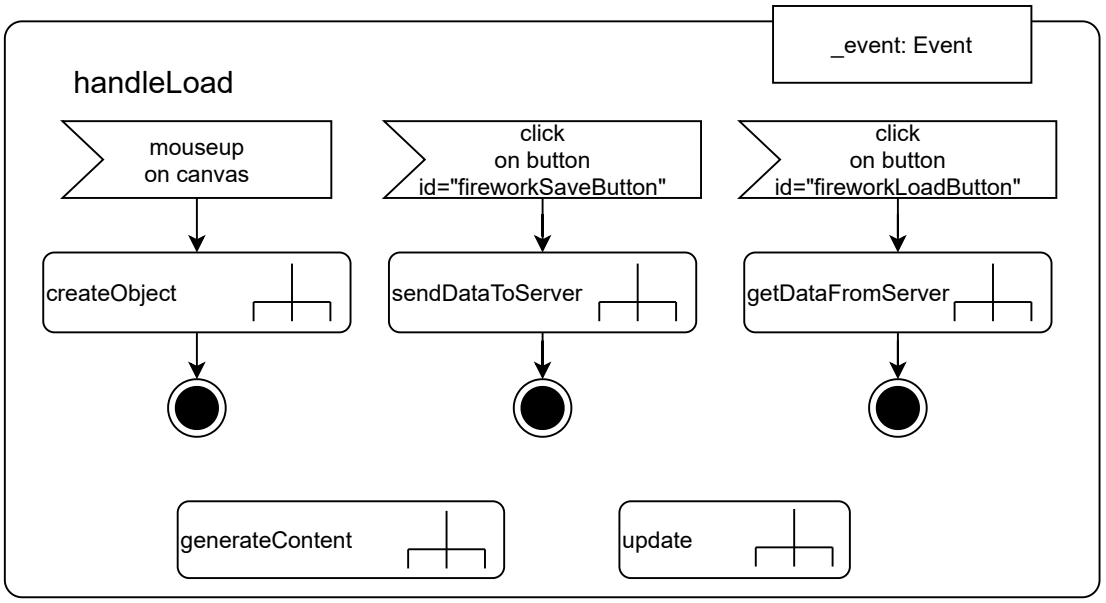
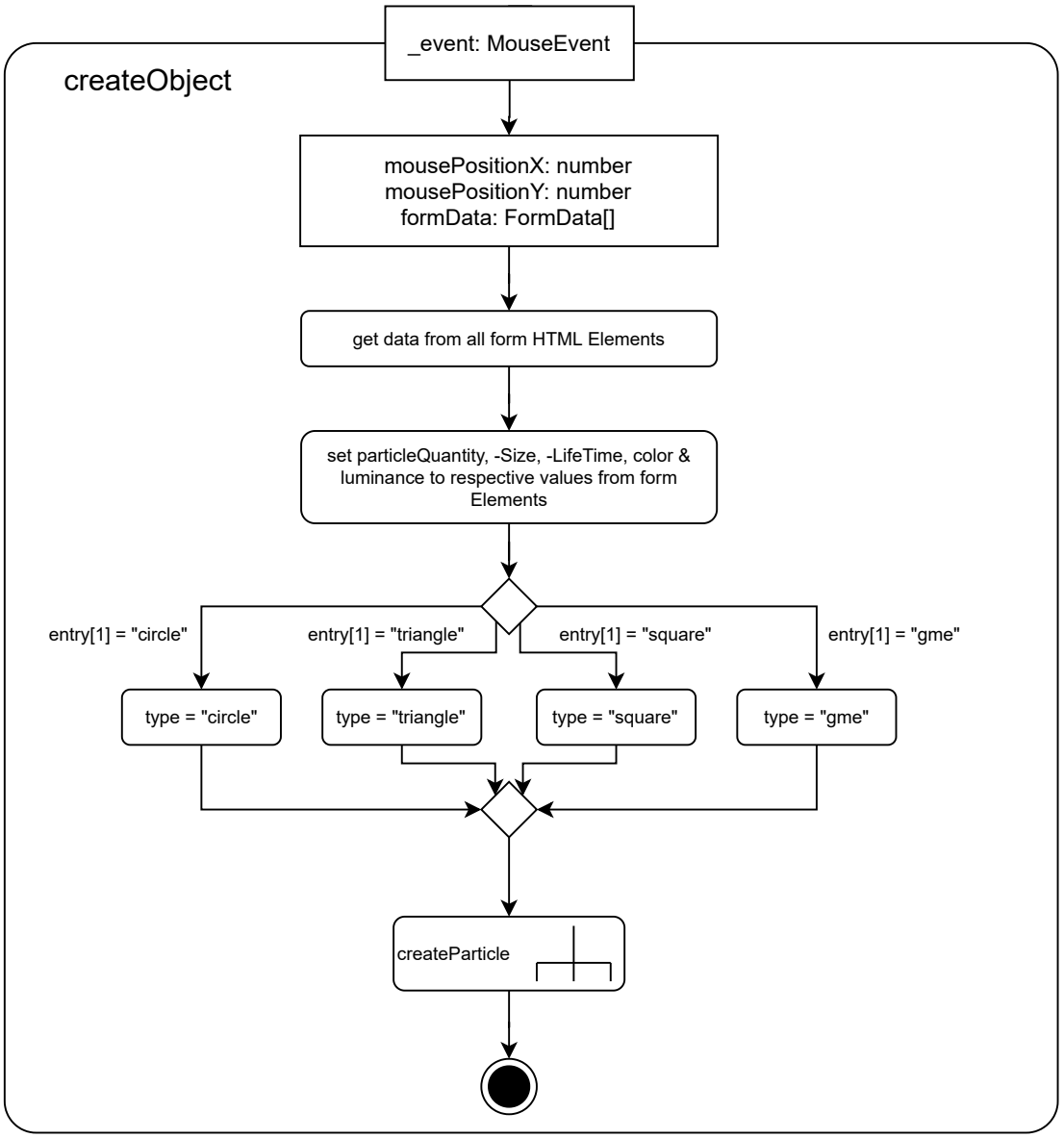
