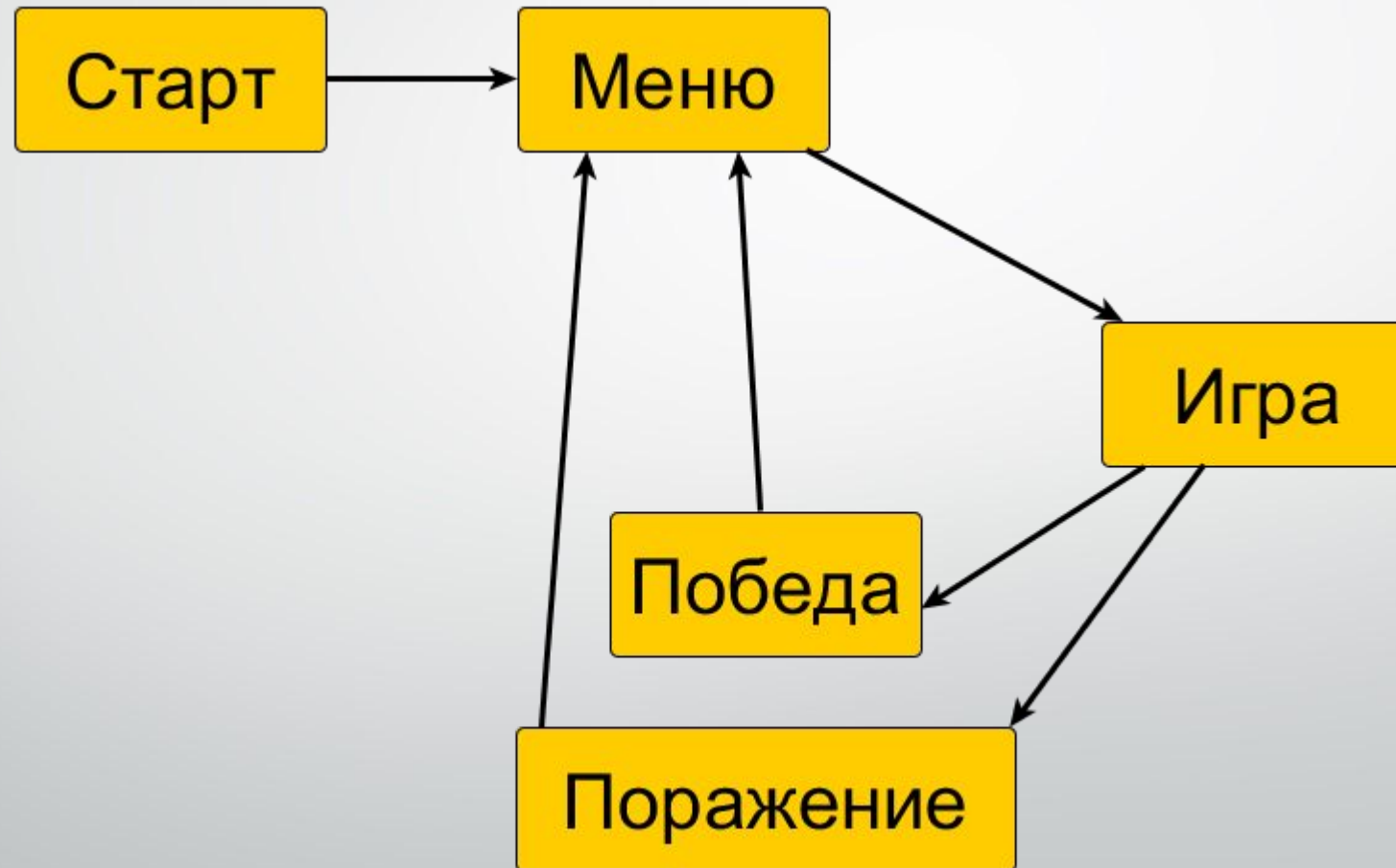




Игровой цикл.

