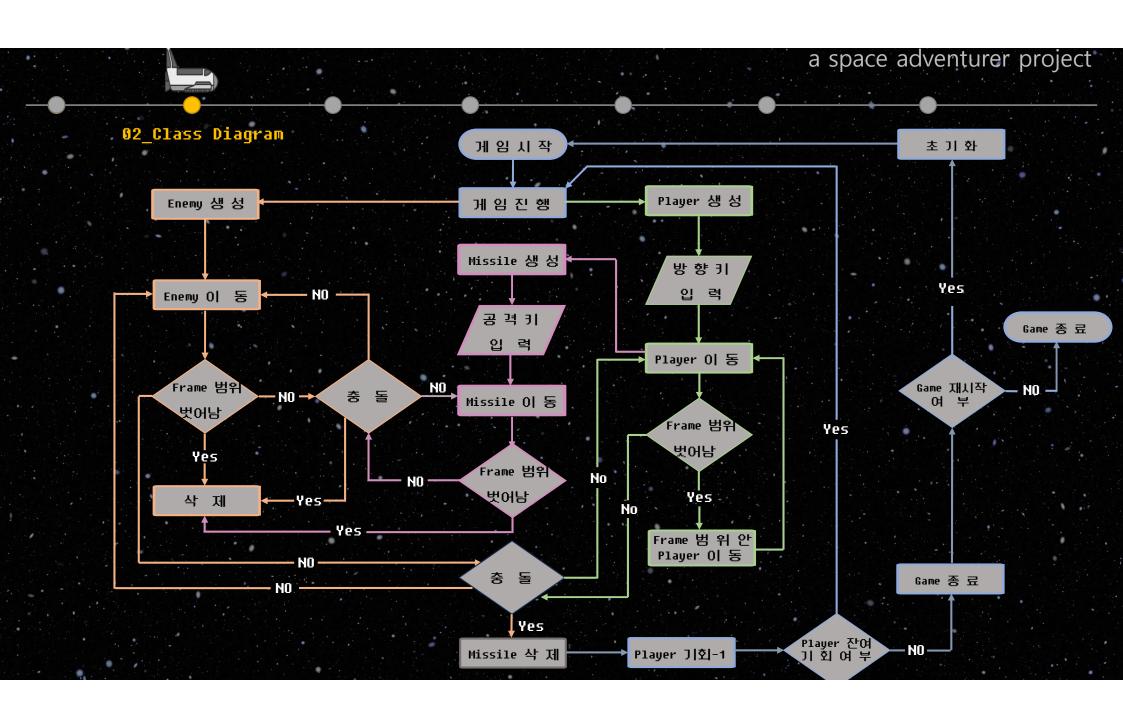
내일 할일 플로우 차트 애니메이션 순차적으로 넣어주기



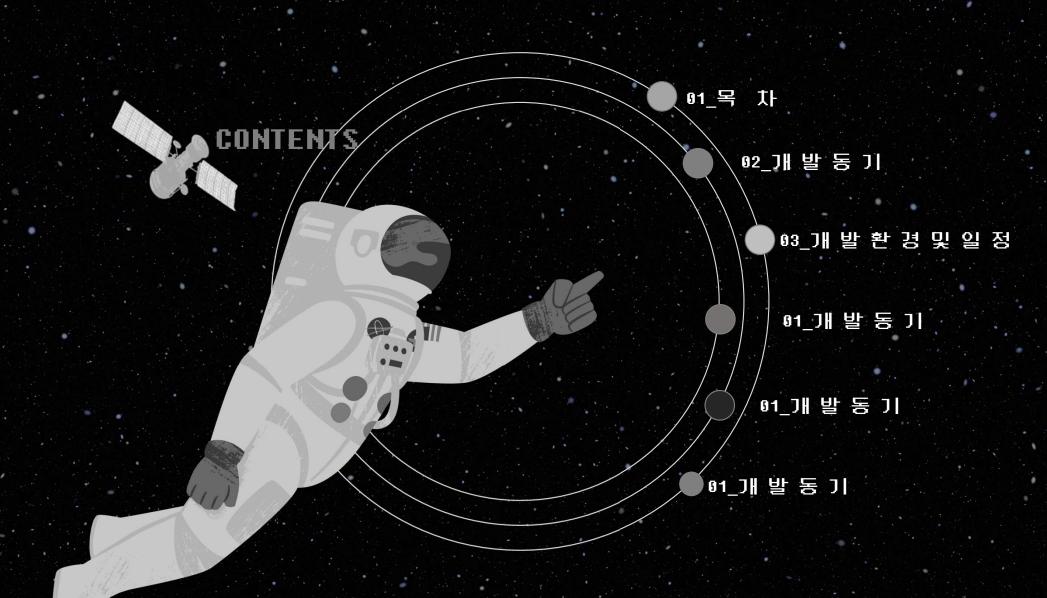
a space adventurer [홍민우]





