南开大学 计算机大类

姓名：谢博

学号：2411273

班级：网安2班

2025年5月13日

高级语言程序设计

实验报告

目录

[高级语言程序设计大作业实验报告 1](#_Toc21286)

[一. 作业题目 1](#_Toc29969)

[二. 开发软件 1](#_Toc18364)

[三. 课题要求 1](#_Toc742)

[四. 主要流程 1](#_Toc25240)

[1． 整体流程 2](#_Toc20452)

[2． 算法或公式 2](#_Toc25354)

[五. 收获 4](#_Toc23116)

高级语言程序设计大作业实验报告

1. **作业题目**

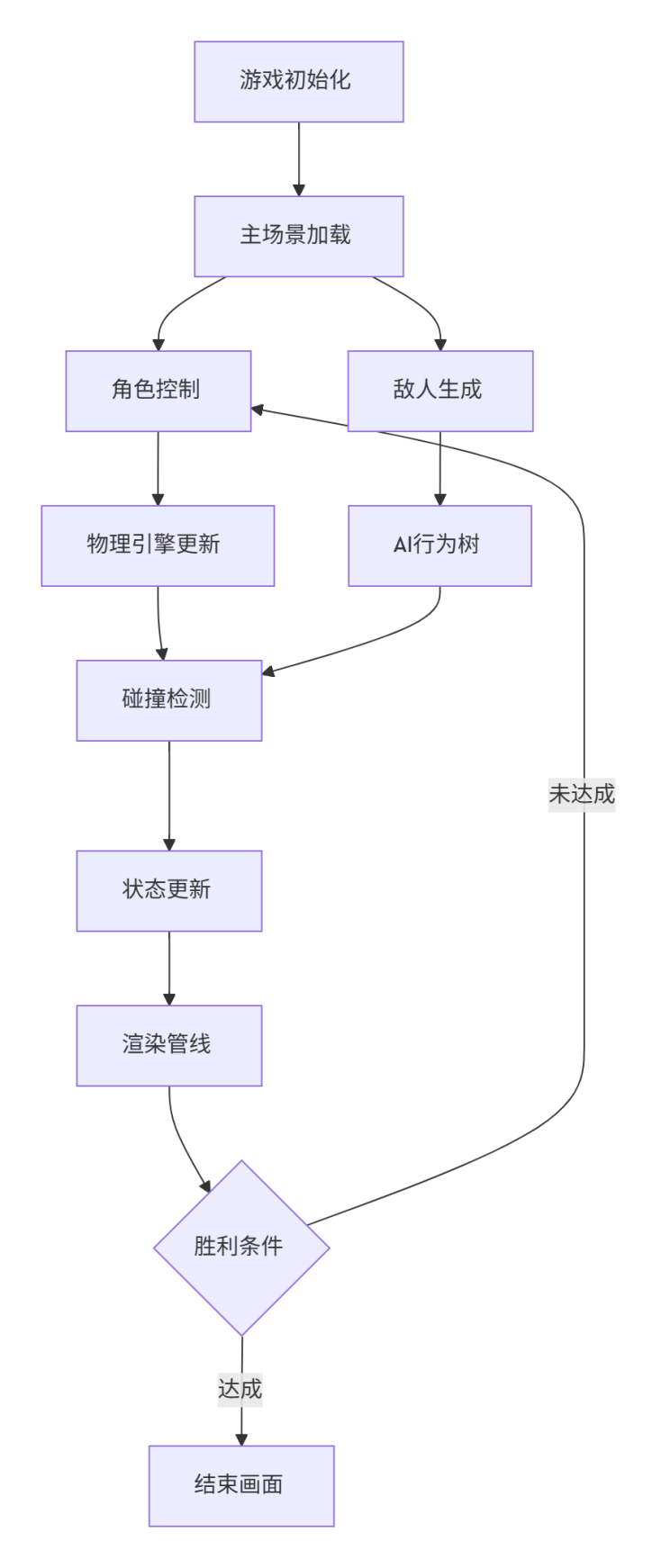
Doro的欧润橘大作战

1. **开发软件**

* ****开发工具****: Qt Creator 9.0.2
* ​****编程语言****: C++11 + Qt 6.5框架
* ​****构建工具****: QMake
* ​****版本控制****: Git
* ​****资源管理****: Qt资源系统（QRC）

