                }//Om spelfiguren rör sig för långt till vänster placera den
                                vid musklickets kordinater.
                                else if (direction == -1 && walktoX > this.x) newx = walktoX;
                                //Rör spelfiguren till höger.
                                else if (walktoX > this.x + herow && this.x + herow >= 548 &&
                                MovieClip(parent).bg.x > scrollBgKant && boundaryRightHit ==
                                false) {
                                        bg_newx -= speed*timePassed/1000;
                                        walktoX -= speed*timePassed/1000;
                                        direction = 1;
                                        horizontalDir = 1;
                        }else if (walktoX > this.x + herow && boundaryRightHit == false){
                                        newx += speed*timePassed/1000;
                                        direction = 1;
                                        horizontalDir = 1;
                                }else if (direction == 1 && walktoX < this.x) newx = walktoX;
                                }//Rör spelfiguren uppåt.
                                if (walktoY != this.y)  {
                                if (walktoY < this.y && boundaryUpHit == false) {
                                        newy -= speed*timePassed/1000;
                                        direction = -2;
                                        verticalDir = -2;
                                }//Om spelfiguren rör sig för långt uppåt placera den vid
                                musklickets kordinater.
                                else if (direction == -2 && walktoY > this.y) newy = walktoY;
                                //Rör spelfiguren neråt.
                                else if (walktoY > this.y + heroh && boundaryDownHit ==
                                false) {
                                        newy += speed*timePassed/1000;
                                        direction = 2;
                                        verticalDir = 2;
                                }
                        }
```

```
        }
}

public function boundaries(){

        switch(MovieClip(parent).partofMap){

                case 1: //Smedens stuga.
                        borderUp = 0;
                        walkmode = "scroll";
                        vilkenBoundary = 2;
                        scrollBoundaries();

                break;
                case 2: //Handeln och Skolan.
                        borderUp = 0;
                        walkmode = "scroll";
                        vilkenBoundary = 0;
                        scrollBoundaries();

                break;
                case 3: //Bondgården.
                        borderUp = 0;
                        walkmode = "scroll";
                        vilkenBoundary = 1;
                        scrollBoundaries();

                break;
                case 4: //Kvarnen.
                        walkmode = "static";
                        borderBottom = 436;
                        borderUp = 399;
                        borderLeft = 0;
                        borderRight = 900;
                        leaveLeft = -6;
                        leaveRight = 888;

                break;
                case 5: //Kloka Gumman.
                        walkmode = "static";
                        borderBottom = 436;
                        borderUp = 356;
                        borderLeft = 0;
                        borderRight = 900;
                        leaveLeft = -6;
                        leaveRight = 888;

                break;

        }

}

// Avgränsar området spelfiguren kan röra sig på med hjälp av kollision i
exteriörer med scrolling.
public function scrollBoundaries(){

        //Uppåt.
if (MovieClip(parent).boundaryArray[vilkenBoundary].hitTestPoint(this.x +
(this.herow), this.y, true)){
                boundaryUpHit = true;


        }else{
                        boundaryUpHit = false;
        }

        //Neråt.
```

```
            if (MovieClip(parent).boundaryArray[vilkenBoundary].hitTestPoint(this.x +
            (this.herow), this.y + (64), true)){
                    trace("downhit");
                    boundaryDownHit = true;

            }else{
                            boundaryDownHit = false;
            }

            //Vänster.
            if (MovieClip(parent).boundaryArray[vilkenBoundary].hitTestPoint(this.x,
            this.y + (this.heroh / 2), true)
            ||  MovieClip(parent).boundaryArray[vilkenBoundary].hitTestPoint(this.x,
            this.y + (this.heroh + 16), true)){

                    boundaryLeftHit = true;
            }else{
                    boundaryLeftHit = false;
            }
            //Höger.
            if (MovieClip(parent).boundaryArray[vilkenBoundary].hitTestPoint(this.x +
            (this.herow + 12), this.y + ((this.heroh / 2)), true)
            ||  MovieClip(parent).boundaryArray[vilkenBoundary].hitTestPoint(this.x +
            (this.herow + 12), this.y + (this.heroh + 16), true)){

                    boundaryRightHit = true;
            }else{
                            boundaryRightHit = false;
            }