Старт -> меню -> игра -> победа || поражение

Игровой цикл



Игровой цикл в коде

- Каждый из пунктов меню – отдельный экран.
- Каждый отдельный экран будет отдельным классом
- Main() никуда не денется, с него начнется работа программы, запустится цикл и будет выполнен переход к первому экрану.

Система евентов (событий)

Event

EnterFrame

MouseEvent

Click

MouseMove

KeyboardEvent

KeyUp

KeyDown

Свои собственные эвенты!

- Создаем переменную String и задаем ей какое-то значение. (например, `GOTO_MENU_STRING`) – она будет определять наш эвент:

```
var GOTO_MENU_STRING : String = "GAMEEVENT_GOTO_MENU";
```

- Создаем сам эвент. Делаем его `public static`, чтобы до него можно было добраться извне. Создаем эвент **СТРОГО** в классе Main.

```
public static var GAMEEVENT_GOTO_MENU: Event;  
GAMEEVENT_GOTO_MENU = new Event(GOTO_MENU_STRING);
```

- Установка листнера на этот эвент не меняется:

```
addEventListener(GOTO_MENU_STRING, GotoMenu);
```

- Этот эвент надо как-то генерировать, чтобы листнер на него среагировал. Делаем вот так:

```
dispatchEvent (Main.GAMEEVENT_GOTO_MENU);
```

Делаем классы-экраны.

