**Pseudo Code**

**StartMenu**

void update()

{

if (NewGameButton is Pressed)

startGame()

else if (ExitGameButton is Pressed)

exitGame()

}

**PauseMenu**

void update()

{

if(continueButton is Pressed)

{

close PauseMenu

unfreeze GameManger.Update

}

}

**GameManager**

void Update()

{

update entities

checkInput()

collision.checkCollision()

if(pauseButtonPressed)

pause Game

isLevelFinished()

finish()

if(player.health <= 0)

gameover()

}

void startLevel()

{

create Level and HUD

create player, enemies, NPCs and objects

}

void gameOver()

{

blackscreen for a few seconds

open startMenu

}

void pause()

{

open pauseMenu

freeze GameManger.Update()

}

void finish()

{

if reached end of level

open startMenu

}

**Collision**

void checkCollision()

{

for(int i = 0; i < listOfObjects.length;i++)

{

for(int j = 0; j < listOfObjects.length;j++)

{

if (isColliding(listOfObject[i],listOfObject[j]))

objects colliding

}

}

}

**Player**

void takeDamage(int damage)

{

health -= damage;

update HUD

}

**Input**

Key registerKeyboardInput()

{

if (key pressed)

return key

}

Vector2 registerMouseInput()

{

If(mouse clicked)

return mouse position

}

**Patrol**

void execute(Enemy enemy)

{

moveAround()

if (playerInSight)

enemy.changeState(Chase.instance)

}

**Chase**

void execute(Enemy enemy)

{

followPlayer()

if(player nearby)

attack()

if(player not in sight)

enemy.changeState(Patrol.instance)

}