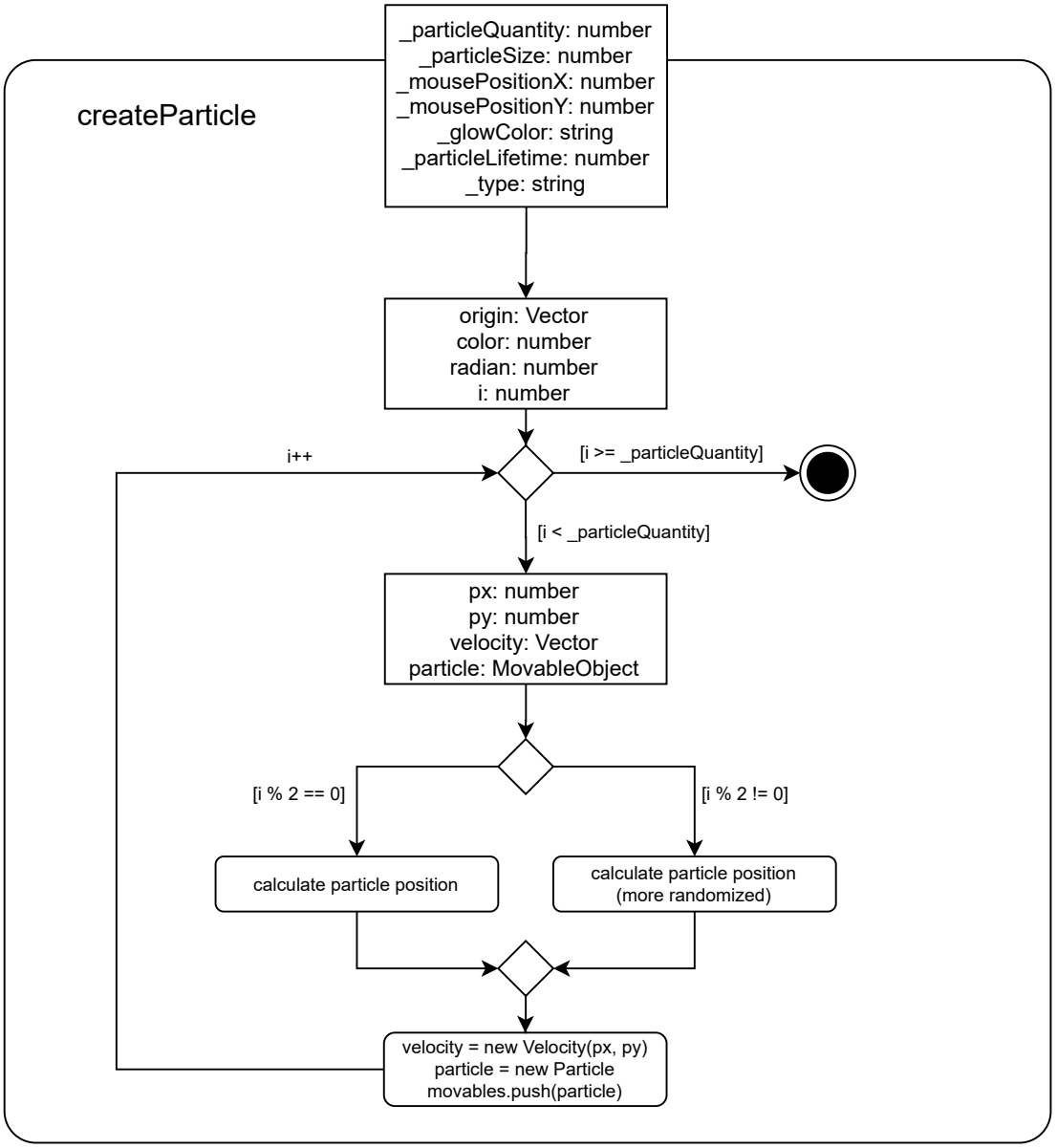
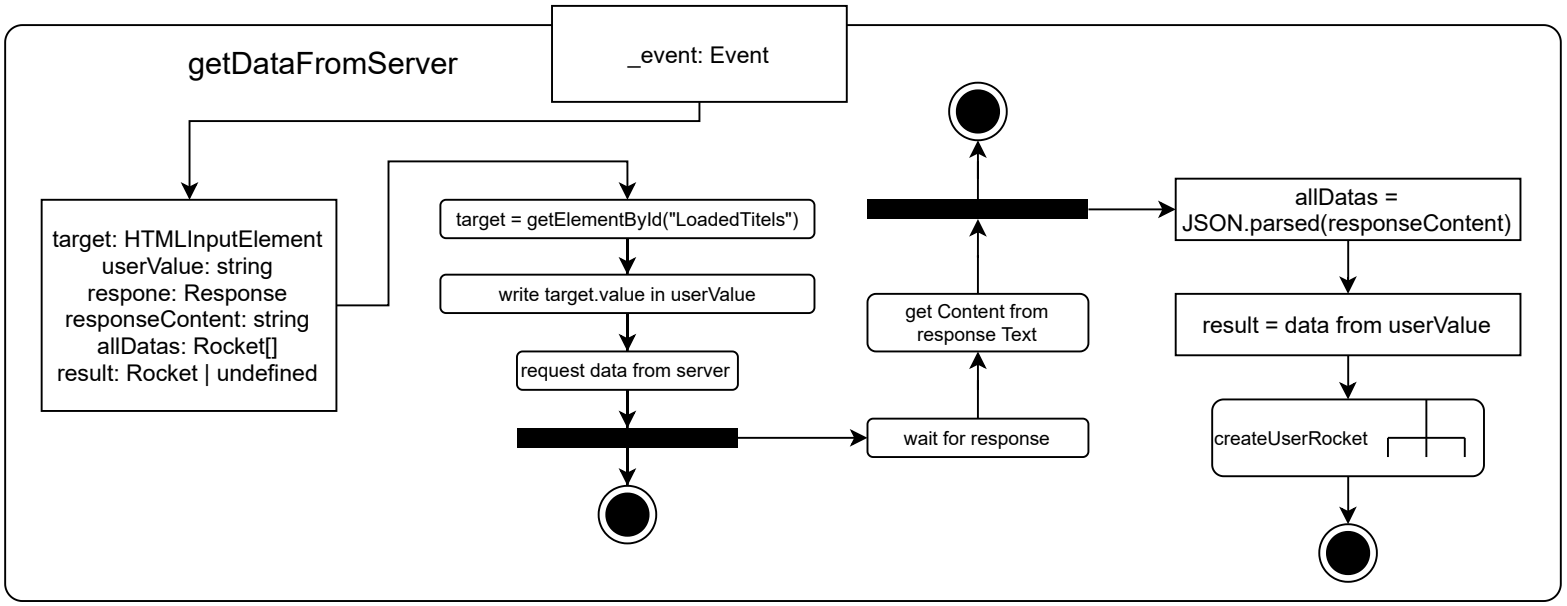
