**Settle And Destroy (SAD)**
**Group 13**
**Jonas Wikberg**
**Christofer Hjalmarsson**
**Daniel Westerberg**
**Saul Amram**
**André Sikborn Erixon**

**1.**

**2.**

**3.**

**4.**

# 5. Design Details

## 5.1. Class Responsibility Collaborator (CRC) Cards

| Class Team | |
|---|---|
| **Responsabilities:** | **Collaborators:** |
| Keeps track of team information such as team name, team race etc. | Race |
| Keeps track of team resources | |

| Class Village | |
|---|---|
| **Responsabilities:** | **Collaborators:** |
| Keeps track of the village owner | Army |
| Keeps track of the village current army | Team |
| Host buildings | |

| Class Army | |
|---|---|
| **Responsabilities:** | **Collaborators:** |
| Know the home village of the army | Village |
| Keeping troops together as one unit | Troop |
| Knowing the army troop amounts | TroopType |
| To merge different armies from the same team | |

| Class Combat | |
|---|---|
| **Responsabilities:** | **Collaborators:** |
| Calculates which army wins a battle | Army |

| Class Troop | |
|---|---|
| **Responsabilities:** | **Collaborators:** |
| Keeping track of troop characteristics such as amount, attack points and defence points | TroopInfo |
| | TroopType |

| Class Map | |
|---|---|
| **Responsabilities:** | **Collaborators:** |
| Keeps track of and displays the game terrain, villages and armies | |
| Provides possibility to move | |

| Interface Building | |
|---|---|

| Responsabilities: | Collaborators: |
|---|---|
| Keeps track of building name, level, costs of upgrading | Village |
| Keeps track of its own building panel where the buildings actions are display such as different building options | BuildableItem |
| Decides which buildable items that are available | |

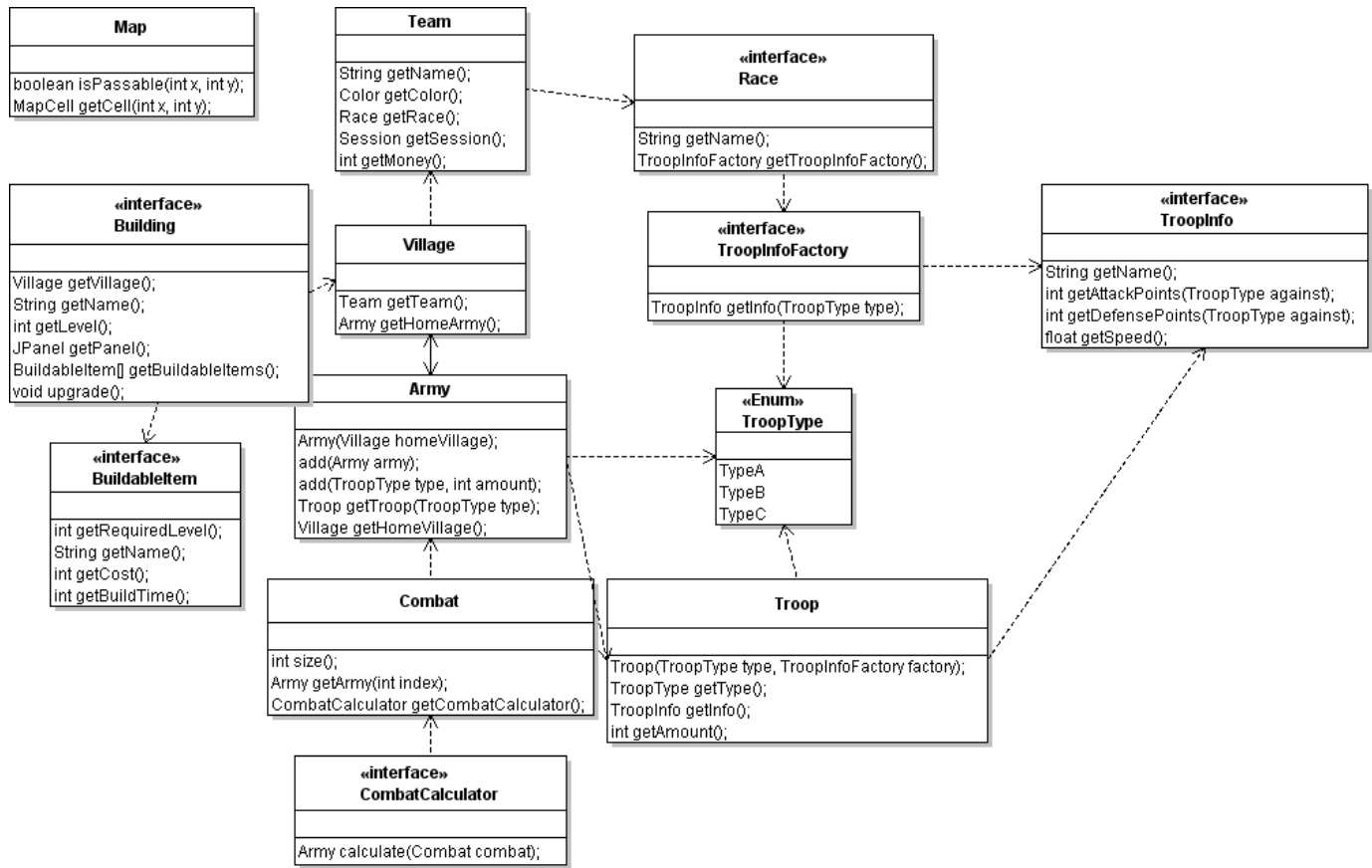| **Interface BuildableItem** | |
|---|---|
| Responsabilities: | Collaborators: |
| Describes an item that can be built in a arbitrary building | |
| Checks required building level to build building | |
| Building cost, name and building time | |

