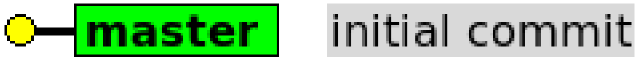
Open-source programming
Exercise 4

Git merge

Rostislav Lisovy
lisovy@gmail.com
12.3.2014

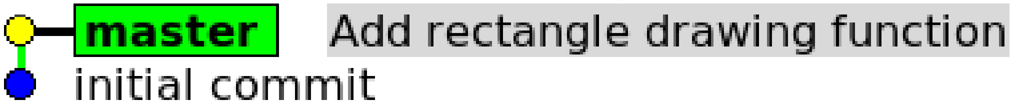
Create repository, first commit

```
$ mkdir osp_git
$ cd osp_git
$ git init .
Initialized empty Git repository in /tmp/osp_git/.git/
$ vim main.c
$ cat main.c
/* Magic Drawing Library version 0.0.1 */

#include <stdint.h>

$ git add main.c
$ git commit -m "initial commit"
$ gitk
```



Commit 2 – "Add rectangle drawing function"

```
$ vim main.c
$ git add main.c
$ git commit -m "Add rectangle drawing function"
$ gitk
```



Add rectangle drawing function

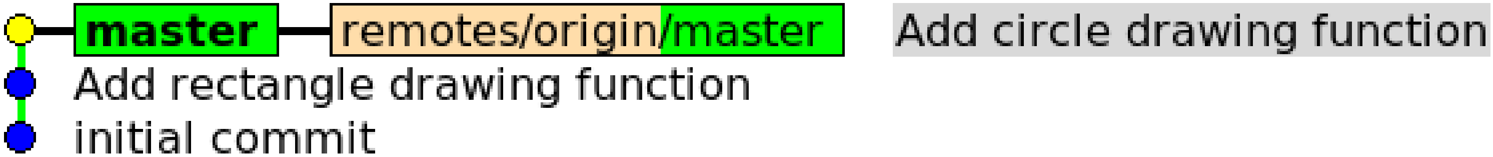
initial commit

Commit 2 – "Add rectangle drawing function"

```
$ git diff HEAD^
diff --git a/main.c b/main.c
index df6d071..d154662 100644
--- a/main.c
+++ b/main.c
@@ -2,3 +2,14 @@

 #include <stdint.h>

+struct mdl_rectangle {
+       uint32_t x;
+       uint32_t y;
+       uint32_t width;
+       uint32_t height;
+};
+
+void mdl_draw_rectangle(struct mdl_rectangle *rect, int flags)
+{
+       // Do something ...
+}
```

Commit 3 – "Add circle drawing function"

```
$ vim main.c
$ git add main.c
$ git commit -m "Add circle drawing function"
$ git remote add origin git@github.com:lisovy/mdl.git
$ git push origin master
The authenticity of host 'github.com (192.30.252.131)' can't be established.
RSA key fingerprint is 16:27:ac:a5:76:28:2d:36:63:1b:56:4d:eb:df:a6:48.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,192.30.252.131' (RSA) to the list of
known hosts.
Counting objects: 9, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 886 bytes, done.
Total 9 (delta 2), reused 0 (delta 0)
To git@github.com:lisovy/mdl.git
 * [new branch]      master -> master
$ gitk
```



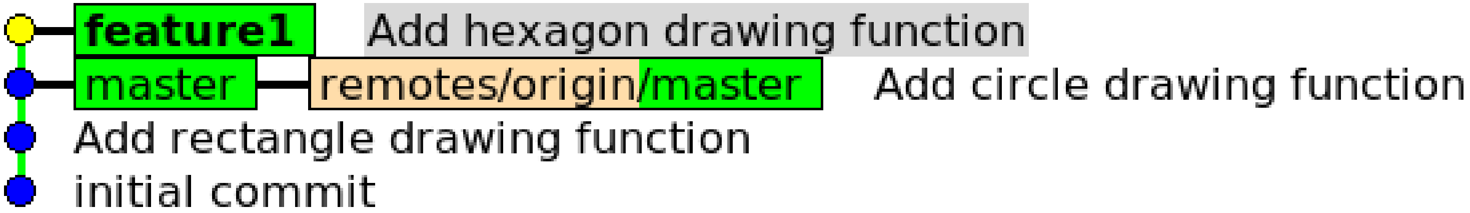
```
master  remotes/origin/master  Add circle drawing function
Add rectangle drawing function
initial commit
```