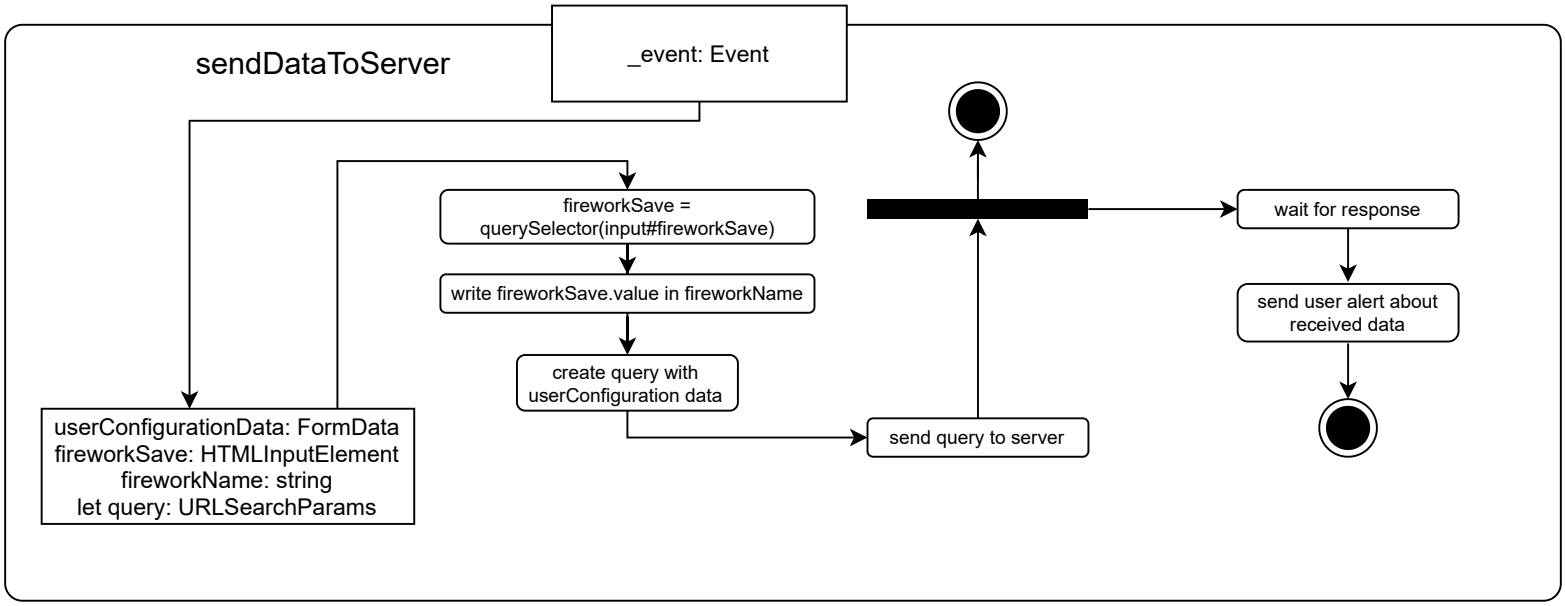
Activitydiagram - Main.ts