    }


    //Funktion som stoppar spelfiguren när en piltangent trycks efter att spelaren
    styrts med musklick.
    public function keyStopHero(){
            switch(direction){
                    case 1://Höger.
                    walktoX = newx + herow;
                            if(verticalDir == 2)walktoY = newy + heroh;
                            else walktoY = newy;
                    break;
                    case -1://Vänster.
                    walktoX = newx;
                            if(verticalDir == 2)walktoY = newy + heroh;
                            else walktoY = newy;
                    break;
                    case 2://Neråt.
                    walktoY = newy + heroh;
                            if(horizontalDir == 1)walktoX = newx + heroh;
                            else walktoX = newx;
                    break;
                    case -2://Uppåt.
                    walktoY = newy;
                            if(horizontalDir == 1)walktoX = newx + heroh;
                            else walktoX = newx;
                    break;
            }
    }

    // Animation.
    public function animateHero(){

            switch(MovieClip(parent).rollfigurK){
                    case "flicka":

                            switch(direction){

                            case 1:
```

```
                if(MovieClip(parent).rightArrow
                || !MovieClip(parent).rightArrow && walktoX > this.x + herow
                || iswalking == "right"){

                        animstep += timePassed/animationspeedLR;
                                if (animstep > 10.99){animstep = 7;}
                                if (animstep < 7){animstep = 7;}
                        this.gotoAndStop(Math.floor(animstep));


                }else this.gotoAndStop(7);

                break;

                case -1:
        if(MovieClip(parent).leftArrow && !MovieClip(parent).rightArrow
                || !MovieClip(parent).leftArrow && walktoX < this.x
                || iswalking == "left"){

                        animstep += timePassed/animationspeedLR;
                                if (animstep > 14.99){animstep = 11;}
                                if (animstep < 11){animstep = 11;}
                        this.gotoAndStop(Math.floor(animstep));


                }else this.gotoAndStop(11);

                break;

                case 2:

                if(MovieClip(parent).downArrow
                || !MovieClip(parent).downArrow && walktoY > this.y + heroh){

                        animstep += timePassed/animationspeed;
                                if (animstep > 3.99){animstep = 2;}
                                if (animstep < 1){animstep = 2;}
                        this.gotoAndStop(Math.floor(animstep));


                }else this.gotoAndStop(1);

                break;

                case -2:
                if(MovieClip(parent).upArrow && !MovieClip(parent).downArrow
                || !MovieClip(parent).upArrow && walktoY < this.y &&
                boundaryUpHit == false){

                        animstep += timePassed/animationspeed;
                                if (animstep > 6.99){animstep = 5;}
                                if (animstep < 4){animstep = 5;}
                        this.gotoAndStop(Math.floor(animstep));


                }else this.gotoAndStop(4);

                break;

                }

        break;
        case "pojke":

        switch(direction){

                case 1:

                if(MovieClip(parent).rightArrow
```

```
                        || !MovieClip(parent).rightArrow && walktoX > this.x
                        + herow
                                || iswalking == "right"){

                                animstep += timePassed/animationspeedLR;
                                        if (animstep > 24.99){animstep = 21;}
                                        if (animstep < 21){animstep = 21;}
                                this.gotoAndStop(Math.floor(animstep));


                        }else this.gotoAndStop(21);

                break;

                case -1:
                        if(MovieClip(parent).leftArrow &&
                        !MovieClip(parent).rightArrow
                        || !MovieClip(parent).leftArrow && walktoX < this.x
                        || iswalking == "left"){

                                animstep += timePassed/animationspeedLR;
                                        if (animstep > 28.99){animstep = 25;}
                                        if (animstep < 25){animstep = 25;}
                                this.gotoAndStop(Math.floor(animstep));


                        }else this.gotoAndStop(25);

                break;

                case 2:

                        if(MovieClip(parent).downArrow
                        || !MovieClip(parent).downArrow && walktoY > this.y +
                        heroh){

                                animstep += timePassed/animationspeed;
                                        if (animstep > 17.99){animstep = 16;}
                                        if (animstep < 15){animstep = 16;}
                                this.gotoAndStop(Math.floor(animstep));


                        }else this.gotoAndStop(15);

                break;

                case -2:
                        if(MovieClip(parent).upArrow &&
                        !MovieClip(parent).downArrow ||
                        !MovieClip(parent).upArrow && walktoY < this.y &&
                        boundaryUpHit == false){

                                animstep += timePassed/animationspeed;
                                        if (animstep > 20.99){animstep = 19;}
                                        if (animstep < 18){animstep = 19;}
                                this.gotoAndStop(Math.floor(animstep));

                        }else this.gotoAndStop(18);

                break;

                }

                break;
        }

}

public function stopHero(){
        MovieClip(parent).downArrow = false;
```

```
                MovieClip(parent).upArrow = false;
                MovieClip(parent).rightArrow = false;
                MovieClip(parent).leftArrow = false;
        }

        //Radera denna instans av spelfiguren (hero).
        public function deleteHero() {

                parent.removeChild(this);

        }
    }
}
```

D.3.7 Funktionslista för klassen ControlPanel

Nr	Funktion	Komplexitet	typ
F1	controlPanel	Normal	Initialisering
	Initialisering av menyn		
F2	CreateMap	Normal	Funktionalitet
	Skapar instans av spelkartan		
F3	Uppdragsprak	Normal	Text
	Text för uppdragsboken		
F4	CreateUppdrag	Normal	Funktionalitet
	Inladdning		
F5	CreateYouarehere	Normal	Funktionalitet
	Skapar pil till spelkartan		
F6	Knapparna	Normal	Knappar
	Skapar menyns knappar		
F7	clickKnapp	Normal	Knappar
	Spelaren klickar på en av knapparna		
F8	pekaKnapp	Normal	Knappar
	Spelaren pekar på en av knapparna		
F9	dissaKnapp	Normal	Knappar
	Spelaren pekar ej mera på knapp		
F10	mapController	Normal	Funktionalitet
	Spelkartans funktionalitet		
F11	clickbtnXsmap	Normal	Knapp
	Stäng kartan knapp		
F12	uppdragsController	Normal	Funktionalitet
	Uppdragsbokens funktionalitet		
F13	closeUppdrag	Normal	Knapp
	Stäng uppdragsboken		
F14	hideKnappar	Normal	Knappar
	Göm knapparna		
F15	deletecontrolPanel	Normal	Minnestömning
	Radera denna instans av menyn		

I spelets centrala fil, vilken alltså består av spelets exteriörer, befinner sig också spelets huvudmeny eller gränssnitt. Menyn består av sex knappar. En knapp för avstägning av audio, en hjälpknapp, en knapp för information om livet förr, en knapp som visar uppdragen, en knapp som visar kartan över området och en knapp för att återgå till spelets startmeny. De enskillda delarna av detta gränssnitt var inte i sig svåra att programmera men eftersom knapparna alltid är synliga då spelaren befinner sig i exteriören behövdes ett relativt invecklat system för att förhindra knapparnas användning i olägliga situationer. T.ex. förhindras spelaren att ge förflyttningsinput efter att kartan tagits fram. Ett annat exempel är

objektens z-position, när man kontrollerar uppdragen bör uppdragsboken naturligtvis befinna sig framför alla andra objekt. Allt detta är småsaker, men de många kombinationerna som knapptryckningarna sammanlagt kunde leda till var många och en uteslutningmetod programmerades.

D.3.8 Programkod för klassen ControlPanel

```
/****   Klassen ControlPanel   ****/

package {
        import flash.display.*;
        import flash.events.*;
        import caurina.transitions.Tweener;

        public class ControlPanel extends MovieClip {

                // Konstanter.
                private static const ANTAL_KNAPPAR:int = 6;