- Сделаем наши эвенты

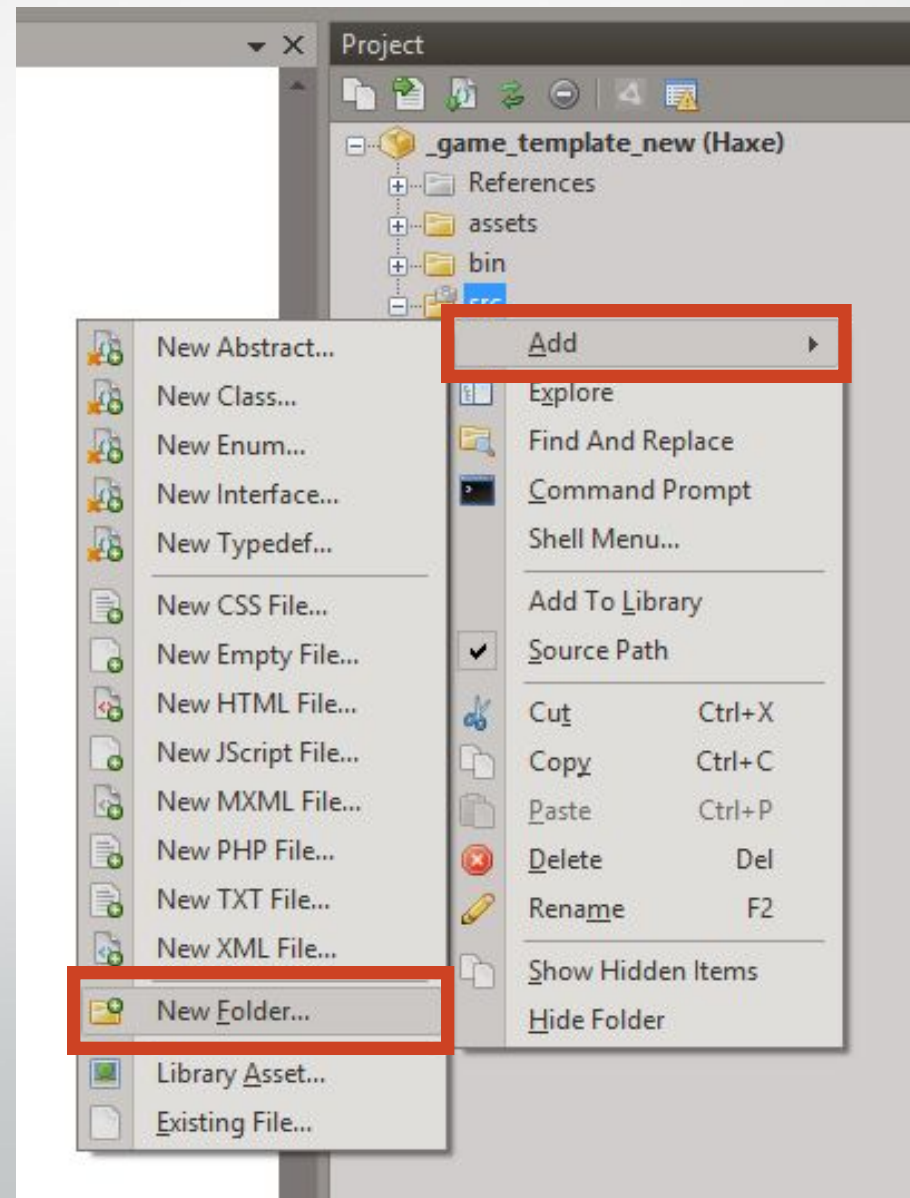
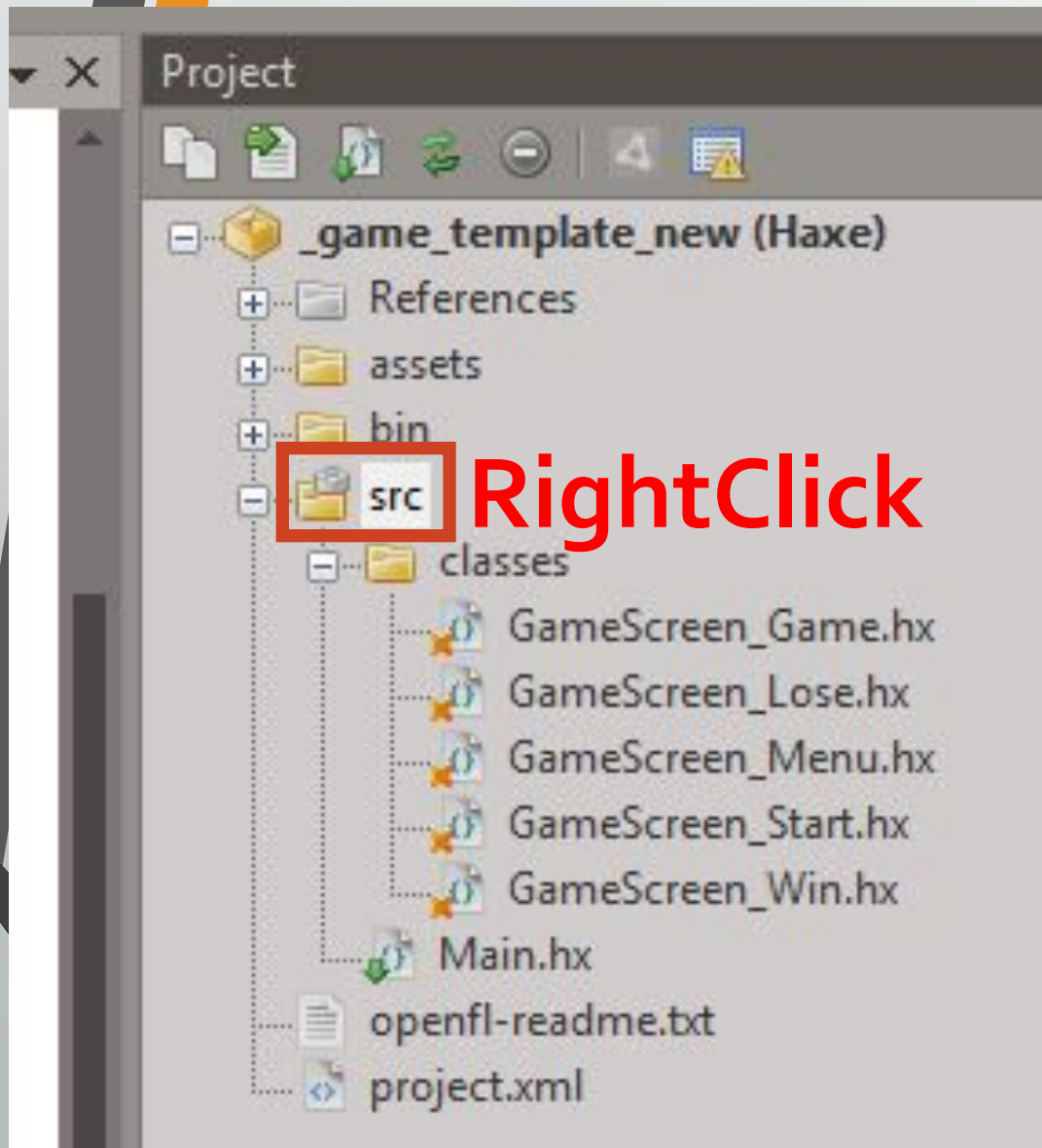
```
Main.hx*  GameScreen_Start.hx  GameScreen_Menu.hx  GameScreen_Game.hx  G
1  package;
2
3  import openfl.display.Sprite;
4  import openfl.Lib;
5  import openfl.events.Event;
6  import classes.*;
7
8  /**
9   * ...
10  * @author Zergling
11  */
12  class Main extends Sprite
13  {
14      // Stage size
15      public static var STAGE_WIDTH: Float = 800;
16      public static var STAGE_HEIGHT: Float = 480;
17
18      // Game Events
19      var GOTO_MENU_STRING : String = "GAMEEVENT_GOTO_MENU";
20      var GOTO_GAME_STRING : String = "GAMEEVENT_GOTO_GAME";
21      var GOTO_WIN_STRING : String = "GAMEEVENT_GOTO_WIN";
22      var GOTO_LOSE_STRING : String = "GAMEEVENT_GOTO_LOSE";
23
24      public static var GAMEEVENT_GOTO_MENU: Event;
25      public static var GAMEEVENT_GOTO_GAME: Event;
26      public static var GAMEEVENT_GOTO_LOSE: Event;
27      public static var GAMEEVENT_GOTO_WIN: Event;
28
```

```
28
29
30  public function new() *
31  {
32      super(); **
33
34      // Assets:
35      // openfl.Assets.getBitmapData("img/assetname.jpg");
36
37      GAMEEVENT_GOTO_MENU = new Event(GOTO_MENU_STRING);
38      GAMEEVENT_GOTO_GAME = new Event(GOTO_GAME_STRING);
39      GAMEEVENT_GOTO_LOSE = new Event(GOTO_LOSE_STRING);
40      GAMEEVENT_GOTO_WIN = new Event(GOTO_WIN_STRING);
41  }
42
43
```