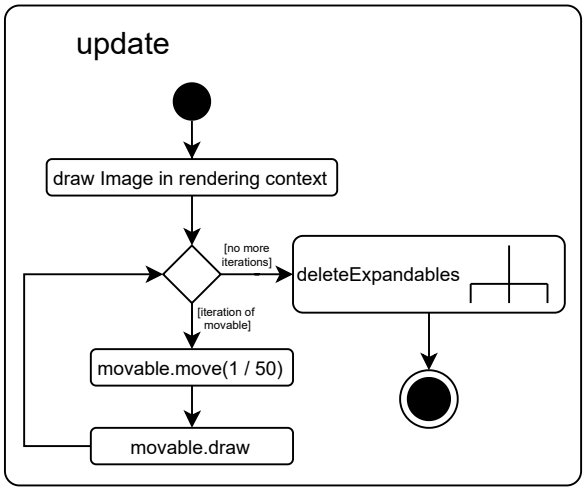


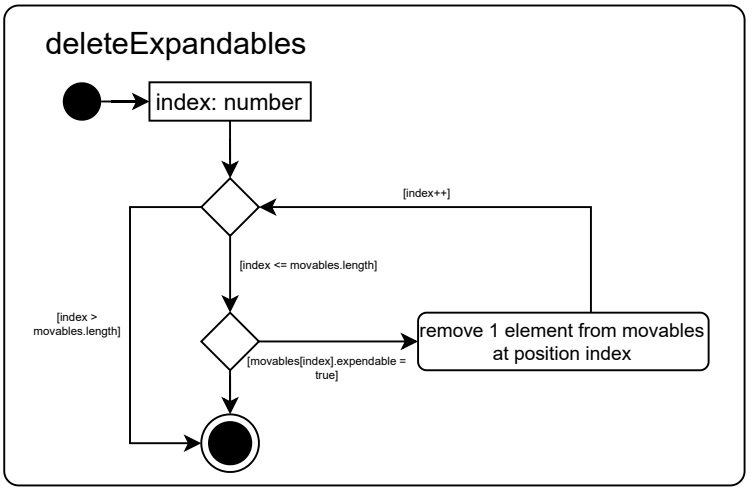



createParticle
_particleQuantity: number
_particleSize: number
_mousePositionX: number
_mousePositionY: number
_glowColor: string
_particleLifetime: number
_type: string
origin: Vector
color: number
radian: number
i: number
i++
[i >= _particleQuantity]
[i < _particleQuantity]
px: number
py: number
velocity: Vector
particle: MovableObject
[i % 2 == 0]
[i % 2 != 0]
calculate particle position
calculate particle position
(more randomized)
velocity = new Velocity(px, py)
particle = new Particle
movables.push(particle)