                // Variabler.
                private var knappTabell:Array = new Array();
                public var mapActive:Boolean = false;
                public var nyMap:Stundarsmap = new Stundarsmap;
                public var youarehere:Youarehere = new Youarehere;
                public var uppdragActive:Boolean = false;
                public var uppdrag:Uppdrag = new Uppdrag;
                public var xuppdrag:Xknapp = new Xknapp;
                public var vissteActive:Boolean = false;
                public var helpActive:Boolean = false;
                public var onOffActive:Boolean = false;

                public function controlPanel() {

                        CreateMap();
                        CreateUppdrag();
                        CreateYouarehere();
                        Uppdragsprak();

                        MovieClip(parent).addChild(xuppdrag);
                        xuppdrag.buttonMode = true;
                        xuppdrag.gotoAndStop(1);
                        xuppdrag.x = 1111;
                        xuppdrag.y = 0;

                        nyMap.btnXsmap.addEventListener(MouseEvent.CLICK,clickbtnXsmap);
                        nyMap.btnXsmap.buttonMode = true;

                }

                private function CreateMap(){
                        MovieClip(parent).addChild(nyMap);
                        nyMap.x = 100;
                        nyMap.y = 1111;
                        nyMap.visible = false;
                }

                public function Uppdragsprak(){

                        if(MovieClip(parent).whatLanguage == 2){
                        uppdrag.txtupp1.text = "Munat voi myydä kauppapuodissa.";
                        uppdrag.txtupp2.text = "Osta kahvinpapuja ja riisiryynejä kaupasta.";
                        uppdrag.txtupp7.text = "Älä unohda että tänään on koulupäivä!";
                        uppdrag.txtupp4.text = "Hae jauhoa myllystä.";
                        uppdrag.txtupp5.text = "Kansanparantaja auttaa sairaita ihmisiä.";
                uppdrag.txtupp6.text = "Juhlaan keitetään puuroa ja leipiin tehdään reijät.";
                        uppdrag.txtupp3.text = "Vieraile maalaistalossa.";
```

```
        }else{
        uppdrag.txtupp1.text = "Äggen kan  säljas i lanthandeln.";
        uppdrag.txtupp2.text = "Handla kaffebönor och risgryn i lanthandeln.";
        uppdrag.txtupp7.text = "Glöm inte att det är skoldag idag!";
        uppdrag.txtupp4.text = "Hämta mjöl från kvarnen.";
        uppdrag.txtupp5.text = "Kloka gumman hjälper sjuka människor.";
        uppdrag.txtupp6.text = "Till festen ska gröten kokas och brödet naggas.";
        uppdrag.txtupp3.text = "Besök bondstugan.";
        }

    }

    private function CreateUppdrag(){
        MovieClip(parent).addChild(uppdrag);
        uppdrag.x = 1111;
        uppdrag.y = 0;
        uppdrag.visible = false;
        if(MovieClip(parent).whatLanguage == 2){
            uppdrag.gotoAndStop(2);
        }else{
            uppdrag.gotoAndStop(1);
        }
    }

    private function CreateYouarehere(){
        MovieClip(parent).addChild(youarehere);
        youarehere.x = 350;
        youarehere.y = 56;
        youarehere.visible = false;
        youarehere.youareHere()
    }


    public function Knapparna(){

        // Skapa knapparna.
        for(var i = 0; i < ANTAL_KNAPPAR; i++){

        var nyKnapp:ButtMap = new ButtMap;

        switch(i){
        case 0: //Ljud knappen.
        MovieClip(parent).addChild(nyKnapp);
        nyKnapp.x = 8;
        nyKnapp.y = 518;
        break;
        case 1: //Hjälp knappen.
        MovieClip(parent).addChild(nyKnapp);
        nyKnapp.x = 107;
        nyKnapp.y = 518;
        break;
        case 2: //”Visste du att” knappen.
        MovieClip(parent).addChild(nyKnapp);
        nyKnapp.x = 208;
        nyKnapp.y = 518;
        break;
        case 3: //Uppdragsknappen.
        MovieClip(parent).addChild(nyKnapp);
        nyKnapp.x = 303;
        nyKnapp.y = 518;
        break;
        case 4: //Kartknappen.
        MovieClip(parent).addChild(nyKnapp);
        nyKnapp.x = 405;
        nyKnapp.y = 518; //+ i * 90
        break;
        case 5: //Till meny-/stäng-knappen.
        MovieClip(parent).addChild(nyKnapp);
        nyKnapp.x = 816;
```