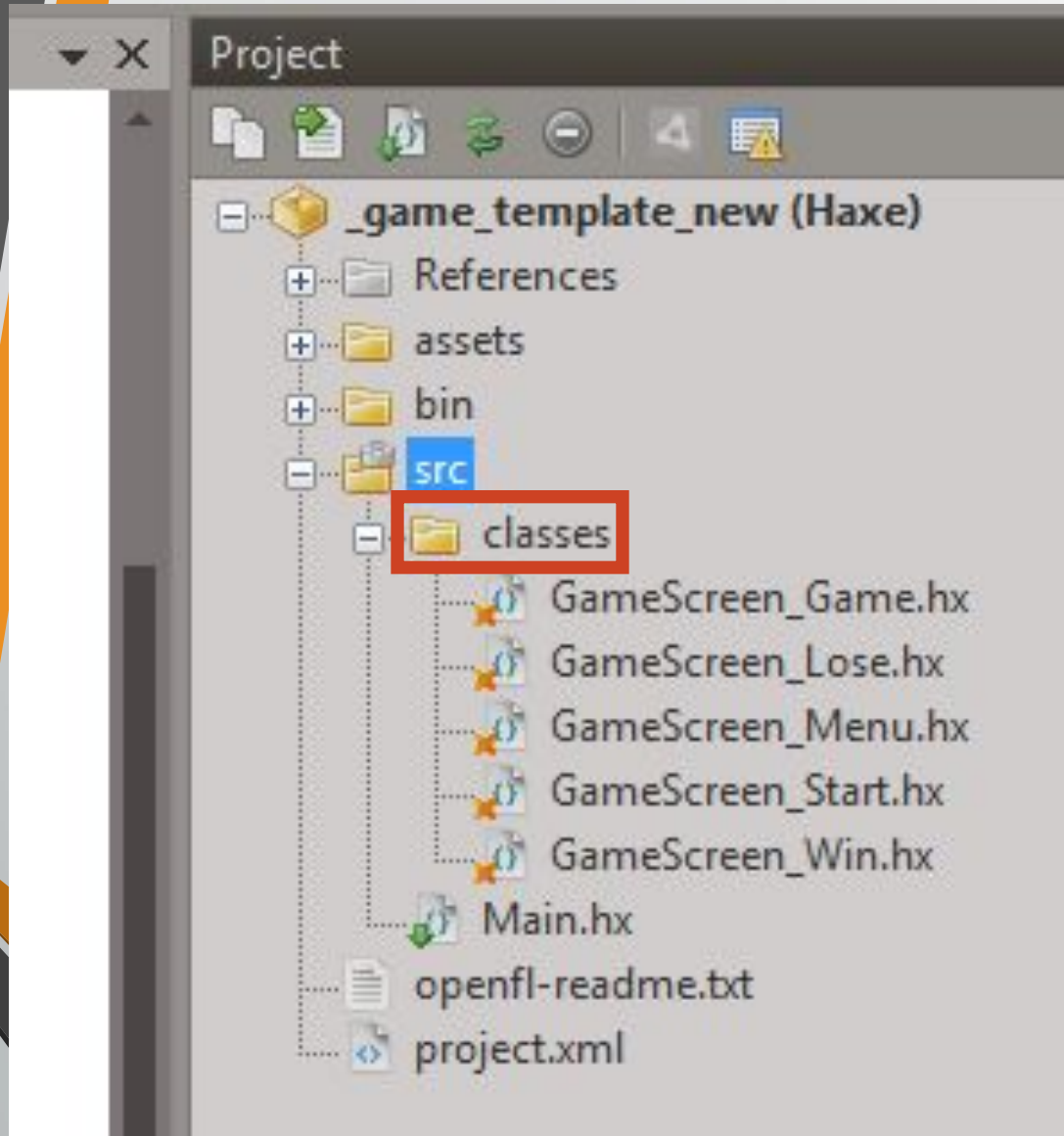
(*) в старом НаХе не new(), а
init()