| **Interface TroopInfoFactory** | |
|---|---|
| Responsabilities: | Collaborators: |
| Defines different attributes for different types of troops | TroopInfo |

| **Interface TroopInfo** | |
|---|---|
| Responsabilities: | Collaborators: |
| Keeps track of attributes for a specific kind of troop | |

| **Interface CombatCalculator** | |
|---|---|
| Responsabilities: | Collaborators: |
| Calculates the outcome of a combat | Combat |

| **Interface Race** | |
|---|---|
| Responsabilities: | Collaborators: |
| Creates and stores a Troopinfofactory that be used by the team | TroopInfoFactory |
| Race name | |

## 5.2. Class Diagram

**Map**

boolean isPassable(int x, int y);
MapCell getCell(int x, int y);

**Team**

String getName();
Color getColor();
Race getRace();
Session getSession();
int getMoney();

**«interface»**
**Race**

String getName();
TroopInfoFactory getTroopInfoFactory();

**«interface»**
**Building**

Village getVillage();
String getName();
int getLevel();
JPanel getPanel();
BuildableItem[] getBuildableItems();
void upgrade();

**Village**

Team getTeam();
Army getHomeArmy();

**«interface»**
**TroopInfoFactory**

TroopInfo getInfo(TroopType type);

**«interface»**
**TroopInfo**

String getName();
int getAttackPoints(TroopType against);
int getDefensePoints(TroopType against);
float getSpeed();

**Army**

Army(Village homeVillage);
add(Army army);
add(TroopType type, int amount);
Troop getTroop(TroopType type);
Village getHomeVillage();

**«interface»**
**BuildableItem**

int getRequiredLevel();
String getName();
int getCost();
int getBuildTime();

**«Enum»**
**TroopType**

TypeA
TypeB
TypeC

**Combat**

int size();
Army getArmy(int index);
CombatCalculator getCombatCalculator();

**Troop**

Troop(TroopType type, TroopInfoFactory factory);
TroopType getType();
TroopInfo getInfo();
int getAmount();