```
            nyKnapp.y = 518; //+ i * 90
            break;
            }

            nyKnapp.buttonMode = true;
            // Knappen har inga children (annars är knapptexten ivägen för själva
            knappen).
            nyKnapp.mouseChildren = false;

            // Sätt in nya knappen i en tabell för senare refferens
            knappTabell.push(nyKnapp);

            }

            // Sätt spelhuvudet på frame 1
            for(var j = 0; j < ANTAL_KNAPPAR; j++){
                    knappTabell[j].gotoAndStop(j+(1+j));

            namngeKnapparna();

            //Gör event listeners för alla knappar
            knappTabell[j].addEventListener(MouseEvent.CLICK,clickKnapp);
            knappTabell[j].addEventListener (MouseEvent.MOUSE_OVER,pekaKnapp);
            knappTabell[j].addEventListener (MouseEvent.MOUSE_OUT,dissaKnapp);
            }

    }

    //När en knapp blir tryckt loopa igenom och kolla vilken knapp som trycktes. Kalla
    sedan på sammanhängande funktion.
    private function clickKnapp(evt:MouseEvent):void{
                    for(var k = 0; k < ANTAL_KNAPPAR; k++){
                            if(evt.target == knappTabell[k]){
                            //event target = vilken knapp som trycktes.

                                    //Vilken knapp som blev tryckt
                                    switch(k){
                                    case 0://Ljud knappen.

                                    if(MovieClip(parent).ljudAv == false){
                                            MovieClip(parent).ljudAv = true;
                                            MovieClip(parent).stangLjud();
                                    }else{
                                            MovieClip(parent).ljudAv = false;
                                    }
                                    if(MovieClip(parent).ljudAv == false){
                                            knappTabell[0].gotoAndStop(1);
                                    }else{
                                            knappTabell[0].gotoAndStop(13);
                                    }
                                    break;
                                    case 1: //Hjälp knappen.
                                    if(uppdragActive == false && vissteActive == false &&
                                    mapActive == false && onOffActive == false &&
                                    helpActive == false &&
                                    MovieClip(parent).maria.harPratatMaria == true) {
                                    MovieClip(parent).showHelp();
                                    helpActive = true;
                                    MovieClip(parent).stopHero();
                                    MovieClip(parent).donotwalk = true;
                                    MovieClip(parent).fjarilen.orderFjaril = false;
                                            }
                                    break;
                                    case 2: //”Visste du att” knappen.
                                    trace("visste");
                                    if(vissteActive == false && mapActive == false &&
                                    uppdragActive == false && helpActive == false &&
                                    onOffActive == false &&
                                    MovieClip(parent).maria.harPratatMaria == true){
```

```
                                MovieClip(parent).moveVisste();
                                vissteActive = true;
                                MovieClip(parent).stopHero();
                                MovieClip(parent).donotwalk = true;
                                MovieClip(parent).fjarilen.orderFjaril = false;
                                }else if(vissteActive == true){
                                        MovieClip(parent).vissteduatt.x = 2222;
                                        MovieClip(parent).hideVisste();
                                        vissteActive = false;
                                        MovieClip(parent).doNotWalk();
                                        MovieClip(parent).donotwalk = false;
                                        MovieClip(parent).startHero();
                                        MovieClip(parent).fjarilen.orderFjaril = true;
                                }
                                break;
                                case 3: //Uppdragsknappen.
                                if(vissteActive == false && mapActive == false &&
                                helpActive == false && onOffActive == false &&
                                MovieClip(parent).maria.harPratatMaria == true){
                                uppdragsController();
                                MovieClip(parent).fjarilen.orderFjaril = false;
                                }