(**) если тут что-то еще
написано – УДАЛЯТЬ НЕ НАДО!

Делаем классы-экраны



Делаем классы-экраны



Дальше на эту папку правой кнопкой -> Add -> New Class. Назвать можно как угодно.

Делаем классы-экраны.

- После этих манипуляций в верхушке каждого класса-экрана будет написано
`package classes;`
- В импорты класса Main добавляем строку
`import classes.*;`
- Это сделает импорт всего содержимого папки `classes`;

Делаем классы-экраны.

- Опишем класс-экран Старт

```
Main.hx*  GameScreen_Start.hx  GameScreen_Menu.hx  GameScreen_Game.hx  GameSc
1  package classes;
2  import openfl.display.Sprite;
3  import openfl.events.MouseEvent;
4  |
5  class GameScreen_Start extends Sprite
6  {
7      var Background: Sprite;
8      var MenuButton: Sprite;
9
10     public function new()
11     {
12         super();
13
14         Background = new Sprite();
15         Background.graphics.beginFill(0x660000);
16         Background.graphics.drawRect(0, 0, 1000, 1000);
17         addChild(Background);
18
19         MenuButton = new Sprite();
20         MenuButton.graphics.beginFill(0xffffffff);
21         MenuButton.graphics.drawCircle(0, 0, 25);
22         MenuButton.x = Main.STAGE_WIDTH / 2; // 400
23         MenuButton.y = Main.STAGE_HEIGHT / 2; // 240
24         MenuButton.addEventListener(MouseEvent.CLICK, GotoMenu);
25         addChild(MenuButton);
26     }
27
28     public function GotoMenu(e: MouseEvent)
29     {
30         dispatchEvent (Main.GAMEEVENT_GOTO_MENU);
31     }
32
33
34 }
```

Делаем классы-экраны.

- Опишем класс-экран Меню

```
Main.hx*  GameScreen_Start.hx  GameScreen_Menu.hx  GameScreen_Game.hx  GameS
1  package classes;
2  import openfl.display.Sprite;
3  import openfl.events.MouseEvent;
4
5  class GameScreen_Menu extends Sprite
6  {
7      var Background: Sprite;
8      var MenuButton: Sprite;
9
10     public function new()
11     {
12         super();
13
14         Background = new Sprite();
15         Background.graphics.beginFill(0x660066);
16         Background.graphics.drawRect(0, 0, 1000, 1000);
17         addChild(Background);
18
19         MenuButton = new Sprite();
20         MenuButton.graphics.beginFill(0xffffffff);
21         MenuButton.graphics.drawCircle(0, 0, 25);
22         MenuButton.x = Main.STAGE_WIDTH / 2; // 400
23         MenuButton.y = Main.STAGE_HEIGHT / 2; // 240
24         MenuButton.addEventListener(MouseEvent.CLICK, GotoGame);
25         addChild(MenuButton);
26     }
27
28     public function GotoGame(e: MouseEvent)
29     {
30         dispatchEvent (Main.GAMEEVENT_GOTO_GAME);
31     }
32
33 }
```

Делаем классы-экраны.

- Создадим эти экраны и свяжем их при помощи нашего эвента (все делается внутри класса Main)

GAMEEVENT_GOTO_MENU

```
24     public static var GAMEEVENT_GOTO_MENU: Event;  
25     public static var GAMEEVENT_GOTO_GAME: Event;  
26     public static var GAMEEVENT_GOTO_LOSE: Event;  
27     public static var GAMEEVENT_GOTO_WIN: Event;  
28  
29     var Start: GameScreen_Start;  
30     var Menu: GameScreen_Menu;  
31
```

```
38  
39     GAMEEVENT_GOTO_MENU = new Event(GOTO_MENU_STRING);  
40     GAMEEVENT_GOTO_GAME = new Event(GOTO_GAME_STRING);  
41     GAMEEVENT_GOTO_LOSE = new Event(GOTO_LOSE_STRING);  
42     GAMEEVENT_GOTO_WIN = new Event(GOTO_WIN_STRING);  
43  
44     Start = new GameScreen_Start();  
45     Start.addEventListener(GAMEEVENT_GOTO_MENU.type, GotoMenu);  
46     addChild(Start);  
47  
48     Menu = new GameScreen_Menu();  
49 }  
50
```

```
55  
56     {  
57         Start.visible = false;  
58         addChild(Menu);  
59     }  
60
```

