Settle And Destroy (SAD)

Group 13
Jonas Wikberg
Christofer Hjalmarsson
Daniel Westerberg
Saul Amram
André Sikborn Erixon

1.

2.

3.

4.

5.

5.1.

5.2.

5.3.

5.4.

5.5. Detailed Design

5.5.1. Class Village

Fields:

Attribute: team
Type: Team
Usage: Assigns a village name

Attribute: homeArmy
Type: Army
Usage: The stationary army where new built troops are gathered

Methods:

increaseMoney()

Method Name: increaseMoney
Parameters: -
Return Value: -
Description: The method increases the player money production
Data structures: -
Pre-conditions: -
Validity Checks, Errors, and other Anomalous Situations: -
Post-conditions: The player money amount is increased
Called by: MainWindow.createThread()
Calls: Team.addMoney()

5.5.2. Interface Building

Fields:

Methods:

getVillage()

Method Name: getVillage()
Parameters: -
Return Value: Village
Description: The method returns the village a building belongs to
Data structures: -
Pre-conditions: -
Validity Checks, Errors, and other Anomalous Situations: -
Post-conditions: The village is returned
Called by: TotalCostPanel.buildItemNumberChanged()
Calls: Team.addMoney()

String getName()

Method Name: getName()
Parameters: -
Return Value: String name
Description: The method returns the building name
Data structures: -
Pre-conditions: -
Validity Checks, Errors, and other Anomalous Situations: -
Post-conditions: The name of the building is returned
Called by: UpgradeBuildingPanel.buttonUpgradeActionPerformed(), BuildingPanel.update(), BuildingButton()
Calls:

int getLevel()

Method Name: getLevel()
Parameters: -
Return Value: int
Description: The method returns the building level
Data structures: -
Pre-conditions: -
Validity Checks, Errors, and other Anomalous Situations: -
Post-conditions: The level of the building is returned
Called by: UpgradeBuildingPanel.updateButton(),UpgradeBuildingPanel.buttonUpgradeActionPerformed(),BuildingPanel.update()
Calls: -

isUpgradable()

Method Name: isUpgradable()
Parameters: -
Return Value: boolean
Description: The method checks if the building is upgradable

Data structures: -
Pre-conditions: -
Validity Checks, Errors, and other Anomalous Situations: -
Post-conditions: A true or false is returned
Called by: BuildingPanel.update()
Calls: -

getUpgradeCost()

Method Name: getUpgradeCost()
Parameters: -
Return Value: int
Description: The method checks the upgrade cost of the building
Data structures: -
Pre-conditions: -
Validity Checks, Errors, and other Anomalous Situations: -
Post-conditions: The cost of upgrade is returned
Called by: UpgradeBuildingPanel.updateButton()
Calls: -

getUpgradeTime()

Method Name: getUpgradeTime()
Parameters: -
Return Value: int
Description: The method checks the upgrade time of the building
Data structures: -
Pre-conditions: -
Validity Checks, Errors, and other Anomalous Situations: -
Post-conditions: The time of upgrade is returned
Called by: UpgradeBuildingPanel.buttonUpgradeActionPerformed()
Calls: -

getBuildableItems()

Method Name: getBuildableItems()
Parameters: -
Return Value: BuildableItem[]
Description: The returns all buildable buildings for a specified village
Data structures: -
Pre-conditions: -
Validity Checks, Errors, and other Anomalous Situations: -
Post-conditions: The buildable buildings are returned
Called by: BuildingPanel.update()
Calls: -

5.5.3. Class Team

Fields

Attribute: name
Type: String

Usage: Every team has a unique name to separate them from each other.

Attribute: money
Type: int
Usage: This is used to keep track of a teams money they can spend.

Attribute: color
Type: Color
Usage: Every team has a unique team color to separate them from other teams.

Attribute moneyListeners
Type: List<MoneyListener> - List of moneyListeners

Methods:

Method: addMoney(int money)
Parameters: money – how much you should add to the team money.
Return Value: -
Description: This method is used to add money to the team. The amount of money added is told by the parameter.
Data structures: -
Pre-condition: A team has gain money in some way and need to add it to there team money.
Validity Checks, Errors, and other Anomalous Situations: -
Post-condition: The teams money has change.
Called by: Village.increadeMoney()
Calls: -

5.5.4. Class Army

Field:

Attribute: unitA
Type: int
Usage: Is used to know how many troops of the type unitA this army consist of.

Attribute: unitB
Type: int
Usage: Is used to know how many troops of the type unitB this army consist of.

Attribute: unitC
Type: int
Usage: Is used to know how many troops of the type unitB this army consist of.

Attribute: speed
Type: float
Usage: Is used to know the speed of the army, the speed is equal to the slowest troop in the army.