                                break;
                                case 4://Kartknappen.
                                if(uppdragActive == false && vissteActive == false &&
                                helpActive == false && onOffActive == false&&
                                MovieClip(parent).maria.harPratatMaria == true){
                                MovieClip(parent).fjarilen.orderFjaril = false;
                                mapController();
                                }
                                break;
                                case 5://Till menyn knappen.
                                if(onOffActive == false && uppdragActive == false &&
                                vissteActive == false && helpActive == false &&
                                mapActive == false){
                                MovieClip(parent).buttonEsc = true;
                                MovieClip(parent).fjarilen.orderFjaril = false;
                                }
                                break;
                                }
                        }
                }

        }
private function pekaKnapp(evt:MouseEvent):void{
                for(var l = 0; l < ANTAL_KNAPPAR; l++){
                        if(evt.target == knappTabell[l]){
//event target = vilken knapp som cursorn svävar över.
                                knappTabell[l].gotoAndStop((l+1)+(l+1));
                                namngeKnapparna();
                        }
                        if(evt.target == knappTabell[0]){
                                if(MovieClip(parent).ljudAv == false){
                                        knappTabell[0].gotoAndStop(2);
                                }else{
                                        knappTabell[0].gotoAndStop(14);
                                }
                        }
                }
        }
private function dissaKnapp(evt:MouseEvent):void{
                for(var m = 0; m < ANTAL_KNAPPAR; m++){
                        if(evt.target == knappTabell[m]){
//event target = vilken knapp som cursorn for bort från.
                                knappTabell[m].gotoAndStop(m+(1+m));
                                namngeKnapparna();
                        }
                        if(evt.target == knappTabell[0]){
```

```
                                if(MovieClip(parent).ljudAv == false){
                                        knappTabell[0].gotoAndStop(1);
                                }else{
                                        knappTabell[0].gotoAndStop(13);
                                }
                        }
                }
        }

//Aktivera eller stäng kartan över Stundars via kartknappen.
private function mapController(){
                if(mapActive == false){
                        MovieClip(parent).stopHero();
                        MovieClip(parent).donotwalk = true;
                        youarehere.switchYouarehere();
                        mapActive = true;
                        nyMap.visible = true;
                        nyMap.alpha = 0;
                        MovieClip(parent).ordnaUppgift();
                        Tweener.addTween(nyMap, {alpha:1, time:0.5});
                        youarehere.visible = true;
                        Tweener.addTween(nyMap, {y:40, time:0.5});
                }else if(mapActive == true){
                        mapActive = false;
                        nyMap.alpha = 0.8;
                        Tweener.addTween(nyMap, {y:1111, time:0.5});
                        Tweener.addTween(nyMap, {alpha:0, time:0.5});
                        youarehere.visible = false;
                        MovieClip(parent).startHero();
                        MovieClip(parent).donotwalk = false;
                }
}
//Stäng Kartan med kryss (stäng) knappen.
private function clickbtnXsmap(evt:MouseEvent):void{

                if(mapActive == true){
                        mapActive = false;
                        nyMap.alpha = 0.8;
                        Tweener.addTween(nyMap, {y:1111, time:0.5});
                        Tweener.addTween(nyMap, {alpha:0, time:0.5});
                        youarehere.visible = false;
                        MovieClip(parent).startHero();
                        MovieClip(parent).donotwalk = false;
                }

                MovieClip(parent).doNotWalk();
                MovieClip(parent).ReallydoNotWalk();