**«interface»**
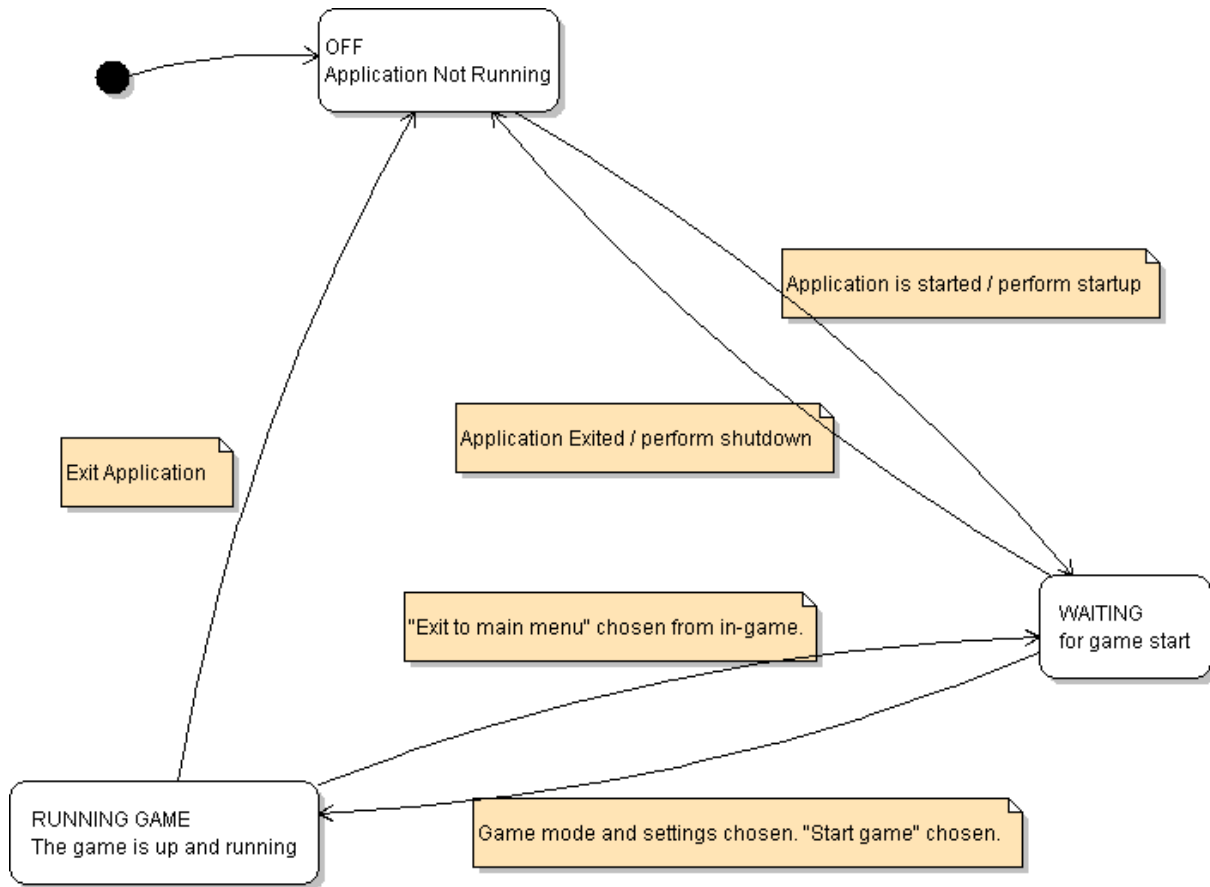**CombatCalculator**