JAVA Game Project

Graphics를 활용한 GUI





01_개 발 동 기 - 설 명

학습 및 숙달 필요 항목

- 객체지향 언어 4가지
- 추상화, 캡슐화, 상속, 다형성
- 접근제어자
- 컬렉션 프레임워크
- 상속과 인터페이스
- 메서드 오버라이드
- 추상클래스
- 디자인 패턴
- 전략(Strategy)패<u>턴</u>
- Graphics를 활용한 GUI



- 캐주얼 게임시장 확대
- 캐주얼 게임분야 매출 증가

02 Class Diagram PlayerCenter Player. R,L,U,D,RU,RD,LU,LD - static PlayerCenter playercenter Interface Player PlayerX int + Static PlayerCenter getPlayerCenter() PlayerY int return playercenter; + Image getPlayer() PlayerImage int getPlayerX() Type String Player Image getPlayer() Image Player 설정별 ● 전략(Strategy)패턴 + int getPlayerY() + setType(Stirnq type) void 8J∭ Class + int qetWidth() This.type = type; getPlayerX() int + int getHeight() + getPlayer() Player + getPlayerY() int Player temp = null; + getWidth() int Return temp; + getHeight() int MissileCenter Missile - static MissileCenter missilecenter R,L,U,D,RU,RD,LU,LD + static MissileCenter qetMissileCenter() return missilecenter: Interface Missile GameFrame - MissileOne Image PlayerType String - Bullet Point + void move() init() + MissileR(x,y) Point + void setPlayertype(String) + Image getImage() + int getbulletX() timerTh() move() void This.playertype = playertype; Missile getMissile(int X,int Y) MAIN Missile 설정별 enemuThR() int getbulletY() getImage() Imae 8J∦ Class GameFrame() enemyThL() + int qetWidth() qetBulletX() int Missile missile = null: + int qetHeight() qetBulletY() int enemyThU() return missile: + qetWidth() int enemyThD() + Missile getMissileXY(Missile,int,int) KeySettingTh() + getHeight() int return null; missileOneTh() EnemyCenter - static EnemyCenter enemycenyer Enemy + static EnemyCenter getEnemyCenter() Left,Right,Up,Down Interface Missile return enemucenter: EnemyOne Image Int x; Int x; Int y; String location; void setLocation(String location) This.location = location; Enemy getEnemyOne(int X, int Y) Enemy enemytemp = null; Return enemytemp; Missile getMissileXY(Missile,int,int) + void move() Body Point + Image getĺmage() Enemy(x,y) Enemy + int qetbulletX() getImage Image + int getbulletY()

return null;

Enemy 설정별 4J⊮ Class

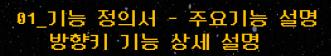
qetMove() void

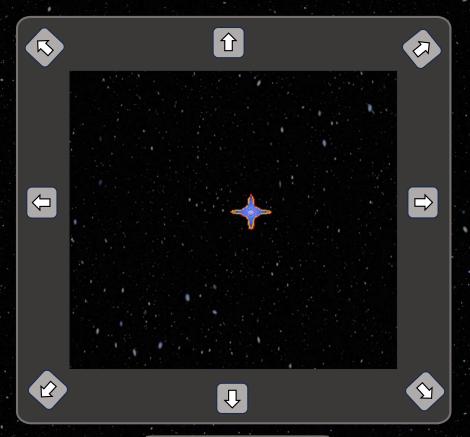
+ getenemyY() int + qetWidth() int + geteHeight() int

qetenemyX() int

+ int qetWidth()