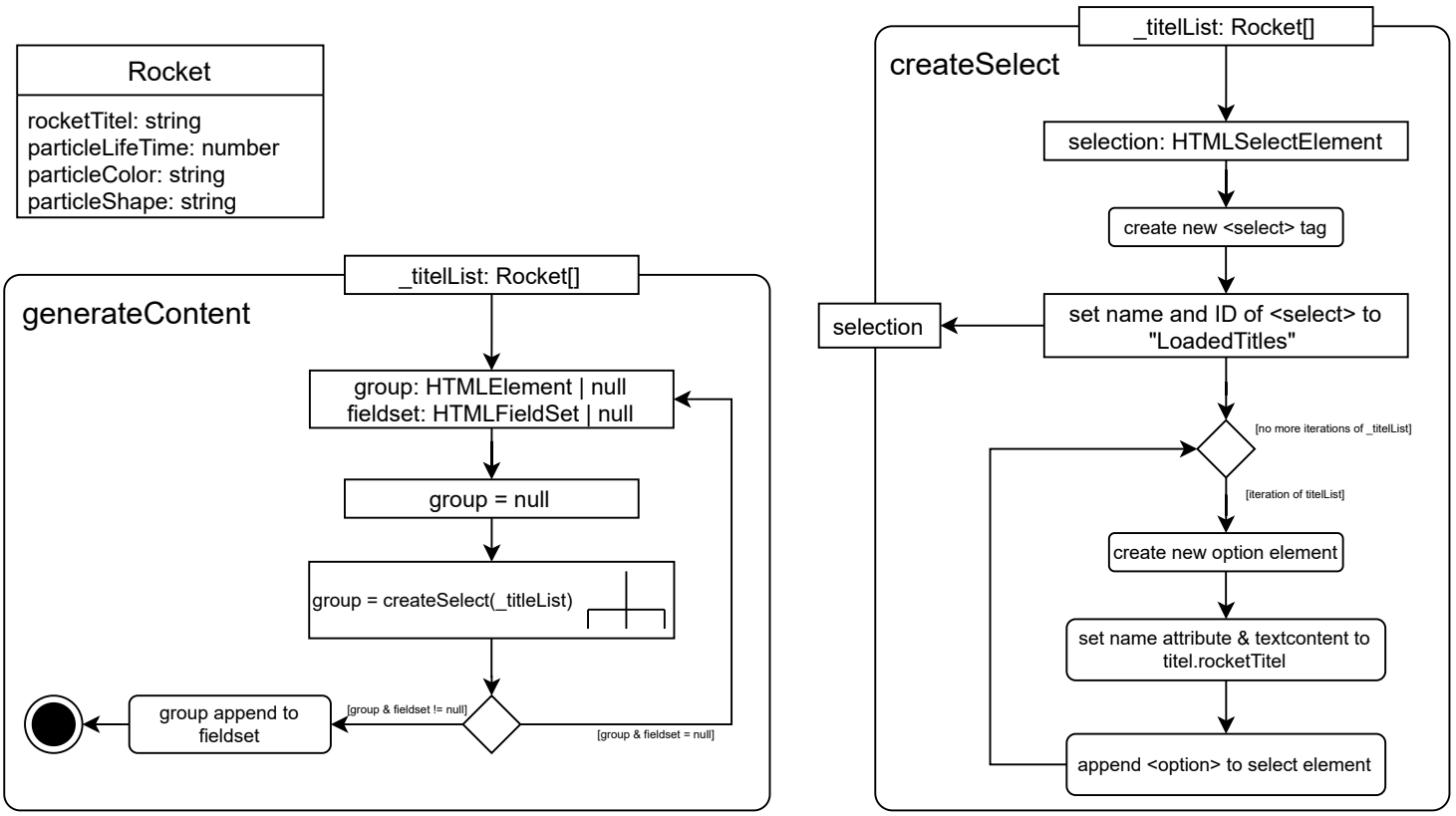

getDataFromServer
_event: Event
target: HTMLInputElement
userValue: string
respone: Response
responseContent: string
allDatas: Rocket[]
result: Rocket | undefined
target = getElementById("LoadedTitels")
write target.value in userValue
request data from server
wait for response
get Content from
response Text
allDatas =
JSON.parsed(responseContent)
result = data from userValue
createUserRocket


sendDataToServer
_event: Event
fireworkSave =
querySelector(input#fireworkSave)
write fireworkSave.value in fireworkName
create query with
userConfiguration data
send query to server
wait for response
send user alert about
received data
userConfigurationData: FormData
fireworkSave: HTMLInputElement
fireworkName: string
let query: URLSearchParams