Method:

Method: getSpeed()
Parameters: -.

Return Value: int speedValue
Description: This method is used to get the speed of the army
Data structures: -
Pre-condition: The army consists of at least 1 unit of any kind.
Validity Checks, Errors, and other Anomalous Situations: -
Post-condition: The team has the speed of the slowest unit in the army.
Called by: Pathfinder.findPath()
Calls: -

Method: setSpeed(int unitA, int unitB, int unitC)
Parameters: unitA – how many troops of the unitA
unitB – how many troops of the unitB
unitC – how many troops of the unitC
Return Value: -
Description: This method is used to set the speed of the army.
Pre-condition: The speed of the army is set to the slowest unit in this army.
Called by: Building.trainTroops(), Map.formMergeArmy
Calls: Army.addArmy(int unitA, int unitB, int UnitC)

5.5.5. Interface BuildableItem

Methods:

getRequiredLevel()

Method Name: getRequiredLevel
Parameters: -
Return Value: int - The minimum level of a building required to build this item
Description: Returns the required minimum level
Data structures: -
Pre-conditions: -
Validity Checks, Errors, and other Anomalous Situations: -
Post-conditions: The required level is returned
Called by: BuildingPanel.update
Calls: -

getName()

Method Name: getName
Parameters: -
Return Value: String - The name of the item
Description: Returns the name of the item
Data structures: -
Pre-conditions: -
Validity Checks, Errors, and other Anomalous Situations: -
Post-conditions: The name is returned
Called by: BuildingPanel.update, BuildItemPanel constructor
Calls: -

getCost()

Method Name: getCost
Parameters: -
Return Value: int - The cost to build this item
Description: Returns the cost
Data structures: -
Pre-conditions: -
Validity Checks, Errors, and other Anomalous Situations: -
Post-conditions: The cost is returned
Called by: TotalCostPanel.getTotalCost, BuildItemPanel.updateCost
Calls: -

getBuildTime ()

Method Name: getBuildTime
Parameters: -
Return Value: int - The time it takes to build on of this item
Description: Returns the build time
Data structures: -
Pre-conditions: -
Validity Checks, Errors, and other Anomalous Situations: -
Post-conditions: The cost build time is returned
Called by: -
Calls: -

5.5.6. Class Map

Fields

Attribute: grid
Type: Cell[][]
Usage: All map cells are stored in this cell-matrix.

Attribute: randomizer
Type: Random
Usage: Machine for producing random seeds for the map creation process. This is needed to make each game map unique.

Methods:

Method: generateRivers()
Parameters: -
Return Value: -
Description: Generates and randomizes amount of rivers that should exist on the map. Also randomizes how long each river should be.
Data structures: -
Pre-conditions: A game is launched and a map is needed.
Validity Checks, Errors, and other Anomalous Situations: -
Post-conditions: A map with a cell matrix full of different cells is created.
Called by: Game.generateMap()
Calls: createRiver(int n)

Method: createRiver(int riverSize)
Parameters: riverSize – specifies how many cells this river should be
Return Value: -
Description: Randomizes rivers positioning and generates the related cells in the cell matrix.
Data structures: -
Pre-conditions: Rivers are being created.
Validity Checks, Errors, and other Anomalous Situations: -
Post-conditions: A river is created.
Called by: generateRivers()
Calls: -

Method: generateRocks()
Parameters: -
Return Value: -
Description: Generates and randomizes amount of rocks/mountains that should exist on the map. Also randomizes how big each rock should be.
Data structures: -
Pre-conditions: A game is launched and a map is needed. Rivers are created.
Validity Checks, Errors, and other Anomalous Situations: -
Post-conditions: A map with a cell matrix full of different cells is created.
Called by: Game.generateMap()
Calls: createRock(int n)

Method: createRock(int rockSize)
Parameters: rockSize – specifies how many cells this rock should be
Return Value: -
Description: Randomizes rock positioning and generates the related cells in the cell matrix.
Data structures: -
Pre-conditions: Rocks are being created.
Validity Checks, Errors, and other Anomalous Situations: -
Post-conditions: A rock is created.
Called by: generateRocks()
Calls: -

Method: generatePlains()
Parameters: -
Return Value: -
Description: Generates and creates plains cells in the empty cells of the cell-matrix.
Pre-conditions: A game is launched and a map is needed. Rivers and rocks are created.
Validity Checks, Errors, and other Anomalous Situations: -
Post-conditions: A map with a cell matrix full of different cells is created.
Called by: Game.generateMap()
Calls: -

5.5.7. Class Combat Calculator

Methods:

