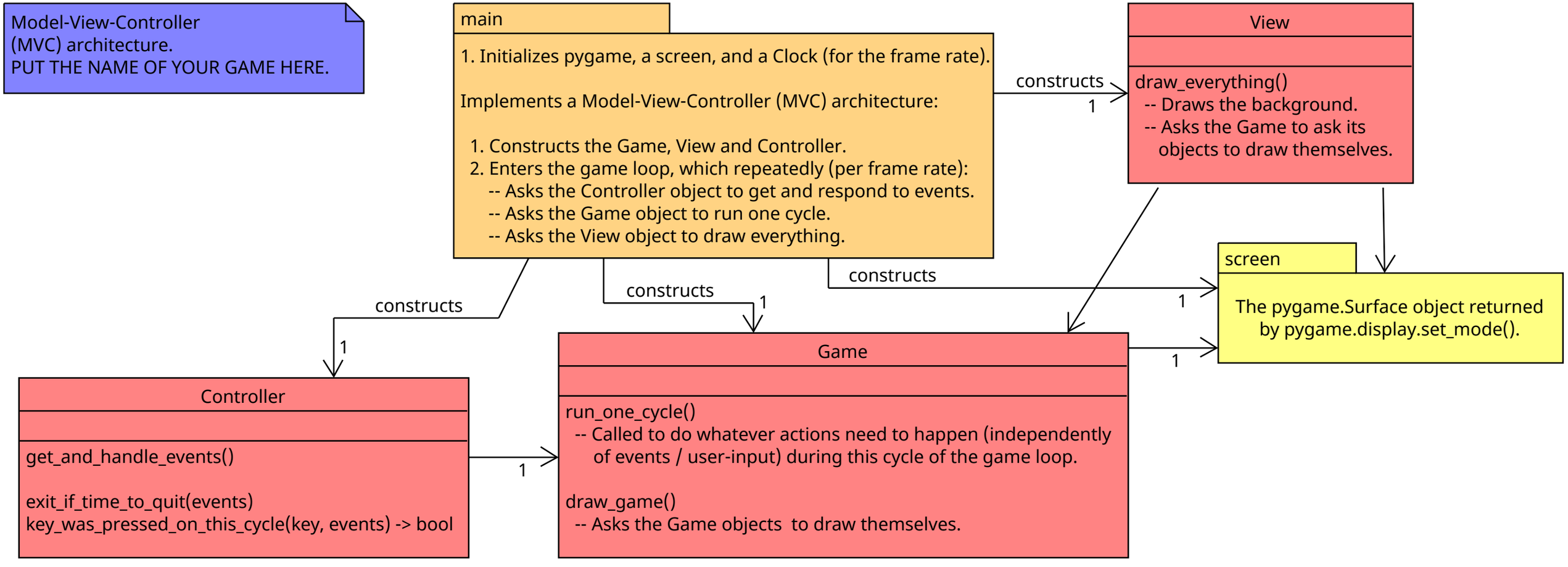
Model-View-Controller
(MVC) architecture.
PUT THE NAME OF YOUR GAME HERE.
main
1. Initializes pygame, a screen, and a Clock (for the frame rate).
Implements a Model-View-Controller (MVC) architecture:
1. Constructs the Game, View and Controller.
2. Enters the game loop, which repeatedly (per frame rate):
-- Asks the Controller object to get and respond to events.
-- Asks the Game object to run one cycle.
-- Asks the View object to draw everything.
constructs
1
View
draw_everything()
-- Draws the background.
-- Asks the Game to ask its
objects to draw themselves.
constructs
1
constructs
1
constructs
1
screen
The pygame.Surface object returned
by pygame.display.set_mode().
1
Controller
get_and_handle_events()
exit_if_time_to_quit(events)
key_was_pressed_on_this_cycle(key, events) -> bool
1
Game
run_one_cycle()
-- Called to do whatever actions need to happen (independently
of events / user-input) during this cycle of the game loop.
draw_game()
-- Asks the Game objects to draw themselves.