- Опишем класс-экран Игра

```
Main.hx  GameScreen_Game.hx  GameScreen_Start.hx  GameScreen_Menu.hx  GameScreen_Wi
1  package classes;
2  import openfl.display.Sprite;
3  import openfl.events.MouseEvent;
4
5  class GameScreen_Game extends Sprite
6  {
7      var Background: Sprite;
8      var MenuButton_Win: Sprite;
9      var MenuButton_Lose: Sprite;
10
11     public function new()
12     {
13         super();
14
15         Background = new Sprite();
16         Background.graphics.beginFill(0x000066);
17         Background.graphics.drawRect(0, 0, 1000, 1000);
18         addChild(Background);
19
20         MenuButton_Win = new Sprite();
21         MenuButton_Win.graphics.beginFill(0x00ff00);
22         MenuButton_Win.graphics.drawCircle(0, 0, 25);
23         MenuButton_Win.x = Main.STAGE_WIDTH / 2 - 100; // 300
24         MenuButton_Win.y = Main.STAGE_HEIGHT / 2; // 240
25         MenuButton_Win.addEventListener(MouseEvent.CLICK, GotoWin);
26         addChild(MenuButton_Win);
27
28         MenuButton_Lose = new Sprite();
29         MenuButton_Lose.graphics.beginFill(0xff0000);
30         MenuButton_Lose.graphics.drawCircle(0, 0, 25);
31         MenuButton_Lose.x = Main.STAGE_WIDTH / 2 + 100; // 500
32         MenuButton_Lose.y = Main.STAGE_HEIGHT / 2; // 140
33         MenuButton_Lose.addEventListener(MouseEvent.CLICK, GotoLose);
34         addChild(MenuButton_Lose);
35     }
36
37     function GotoWin(e: MouseEvent)
38     {
39         dispatchEvent (Main.GAMEEVENT_GOTO_WIN);
40     }
41
42     function GotoLose(e: MouseEvent)
43     {
44         dispatchEvent (Main.GAMEEVENT_GOTO_LOSE);
45     }
46
47 }
```

Делаем классы-экраны.

- Создадим экран Игры и свяжем его с экраном Меню эвентом

GAMEEVENT GOTO GAME

```
47 Start = new GameScreen_Start();
48 Start.addListener(GAMEEVENT_GOTO_MENU.type, GotoMenu);
49 addChild(Start);
50
51 Menu = new GameScreen_Menu();
52 Menu.addListener(GAMEEVENT_GOTO_GAME.type, GotoGame);
53
54 Game = null;
55 }
56
57 public function GotoMenu(e: Event)
58 {
59     Start.visible = false;
60     addChild(Menu);
61 }
62
63 public function GotoGame(e: Event)
64 {
65     removeChild(Menu);
66     if (Game == null)
67     {
68         Game = new GameScreen_Game();
69         addChild(Game);
70     }
71 }
72
```

```
23
24 public static var GAMEEVENT_GOTO_MENU: Event;
25 public static var GAMEEVENT_GOTO_GAME: Event;
26 public static var GAMEEVENT_GOTO_LOSE: Event;
27 public static var GAMEEVENT_GOTO_WIN: Event;
28
29 var Start: GameScreen_Start;
30 var Menu: GameScreen_Menu;
31 var Game: GameScreen_Game;
32
```

Делаем классы-экраны.

