```
┌─────────────────────────────┐
│ Model                       │
│ Character    character      │
│ Enemy        enemys         │
│ Enemys       enemys         │
├─────────────────────────────┤
│ processKeyTyped()           │
│ processTimeElapsed()        │
└─────────────────────────────┘
```

```
┌─────────────────────────────────┐
│ Character                       │
│ int x, y                        │
│ int gravity                     │
│ int hp                          │
│ int ex                          │
├─────────────────────────────────┤
│ move()                          │
│ damageConfirm(EnemysBullets ebs)│
│ exConfirm(Enemys enemys)        │
│ attack(Bullets bullet)          │
│ power up()                      │
└─────────────────────────────────┘
```

```
┌──────────────────────────────┐
│ Enemy                        │
│ int x, y                     │
│ int hp                       │
├──────────────────────────────┤
│ move()                       │
│ damageConfirm(Bullets bulle) │
│ attack(Enemysbullets ebs)    │
└──────────────────────────────┘
```

```
┌──────────────────┐
│ Enemys           │
│ add(Enemy e)     │
│ move()           │
│ damageConfirm()  │
│ attack()         │
└──────────────────┘
```

addはランダムに敵を出なし
ArrayList<Enemy>に追加す

```
┌──────────────────┐
│ ArrayList<Enemy> │
└──────────────────┘
```

ArrayList<Enemy>に
入っている全ての Enemyに
すべて、move、damage
Confirm、attackを
行う。

● moveはキータイプイベントによってキャラクターの座標を変える
● damageConfirmは敵の複数ある弾に当たっていないか
　確認し、当たっていたらその弾のダメージ分hpをへらす
● exConfirmは複数いる敵を確認し 倒れた敵が
　いたらその分 exをふやす。
● attack は bullets の中にある1つ1つの弾を
　update してから 描写する。

```
Model
Bullet bullet
Bullets bullets
EnemyBullet enemybullet
EnemyBullets enemybullets
processkeyTyped()
processTimeElapsed()
```

```
Bullet
int x, y
int vx, vy
int damage
update()
collisionEnemy(Enemys enemys)
collisionEnemyBullet(EnemysBullets enemybullets)
```

updateは弾の更新をする。

collisionEnemyは Enemysに入っている複数の敵と衝突しているか石を確かめる

これだけに対して

collisionEnemysBulletsは 複数の敵の弾と衝突しているかを確かめる

```
Bullets
add(Bullet bullet)
update()
collisionEnemy()
collisionEnemyBullet()
```
↓
```
ArrayList<Bullet>
```