Commit 3 – "Add circle drawing function"

```
$ git diff HEAD^
diff --git a/main.c b/main.c
index d154662..be70975 100644
--- a/main.c
+++ b/main.c
@@ -9,7 +9,18 @@ struct mdl_rectangle {
        uint32_t height;
 };

+struct mdl_circle {
+       uint32_t x; /* Centre */
+       uint32_t y; /* Centre */
+       uint32_t radius;
+}
+
 void mdl_draw_rectangle(struct mdl_rectangle *rect, int flags)
 {
        // Do something ...
 }
+
+void mdl_draw_circle(struct mdl_circle *circ, int flags)
+{
+       // Do something ...
+}
```

feature1 branch
"Add hexagon drawing function"

```
$ git checkout -b feature1
$ vim main.c
$ git add main.c
$ git commit -m "Add hexagon drawing function"
$ gitk
```



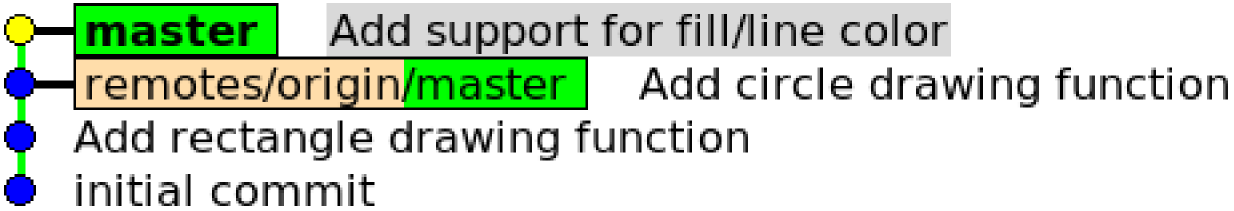
```
feature1    Add hexagon drawing function
master    remotes/origin/master    Add circle drawing function
Add rectangle drawing function
initial commit
```

feature1 branch
"Add hexagon drawing function"

```
$ git diff HEAD^
diff --git a/main.c b/main.c
index be70975..7b732b3 100644
--- a/main.c
+++ b/main.c
@@ -15,6 +15,13 @@ struct mdl_circle {
        uint32_t radius;
 }

+struct mdl_hexagon {
+       uint32_t x; /* Centre */
+       uint32_t y; /* Centre */
+       uint32_t radius;
+       uint32_t rotation;
+}
+
 void mdl_draw_rectangle(struct mdl_rectangle *rect, int flags)
 {
        // Do something ...
@@ -24,3 +31,8 @@ void mdl_draw_circle(struct mdl_circle *circ, int flags)
 {
        // Do something ...
 }
+
+void mdl_draw_hexagon(struct mdl_hexagon *hex, int flags)
+{
+       // Do something ...
+}
```

Back to the master
"Add support for fill/line color"

```
$ git checkout master
$ vim main.c
$ git add main.c
$ git commit -m "Add support for fill/line color"
$ gitk
```