- Классы-экраны Победа и Поражение **В ДАННОМ СЛУЧАЕ** полностью повторяют класс-экран Старт. Свяжем экран Игра с ними.

```
23
24     public static var GAMEEVENT_GOTO_MENU: Event;
25     public static var GAMEEVENT_GOTO_GAME: Event;
26     public static var GAMEEVENT_GOTO_LOSE: Event;
27     public static var GAMEEVENT_GOTO_WIN: Event;
28
29     var Start: GameScreen_Start;
30     var Menu: GameScreen_Menu;
31     var Game: GameScreen_Game;
32     var Lose: GameScreen_Lose;
33     var Win: GameScreen_Win;
34
35     public function new()
36     {
37         super();
38
39         // Assets:
40         // openfl.Assets.getBitmapData("img/assetname.jpg");
41
42         GAMEEVENT_GOTO_MENU = new Event(GOTO_MENU_STRING);
43         GAMEEVENT_GOTO_GAME = new Event(GOTO_GAME_STRING);
44         GAMEEVENT_GOTO_LOSE = new Event(GOTO_LOSE_STRING);
45         GAMEEVENT_GOTO_WIN = new Event(GOTO_WIN_STRING);
46
47         Start = new GameScreen_Start();
48         Start.addEventListener(GAMEEVENT_GOTO_MENU.type, GotoMenu);
49         addChild(Start);
50
51         Menu = new GameScreen_Menu();
52         Menu.addEventListener(GAMEEVENT_GOTO_GAME.type, GotoGame);
53
54         Game = null;
55
56         Lose = new GameScreen_Lose();
57
58         Win = new GameScreen_Win();
59     }
60
```

```
67 public function GotoGame(e: Event)
68 {
69     removeChild(Menu);
70     if (Game == null)
71     {
72         Game = new GameScreen_Game();
73         Game.addEventListener(GAMEEVENT_GOTO_WIN.type, GotoWin);
74         Game.addEventListener(GAMEEVENT_GOTO_LOSE.type, GotoLose);
75         addChild(Game);
76     }
77 }
78
79 public function GotoWin(e: Event)
80 {
81     if (Game != null)
82     {
83         removeChild(Game);
84         Game.removeEventListener(GAMEEVENT_GOTO_WIN.type, GotoWin);
85         Game.removeEventListener(GAMEEVENT_GOTO_LOSE.type, GotoLose);
86         Game = null;
87     }
88     addChild(Win);
89 }
90
91 public function GotoLose(e: Event)
92 {
93     if (Game != null)
94     {
95         removeChild(Game);
96         Game.removeEventListener(GAMEEVENT_GOTO_WIN.type, GotoWin);
97         Game.removeEventListener(GAMEEVENT_GOTO_LOSE.type, GotoLose);
98         Game = null;
99     }
100     addChild(Lose);
101 }
102
```


Делаем классы-экраны.

- Последний штрих:
связываем экраны
Победа и Поражение с
экраном Меню

```
55  
56 Lose = new GameScreen_Lose();  
57 Lose.addEventListener(GAMEEVENT_GOTO_MENU.type, GotoMenu_Lose);  
58  
59 Win = new GameScreen_Win();  
60 Win.addEventListener(GAMEEVENT_GOTO_MENU.type, GotoMenu_Win);  
61 }
```

```
104  
105 public function GotoMenu_Lose(e: Event)  
106 {  
107     removeChild(Lose);  
108     addChild(Menu);  
109 }  
110  
111 public function GotoMenu_Win(e: Event)  
112 {  
113     removeChild(Win);  
114     addChild(Menu);  
115 }  
116
```

Итог.

- Переключение между Игрой и Поражением (Победой) **В ДАННОМ СЛУЧАЕ** получилось ручное (по клику мыши).
- При создании игры достаточно внутри класса GameScreen_Game достаточно будет в нужный момент вызывать функции GotoWin и GotoLose, предварительно сделав их безаргументными (очистить содержимое круглых скобок в заголовке).
- Сама игра создается во время обработки эвента GAMEEVENT_GOTO_GAME и убивается при обработке эвентов GAMEEVENT_GOTO_WIN и GAMEEVENT_GOTO_LOSE, что гарантирует полное пересоздание игры (каждый раз будет начинать именно новую игру и если там используются какие-то алгоритмы генерации – они будут выполняться заново)

Задание

- Сделать игровой цикл. Проверить, что все работает.
- Подумать, что будем делать на следующем занятии:
 - Продолжим изучать ООП
 - Научимся делать эффекты при помощи системы частиц
 - Повторим, как делаются всякие штуки, которые делали до этого
 - Свой вариант: _____