Method: calculateCombat(Army armyB, Army armyC)
Parameters: armyB – An army
armyC – An army of another player
Return Value: Army winningArmy
Description: Calculates who is the combat's winning army, dependant on army factors, troops relations and some random factors.
Pre-conditions: Two different player's armies meet at the same map cell.
Validity Checks, Errors, and other Anomalous Situations: -
Post-conditions: One army has been eliminated,
Called by: Cell.calculateCombat(Army armyB, Army armyC)
Calls: -

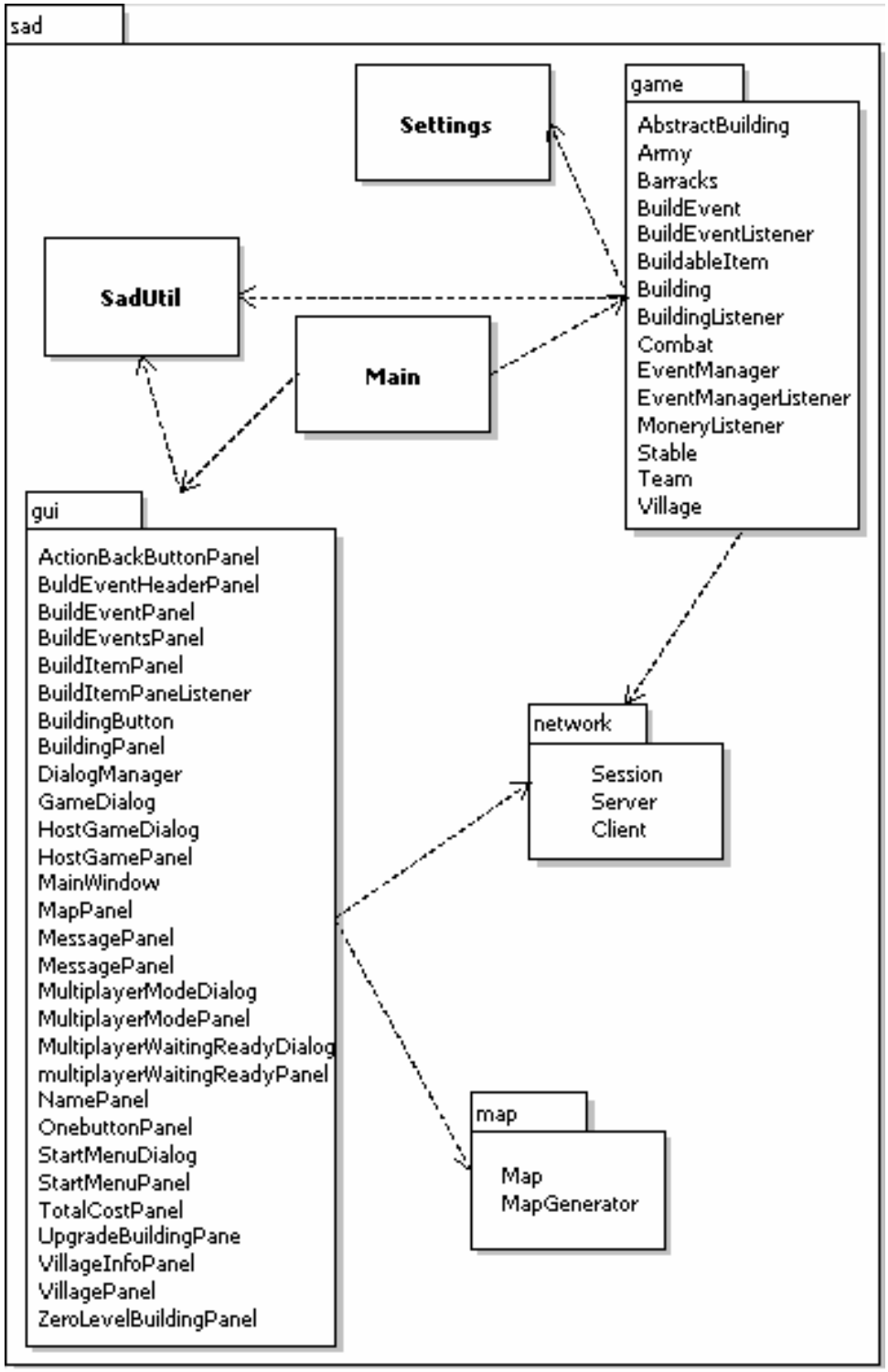
5.5.8. Interface Race

Methods:

String getName()

Method Name: getName()
Parameters: -
Return Value: The name of the race
Description: The method returns the name of the race
Data structures: -
Pre-conditions: -
Validity Checks, Errors, and other Anomalous Situations: -
Post-conditions: The name of the race is returned
Called by: BuildingPanel update
Calls: -

5.6. Package Diagram



Figur 1. Package Diagram