master Add support for fill/line color

remotes/origin/master Add circle drawing function

Add rectangle drawing function

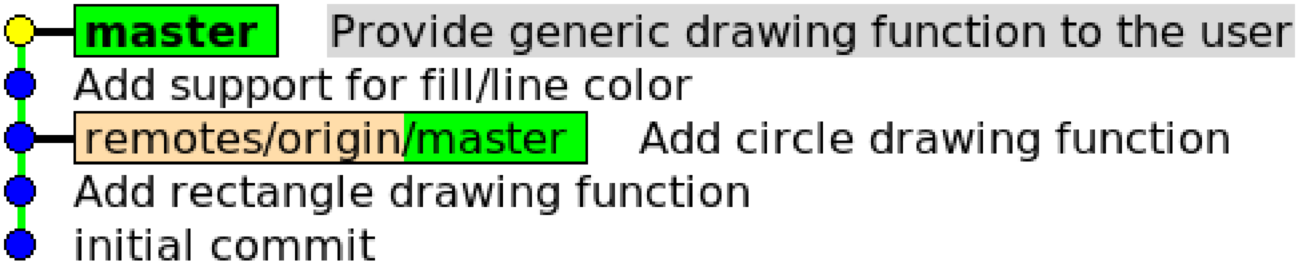
initial commit

"Add support for fill/line color"

```
$ git diff HEAD^
diff --git a/main.c b/main.c
index be70975..9a1efa6 100644
--- a/main.c
+++ b/main.c
@@ -7,12 +7,16 @@ struct mdl_rectangle {
        uint32_t y;
        uint32_t width;
        uint32_t height;
+       uint32_t color_fill;
+       uint32_t color_line;
 };

 struct mdl_circle {
        uint32_t x; /* Centre */
        uint32_t y; /* Centre */
        uint32_t radius;
+       uint32_t color_fill;
+       uint32_t color_line;
 };
```

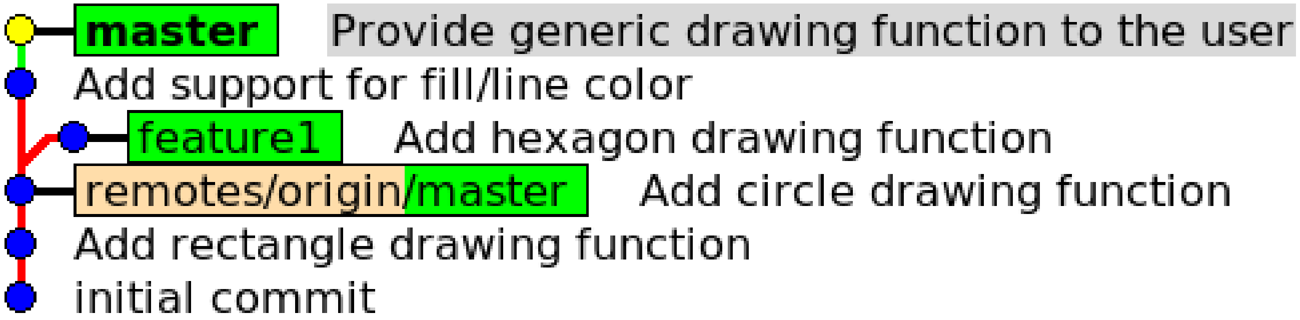
"Provide generic drawing function to the user"

```
$ vim main.c
$ git add main.c
$ git commit -m "Provide generic drawing function to the user"
$ gitk
```



master Provide generic drawing function to the user
Add support for fill/line color
remotes/origin/master Add circle drawing function
Add rectangle drawing function
initial commit

```
$ gitk --all
```



master Provide generic drawing function to the user
Add support for fill/line color
feature1 Add hexagon drawing function
remotes/origin/master Add circle drawing function
Add rectangle drawing function
initial commit

"Provide generic drawing function to the user"

```
$ git diff HEAD^
diff --git a/main.c b/main.c
index 9a1efa6..c7722c7 100644
--- a/main.c
+++ b/main.c
@@ -2,6 +2,11 @@

 #include <stdint.h>

+enum mdl_object {
+       MDL_RECTANGLE,
+       MDL_CIRCLE,
+};
+
 struct mdl_rectangle {
        uint32_t x;
        uint32_t y;
@@ -19,12 +24,24 @@ struct mdl_circle {
        uint32_t color_line;
 }

-void mdl_draw_rectangle(struct mdl_rectangle *rect, int flags)
+static void mdl_draw_rectangle(struct mdl_rectangle *rect, int flags)
 {
        // Do something ...
 }

-void mdl_draw_circle(struct mdl_circle *circ, int flags)
+static void mdl_draw_circle(struct mdl_circle *circ, int flags)
 {
        // Do something ...
 }
+
+void mdl_draw(void *object, enum mdl_object obj_type, int flags)
+{
+       switch (obj_type) {
+       case MDL_RECTANGLE:
+               mdl_draw_rectangle((struct mdl_rectangle*) object, flags);
+               break;
+       case MDL_CIRCLE:
+                mdl_draw_circle((struct mdl_circle*) object, flags);
+               break;
+       }
+}
```