}
//Aktivera eller stäng uppdragsinterfacet med bok-knappen.
private function uppdragsController(){
                if(uppdragActive == false){
                        if(MovieClip(parent).whatLanguage == 2){
                                uppdrag.gotoAndStop(2);
                        }else{
                                uppdrag.gotoAndStop(1);
                        }
                        MovieClip(parent).stopHero();
                        MovieClip(parent).donotwalk = true;
                        uppdragActive = true;
                        uppdrag.visible = true;
                        uppdrag.alpha = 0;
                        MovieClip(parent).ordnaUppgift();
                        Tweener.addTween(uppdrag, {alpha:1, time:0.5});
                        xuppdrag.addEventListener(MouseEvent.CLICK,closeUppdrag);
                        uppdrag.x = 0;
                        xuppdrag.x = 900 - 38 - 10;
                        xuppdrag.y = 0 + 10;
                        MovieClip(parent).showUpBockar();
```

```
                        }else if(uppdragActive == true){

                        for(var n = 0; n < MovieClip(parent).tblUppbockar.length; n++){
                                MovieClip(parent).tblUppbockar[n].x = 2222;
                                MovieClip(parent).tblUppbockar[n].visible = false;
                        }
                        uppdragActive = false;
                        uppdrag.alpha = 1;
                        Tweener.addTween(uppdrag, {alpha:0, time:0.5});
                        uppdrag.x = 1111;
                        xuppdrag.x = 1111;
                        MovieClip(parent).startHero();
                        MovieClip(parent).donotwalk = false;

                        }
                }
                //Stäng uppdragsinterfacet med kryss (stäng) knappen.
                private function closeUppdrag(evt:MouseEvent):void{
                        uppdragActive = false;
                        uppdrag.alpha = 1;
                        Tweener.addTween(uppdrag, {alpha:0, time:0.5});
                        xuppdrag.removeEventListener(MouseEvent.CLICK,closeUppdrag);
                        uppdrag.x = 1111;
                        xuppdrag.x = 1111;
                        MovieClip(parent).startHero();
                        MovieClip(parent).donotwalk = false;

                        for(var n = 0; n < MovieClip(parent).tblUppbockar.length; n++){
                                MovieClip(parent).tblUppbockar[n].x = 2222;
                                MovieClip(parent).tblUppbockar[n].visible = false;
                        }

                }
                //Göm knappar
                public function hideKnappar(){
                        for(var r = 0; r < ANTAL_KNAPPAR; r++){
                                knappTabell[r].x = 2222;
                        }
                }
                //Radera menyn och frigör minne.
                public function deletecontrolPanel() {

                        if(uppdragActive == true){
                                xuppdrag.removeEventListener(MouseEvent.CLICK,closeUppdrag);
                        }

                        for(var j:int=knappTabell.length-1;j>=0;j--) {

                                knappTabell[j].removeEventListener(MouseEvent.CLICK,clickKnapp);
                                knappTabell[j].removeEventListener
                                (MouseEvent.MOUSE_OVER,pekaKnapp);
                                knappTabell[j].removeEventListener
                                (MouseEvent.MOUSE_OUT,dissaKnapp);
                                knappTabell.pop();
                        }

                        parent.removeChild(this);

                }

        }

}
```

D.4. ANVÄNDARGRÄNSSNITT

Bild 2. Smedens exteriör, spelets meny, rollfiguren Smedis Maria och spelaren.

Menyknappar

Audio	Stäng av eller lägg på audion
Frågetecken	Visar spelinstruktioner
Utropstecken	Visar information om livet förr
Bok	Visar speluppdrag
Kompass	Visar karta över området och var spelaren befinner sig
On-Off knapp	Frågar om spelaren vill avsluta spelet och gå till startmenyn

Tangentbord och Mus

Markör (cursor)	Visa vart spelaren vill gå. Utlöser pop-up fönster när spelaren pekar på vissa objekt, t.ex. dörrar.
Vänster musknapp	Knapp för handling (action button), t.ex. spelfiguren rör sig till markörens kordinater eller går in genom dörr och startar nytt delspel.
Piltangent vänster	Förflyttar spelfiguren till vänster
Piltangent höger	Förflyttar spelfiguren till höger
Piltangent upp	Förflyttar spelfiguren till uppåt
Piltangent ner	Förflyttar spelfiguren till neråt