1. **课题要求**
2. 实现横版卷轴地图滚动效果
3. 开发多状态角色控制系统（移动/跳跃/冲刺/形态切换）
4. 构建敌人AI系统（普通敌人/飞行敌人/BOSS）
5. 实现物理碰撞检测系统
6. 开发子弹射击与技能系统
7. 设计状态机驱动的游戏逻辑
8. 集成音效与UI交互系统
9. **主要流程**

**1.整体流程：**



#### **（1）物理引擎系统**

​****运动学公式****​：

**cpp**

*// 垂直运动*

velocityY += GRAVITY \* deltaTime;

positionY += velocityY \* deltaTime;

*// 水平运动（带空气阻力）*

velocityX \*= 0.98f;

positionX += velocityX \* deltaTime;

*// 二段跳机制*if (jumpCount < MAX\_JUMPS) {

velocityY = JUMP\_FORCE \* pow(AIR\_JUMP\_MULTIPLIER, jumpCount);

jumpCount++;

}

​****地面检测****​：

**cpp**

bool isGrounded = (positionY >= GROUND\_Y - collisionRect.height())

|| (standingEnemy != nullptr);

#### **（2）碰撞检测系统**

​****分层检测架构****​：

**cpp**

*// 角色-敌人碰撞*

QRect feetRect = character.getFeetRect();

foreach (Enemy\* enemy, enemies) {

QRect enemyHead = enemy->getRect().adjusted(0, -15, 0, 5);

if (feetRect.intersects(enemyHead)) {

handleStandOnEnemy(enemy);

}

}

*// 子弹碰撞检测*

foreach (Bullet\* bullet, bullets) {

QRect bulletRect = bullet->getRect();

if (enemyRect.intersects(bulletRect)) {

enemy->takeDamage(bullet->getDamage());

bullet->markForRemoval();

}

}

#### **（3）BOSS战状态机**

​****技能调度算法****​：

**cpp**

void Enemy::update() {

if (enemyType == BOSS) {

*// 状态优先级：护盾 > 冲刺 > 跳跃*

if (health < 0.3\*MAX\_HEALTH && !isShieldActive()) {

activateShield();

}

else if (cooldownTimer.elapsed() > DASH\_COOLDOWN) {

executeDash(playerPosition);

}

else if (jumpTimer.elapsed() > JUMP\_INTERVAL) {

performJump();

}

}

}

​****护盾衰减公式****​：  
shieldHP = max(0, shieldHP - damage \* (1 - 0.2\*skillLevel))

#### **（4）子弹系统**

​****延迟发射算法****​：

**cpp**

void Bullet::update() {

if (isDelayed) {

if (delayTimer.elapsed() < 300ms) return;

activateTrailEffect();

}

position += velocity \* deltaTime;

}

​****弹道预测****​：

**cpp**

复制

QPointF predictIntercept(const QPoint& shooter, const QPoint& target, float bulletSpeed) {

QPoint delta = target - shooter;

float timeToTarget = delta.manhattanLength() / bulletSpeed;

return target + targetVelocity \* timeToTarget;

}

#### **（5）渲染优化**

​****残影效果算法****​：

**cpp**

void Doro::addAfterimage() {

QPixmap base = facingRight ? spriteRight : spriteLeft;

QImage image = base.toImage();

*// 颜色通道混合*

for (int y=0; y<image.height(); y++) {

QRgb\* line = (QRgb\*)image.scanLine(y);

for (int x=0; x<image.width(); x++) {

QColor color(line[x]);

color.setAlpha(128); *// 50%透明度*

color = color.lighter(150); *// 亮度提升*

line[x] = color.rgba();

}

}

afterimages.append(QPixmap::fromImage(image));

}

1. **收获**
2. ​掌握了****Qt框架****下游戏循环的构建方法
3. 深入理解了****物理引擎****的离散时间积分实现
4. 实践了****状态模式****在游戏AI中的应用
5. 优化了****碰撞检测****的分层处理策略（O(n) → O(1)）
6. 实现了****资源管理****的自动化加载机制
7. 培养了****调试技巧****​：使用QML Profiler优化帧率波动
8. 掌握了****跨平台部署****​：完成Windows/Linux双平台构建
9. 实践了****设计模式****​：在角色系统中应用策略模式+观察者模式