弾を時間経過ごとに生成し、それを ArrayList<Bullet>に追加する

ArrayListに入っているBulletオブジェクトに対して
update, collisionEnemy, collisionEnemyBullet が実行される。

```
EnemyBullet
int x, y
int vx, vy
int damage
update()
collisionCharacter(Character
collisionBullets(Bullets bu
```

```
EnemyBullets
add(EnemyBullet enemybullet)
update()
collisionCharacter()
collisionBullets()
```
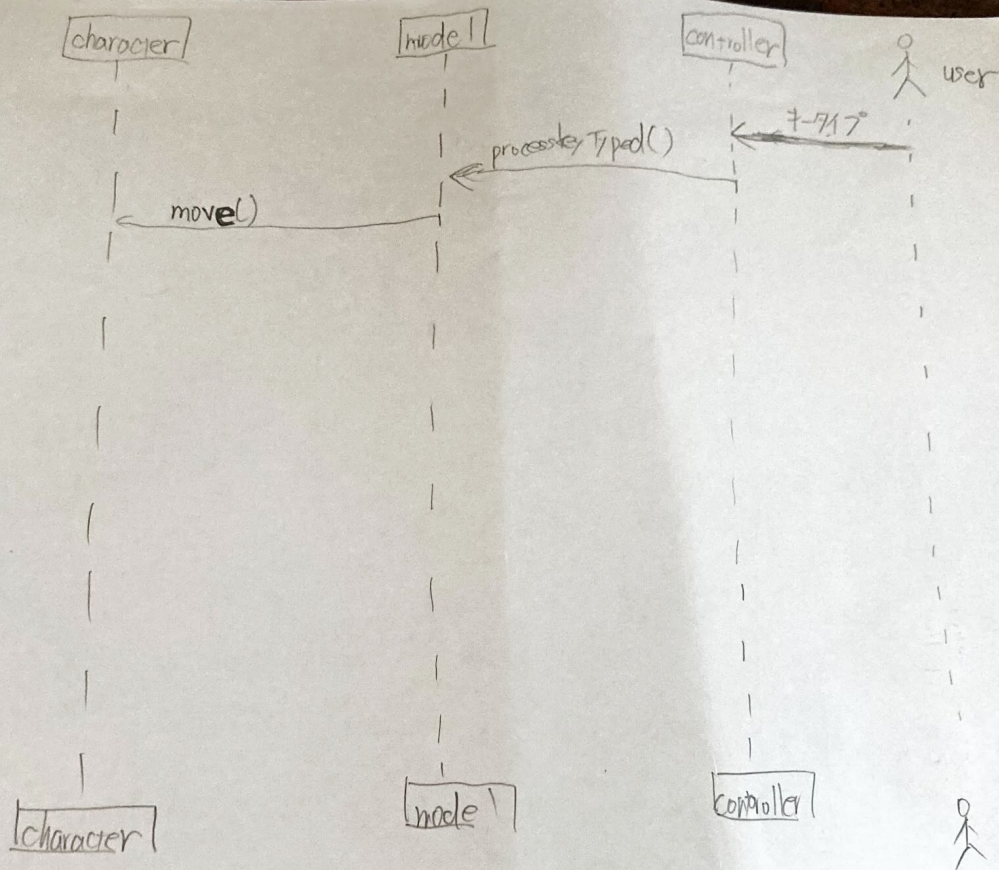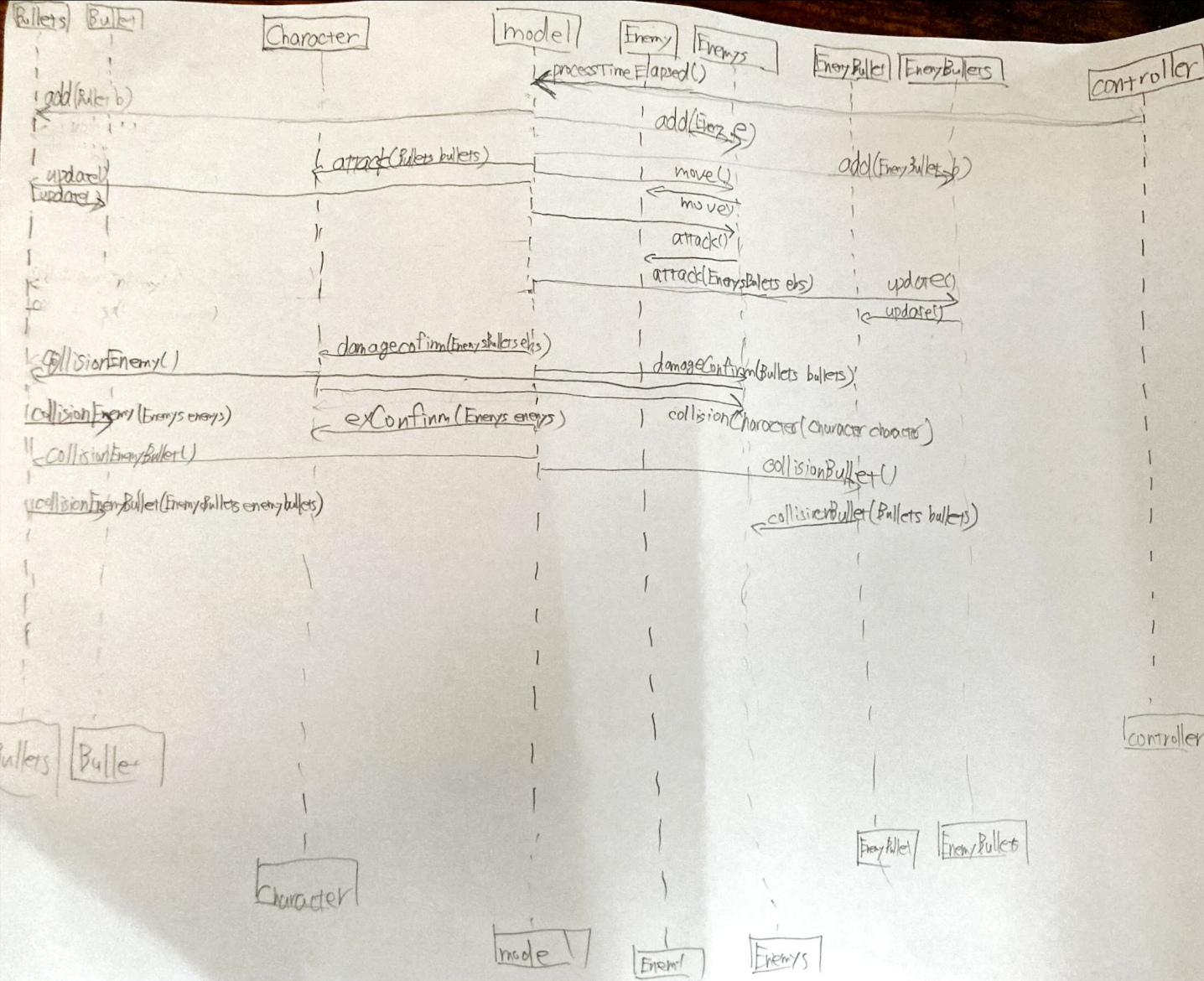↓
```
ArrayList<EnemyBullet>
```

キータイプ
イベント

character   model   controller   user

processKeyTyped()

キータイプ

move()

character   model   controller

Sequence diagram (hand-drawn). Lifelines and messages as labeled:

Lifelines (top): Bullets | Bullet, Character, model, Enemy | Enemys, EnemyBullet | EnemyBullets, controller

- processTimeElapsed()
- add(Bullet b)
- add(Enemy e)
- add(EnemyBullet b)
- attack(Bullets bullets)
- update()
- update()
- move()
- move()
- attack()
- attack(EnemysBullets ebs)
- update()
- update()
- collisionEnemy()
- damageConfirm(EnemyBullets ebs)
- damageConfirm(Bullets bullets)
- collisionEnemy(Enemys enemys)
- exConfirm(Enemys enemys)
- collisionCharacter(Character character)
- collisionEnemyBullet()
- collisionBullet()
- collisionEnemyBullet(EnemyBullets enemybullets)
- collisionBullet(Bullets bullets)

Lifelines (bottom): Bullets | Bullet, Character, model, Enemy, Enemys, EnemyBullet | EnemyBullets, controller

時間経過イベント