Merge **feature1** to **master**

```
$ git merge feature1
Auto-merging main.c
CONFLICT (content): Merge conflict in main.c
Automatic merge failed; fix conflicts and then commit the result.
$ vim main.c # solve the conflict by hand
```

```
File structure:
  <<<<<<< HEAD
+ functionality present in the master branch
  =======
+ functionality added by the feature1 branch
  >>>>>>> feature1
```

Correct solution?

```
/* Magic Drawing Library version 0.0.1 */

#include <stdint.h>

enum mdl_object {
    MDL_RECTANGLE,
    MDL_CIRCLE,
};

struct mdl_rectangle {
    uint32_t x;
    uint32_t y;
    uint32_t width;
    uint32_t height;
    uint32_t color_fill;
    uint32_t color_line;
};

struct mdl_circle {
    uint32_t x; /* Centre */
    uint32_t y; /* Centre */
    uint32_t radius;
    uint32_t color_fill;
    uint32_t color_line;
};

struct mdl_hexagon {
    uint32_t x; /* Centre */
    uint32_t y; /* Centre */
    uint32_t radius;
    uint32_t rotation;
}
```

```
void mdl_draw_rectangle(
        struct mdl_rectangle *rect,
        int flags)
{
    // Do something ...
}

static void mdl_draw_circle(
        struct mdl_circle *circ,
        int flags)
{
    // Do something ...
}

void mdl_draw(
        void *object,
        enum mdl_object obj_type,
        int flags)
{
    switch (obj_type) {
    case MDL_RECTANGLE:
        mdl_draw_rectangle(
        (struct mdl_rectangle*)object,
        flags);
        break;
    case MDL_CIRCLE:
        mdl_draw_circle(
        (struct mdl_circle*)object,
        flags);
        break;
    }
}

void mdl_draw_hexagon(
        struct mdl_hexagon *hex,
        int flags)
{
    // Do something ...
}
```

Much better

```
/* Magic Drawing Library version 0.0.1 */

#include <stdint.h>

enum mdl_object {
    MDL_RECTANGLE,
    MDL_CIRCLE,
    MDL_HEXAGON,
};

struct mdl_rectangle {
    uint32_t x;
    uint32_t y;
    uint32_t width;
    uint32_t height;
    uint32_t color_fill;
    uint32_t color_line;
};

struct mdl_circle {
    uint32_t x; /* Centre */
    uint32_t y; /* Centre */
    uint32_t radius;
    uint32_t color_fill;
    uint32_t color_line;
};

struct mdl_hexagon {
    uint32_t x; /* Centre */
    uint32_t y; /* Centre */
    uint32_t radius;
    uint32_t rotation;
}
```

```
void mdl_draw_rectangle(
        struct mdl_rectangle *rect,
        int flags)
{
    // Do something ...
}

static void mdl_draw_circle(
        struct mdl_circle *circ,
        int flags)
{
    // Do something ...
}

static void mdl_draw_hexagon(
        struct mdl_hexagon *hex,
        int flags)
{
    // Do something ...
}

void mdl_draw(
        void *object,
        enum mdl_object obj_type,
        int flags)
{
    switch (obj_type) {
    case MDL_RECTANGLE:
        mdl_draw_rectangle(
            (struct mdl_rectangle*)object,
            flags);
        break;
    case MDL_CIRCLE:
        mdl_draw_circle(
            (struct mdl_circle*)object,
            flags);
        break;
    case MDL_HEXAGON:
        mdl_draw_hexagon(
            (struct mdl_hexagon*)object,
            flags);
        break;
    }
}
```

Conclusion

Read the code

Understand the code

Modify the code **by hand**