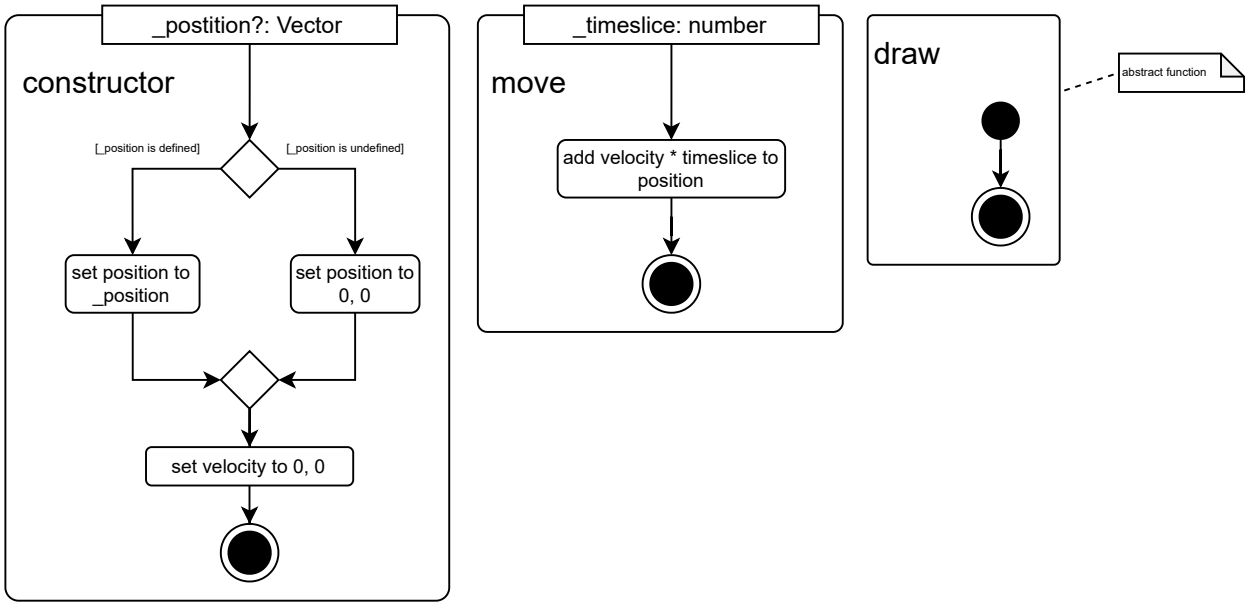

update
draw Image in rendering context
[no more
iterations]
deleteExpandables
[iteration of
movable]
movable.move(1 / 50)
movable.draw


deleteExpandables
index: number
[index++]
[index <= movables.length]
[index >
movables.length]
[movables[index].expendable =
true]
remove 1 element from movables
at position index