+ int qetHeight()





- 상 세 설 명
- 1. 방향키 조작 움직임 구현
- 2. 인게임 내 Player 속도 조절 가능
- 3. 키(방향) 셋팅 가능













a space adventurer project

a space a venturer project



CONTENTS

목자

001

개 발 동 기

002



개 발 환 경 및 일 정 003



개 발 일 정

004



UML

005



요구사항

996



기 능 정 의 서

007



결 과 화 면

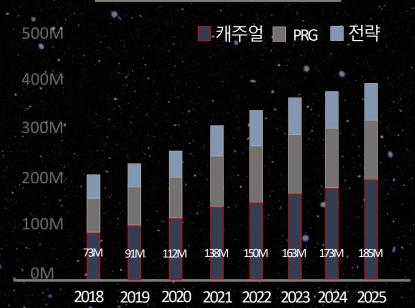
008

개발동기

학습 및 숙달 필요 항목

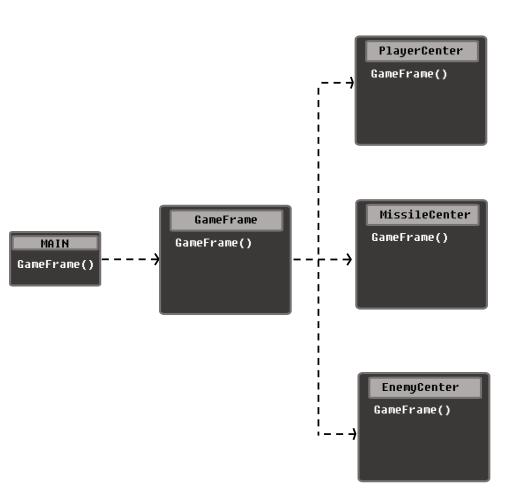
- 백체지향 언어 4가지
- 추상화, 캡슐화, 상속, 다형성
- 접근제어자
- 컬렉션 프레임워크
- 상속과 인터페이스
- 메서드 오버라이드
- 추상클래스
- 디자인 패턴
- 전략(Strategy)패턴
- ◆ Graphics를 활용한 GUI

다운로드 점유율 장르별



- 캐주얼 게임시장 확대
- 캐주얼 게임분야 매출 증가





PlayerCenter |

- static PlayerCenter playercenter
- static PlayerCenter getPlayerCenter()
- Type String
- getPlayer() Player

- return playercenter;
- setType(Stirng type) void This.type = type;
- Player temp = null; Return temp;

MissileCenter

- static MissileCenter missilecenter
- + static MissileCenter getMissileCenter() return missilecenter;
- Type String
- setType(Stirng type) void This.type = type;
- + qetPlayer() Player Player temp = null; Return temp;

Plauer R,L,U,D,RU,RD,LU,LD

- PlayerX int
- PlayerY int PlayerImage
- Player Image + getPlayer() Image
- + qetPlayerX() int
- + qetPlayerY() int
- + qetWidth() int
- + getHeight() int

Player 설정별 8J∰ Class

Missile R,L,U,D,RU,RD,LU,LD

- MissileOne Image
- Bullet Point MissileR(x,y) Point
- + move() void
- + getImage() Imae + getBulletX() int
- + getBulletY() int
- + qetWidth() int
- + qetHeiqht() int

Missile 설정별 8J∦ Class

EnemyCenter

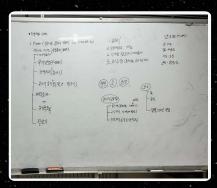
GameFrame()

Enemu Left,Right,Up,Down

- EnemyOne Image Bodu Point
- + Enemy(x,y) Enemy
- + getImage Image + getMove() void
- + qetenemyX() int
- + qetenemyY() int
- + qetWidth() int
- + geteHeight() int

Enemy 설정별 4개 Class





• 시나리오 작성



• 손코딩

마지막장에 두개널기 모험가,행성 페이지 넘어

갈때마다 이모션 조금씩

널어주)

◆ 디자인패턴 학습 (스트래티지 패턴사용)