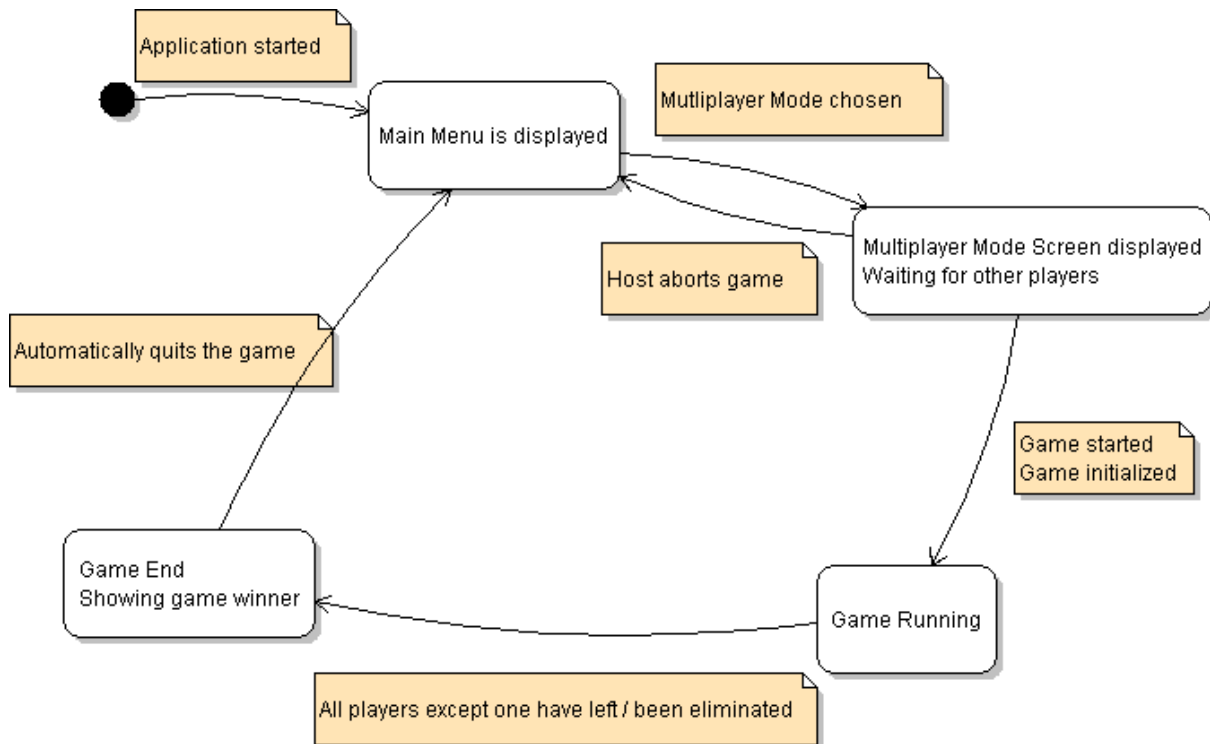Army calculate(Combat combat);