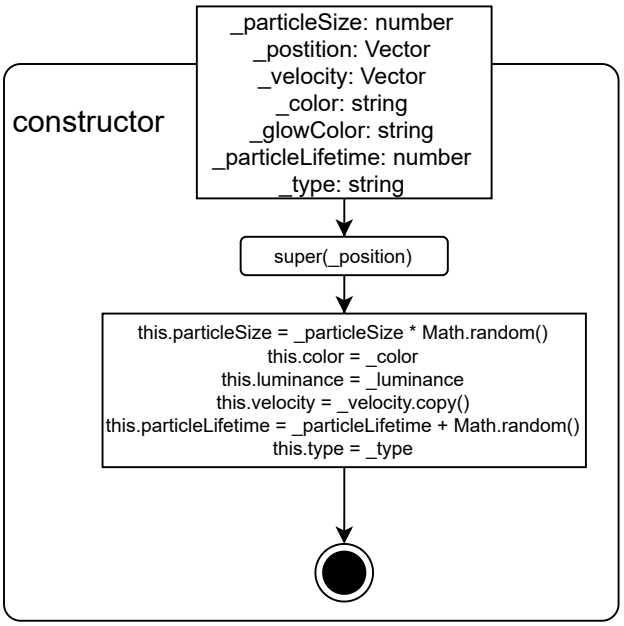
Activitydiagram - generateContent.ts

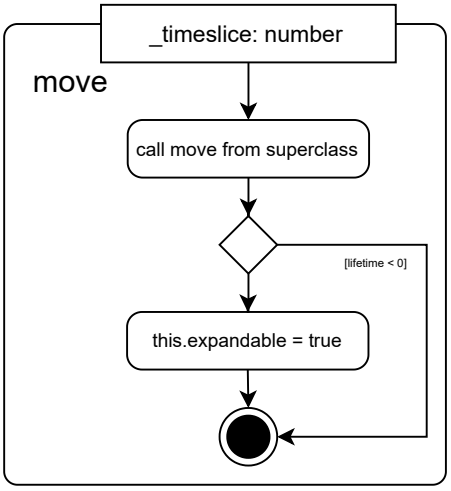


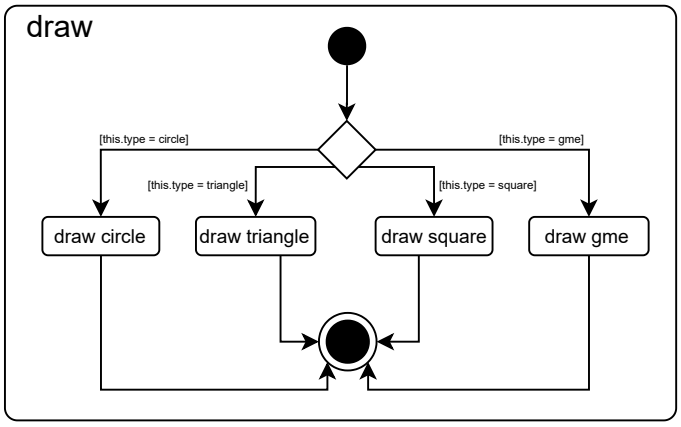
Activitydiagram - MoveableObject.ts



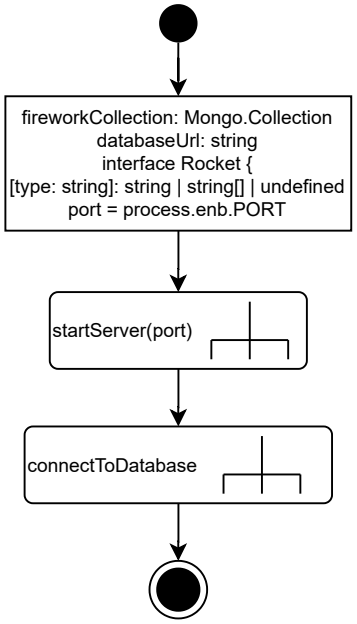
Activitydiagram - Particle.ts

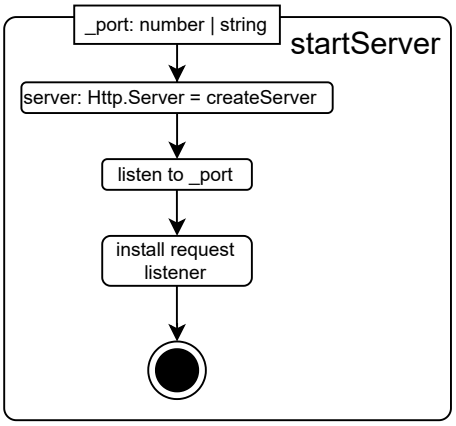


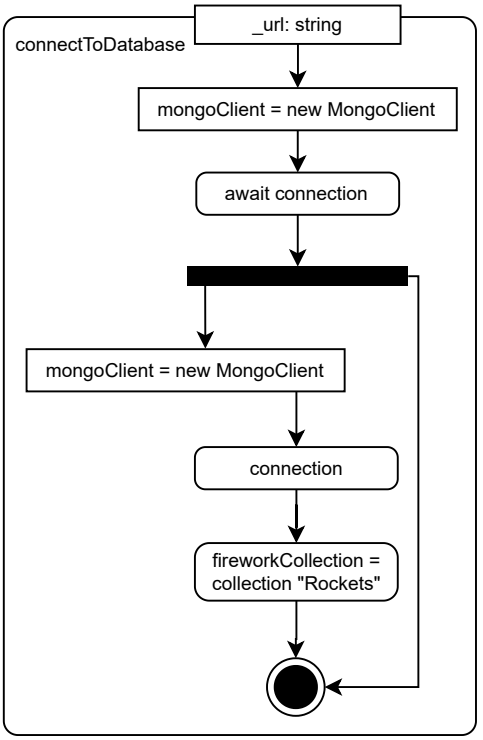


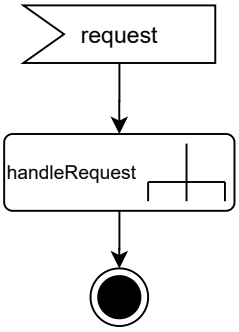


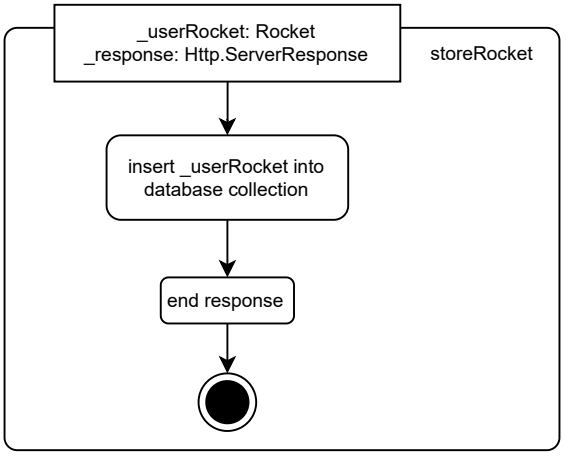
Activitydiagram - Server.ts

