# GameUtil : WarriorAttack(Warrior warrior, AttackType attackType, Direction firstDirection, Direction secondDirection = Direction.INVALID\_DIRECTION)

Figure: Attached separately (GameUtil: WarriorAttack data flowgraph)

Def() and c-use() Sets of Nodes

|  |  |  |
| --- | --- | --- |
| Node i | def(i) | c-use(i) |
| 1 | {warrior, attackType, firstDirection, secondDirection} |  |
| 2 | {warriorPosition} | {warrior.Position} |
| 3 | {attackPosition, opponent} | {warriorPosition, firstDirection, secondDirection, attackPosition} |
| 4 | {} | {} |
| 5 | {warrior.MeleePower, warrior.MagicPower} | {warrior, PowerType.MELEE\_POWER, PowerType.MAGIC\_POWER} |
| 6 | {} | {} |
| 7 | {warrior.MeleePower, warrior.MagicPower} | {warrior, PowerType.MELEE\_POWER, PowerType.MAGIC\_POWER} |
| 8 | {} | {} |
| 9 | {} | {} |
| 10 | {} | {} |
| 11 | {} | {} |
| 12 | {meleeDamage, opponent.HitPoints} | {opponent.MeleePower, opponent.DefensePercentage, opponent.HitPoints, meleeDamage |
| 13 | {} | {} |
| 14 | {magicDamage, opponent.HitPoints} | {opponent.MagicPower, opponent.DefensePercentage, opponent.HitPoints, magicDamage} |
| 15 | {opponent.DefensePercentage} | { opponent, PowerType.DEFENCE\_PERCENTAGE} |
| 16 | {} | {warrior, opponent} |
| 17 | {} | {} |

Predicates and p-use() Set of Edges

|  |  |  |
| --- | --- | --- |
| Edges(i, j) | predicate(i,j) | p-use(i,j) |
| (1,2) | True | {} |
| (2,3) | board.IsPositionInsideBoundary(warriorPosition, firstDirection, secondDirection) | {warriorPosition, firstDirection, secondDirection} |
| (2,4) | ~(board.IsPositionInsideBoundary(warriorPosition, firstDirection, secondDirection)) | {warriorPosition, firstDirection, secondDirection} |
| (3,5) | warrior is MeleeWarrior | {warrior, MeleeWarrior} |
| (3,6) | ~(warrior is MeleeWarrior) | {warrior, MeleeWarrior} |
| (6,7) | warrior is MagicWarrior | {warrior, MagicWarrior} |
| (6,8) | ~(warrior is MagicWarrior) | {warrior, MagicWarrior} |
| (5,8) | True | {} |
| (8,9) | opponent != null | {opponent} |
| (8,17) | ~(opponent != null) | {opponent} |
| (9,10) | attackType == AttackType.MeleeAttack | {attackType, AttackType.MeleeAttack} |
| (9,13) | ~(attackType == AttackType.MeleeAttack) | {attackType, AttackType.MeleeAttack} |
| (10,11) | secondDirection != Direction.INVALID\_DIRECTION | {secondDirection, Direction.INVALID\_DIRECTION} |
| (10,12) | ~(secondDirection != Direction.INVALID\_DIRECTION) | {secondDirection, Direction.INVALID\_DIRECTION} |
| (13,14) | attackType == AttackType.MagicAttack | {attackType, AttackType.MagicAttack} |
| (13,15) | ~(attackType == AttackType.MagicAttack) | {attackType, AttackType.MagicAttack} |
| (14,15) | True | {} |
| (12,15) | True | {} |
| (15,16) | opponent.HitPoints <= 0 | {opponent.HitPoints} |
| (15,17) | True | {} |
| (16,17) | True | {} |

The change in test case is seen control flow test. No change in data flow testing.

# Weapon : constructor

Def() and c-use() Sets of Nodes

|  |  |  |
| --- | --- | --- |
| **Node i** | **def(i)** | **c-use(i)** |
| **1** | {weaponType} | {} |
| **2** | {this.weaponType} | {weaponType} |
| **3** | {this.Power} | {} |
| **4** | {this.Power} | {} |
| **5** | {} | {} |

Predicates and p-use() Set of Edges

|  |  |  |
| --- | --- | --- |
| **Edges (i, j)** | **predicate(i, j)** | **p-use(i, j)** |
| **(1, 2)** | True | {} |
| **(2, 3)** | ((this.weaponType == WeaponType.Staff)) | {weaponType} |
| **(2, 4)** | ~((this.weaponType == WeaponType.Staff)) | {} |
| **(3, 5)** | True | {} |
| **(4, 5)** | True | {} |

No change in the path based on this test.

# Warrior: addWeapon(Weapon weapon)

Def() and c-use() Sets of Nodes

|  |  |  |
| --- | --- | --- |
| **Node i** | **def(i)** | **c-use(i)** |
| **1** | {weapon} | {} |
| **2** | {this.weapon} | {weapon} |
| **3** | {InvalidOperationException} | {} |
| **4** | {} | {} |
| **5** | {this.MeleePower} | {this.MeleePower, this.weapon.Power} |
| **6** | {this.MagicPower} | {this.MagicPower, this.weapon.Power} |
| **7** | {} | {} |

Predicates and p-use() Set of Edges

|  |  |  |
| --- | --- | --- |
| **Edges (i, j)** | **predicate(i, j)** | **p-use(i, j)** |
| **(1, 2)** | ~(this.weapon != null) | {this.weapon} |
| **(1, 3)** | (this.weapon != null) | {this.weapon} |
| **(2, 4)** | ~(weapon.WeaponType == WeaponType.Sword) | {weapon.WeaponType} |
| **(2, 5)** | (weapon.WeaponType == WeaponType.Sword) | {weapon.WeaponType} |
| **(4, 6)** | (weapon.WeaponType == WeaponType.Staff) | {weaon.WeaponType} |
| **(4,7)** | ~(weapon.WeaponType == WeaponType.Staff) | {weapon.WeaponType} |
| **(5,7)** | True | {} |
| **(6,7)** | True | {} |

No change in path based on this test.

# Board:

GetPositionInADirection(Position position, Direction firstDirection, Direction secondDirection = Direction.INVALID\_DIRECTION)

Def() and c-use() Sets of Nodes

|  |  |  |
| --- | --- | --- |
| **Node i** | **def(i)** | **c-use(i)** |
| **1** | {position, firstDirection, secondDirection} | {} |
| **2** | {deltaPosition, newPosition, X, Y} | {Position, position.X, deltaPositon.X, position.Y, deltaPositon.Y} |
| **3** | {newDelta, newPosition.X, newPosition.Y} | {secondDirection, newPosition.X, newDelta.X, newPositon.Y, newDelta.Y} |
| **4** | {} | {} |

Predicates and p-use() Set of Edges

|  |  |  |
| --- | --- | --- |
| **Edges (i, j)** | **predicate(i, j)** | **p-use(i, j)** |
| **(1, 2)** | True | {} |
| **(2, 3)** | (secondDirection != Direction.INVALID\_DIRECTION) | {secondDirection, Direction.INVALID\_DIRECTION} |
| **(2, 4)** | ~(secondDirection != Direction.INVALID\_DIRECTION) | {secondDirection, Direction.INVALID\_DIRECTION} |
| **(3, 4)** | True | {} |

No change in path based on this test.