**Figur 1. Class Diagram**

## 5.3. State Charts
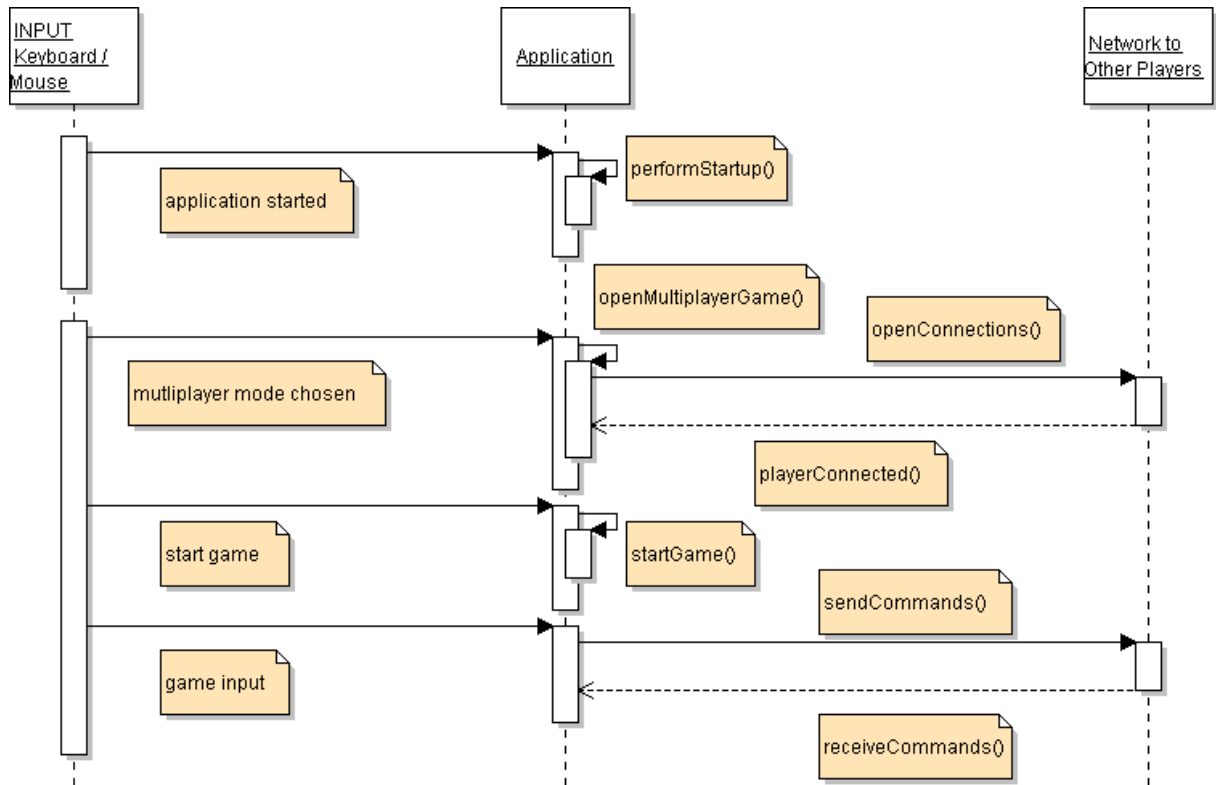
### 5.3.1. State Chart 1



**Figur 2. State Chart 1 - Overview**

### 5.3.2. State Chart 2



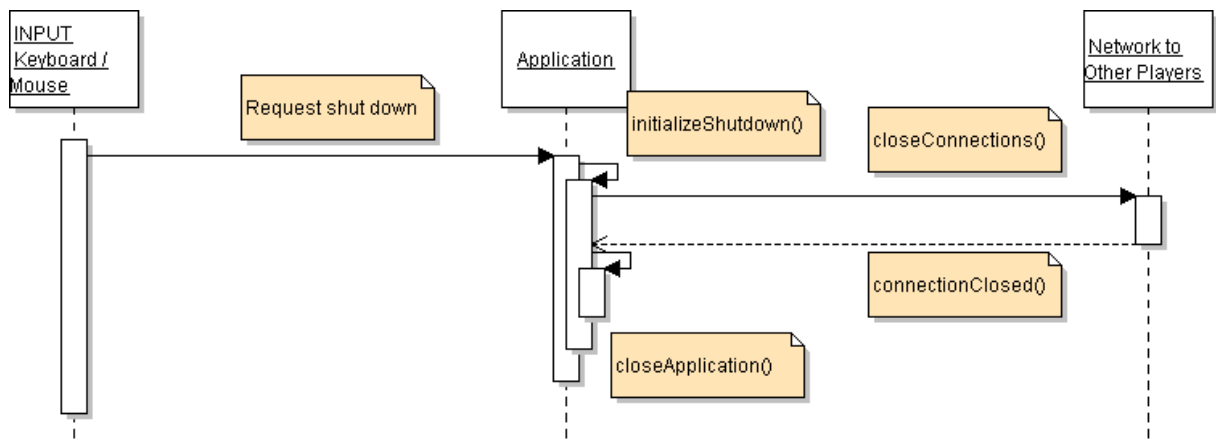**Figur 3. State Chart 2 – Staring, running and ending a game**

## 5.4. Interaction Diagrams



**Figur 4. Application start and game running**



**Figur 5. Game shutdown**