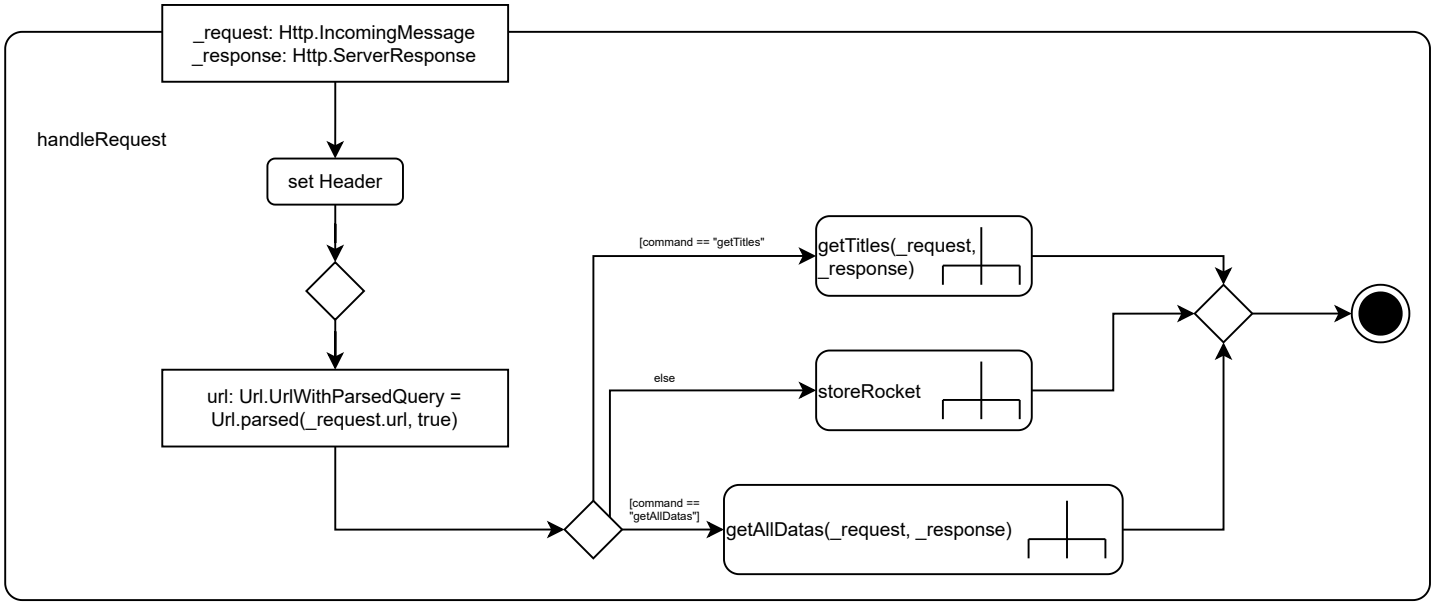


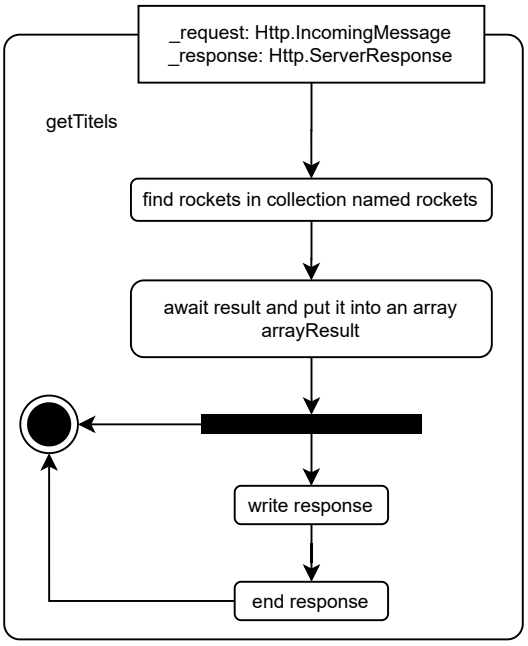


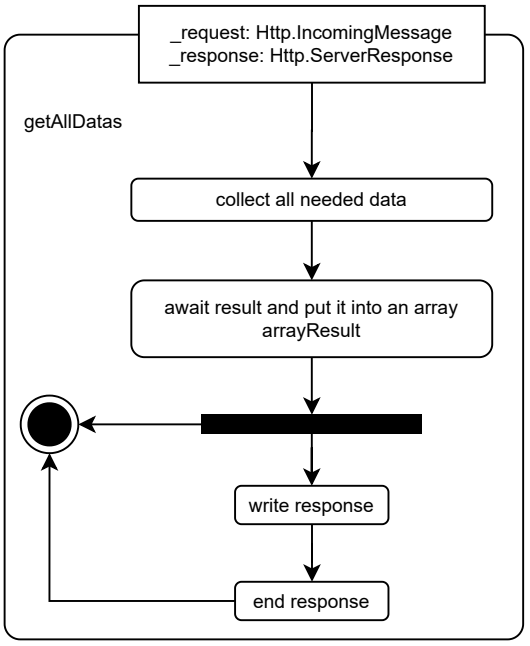









_request: Http.IncomingMessage
_response: Http.ServerResponse
getTitels
find rockets in collection named rockets
await result and put it into an array
arrayResult
write response
end response


_request: Http.IncomingMessage
_response: Http.ServerResponse
getAllDatas
collect all needed data
await result and put it into an array
arrayResult
write response
end response