Group 15 project plan

- Scope of the work: what features, and functionalities will be implemented, how is the program used, and how does it work

Basic features

- **Simple 2D graphics**
- **Moving through corridors and rooms.**
- **Combat between the player and several types of monsters**
- **Collectibles which can be used later (weapons, potions, etc.)**
- **Some sort of progression (winning & losing conditions)**

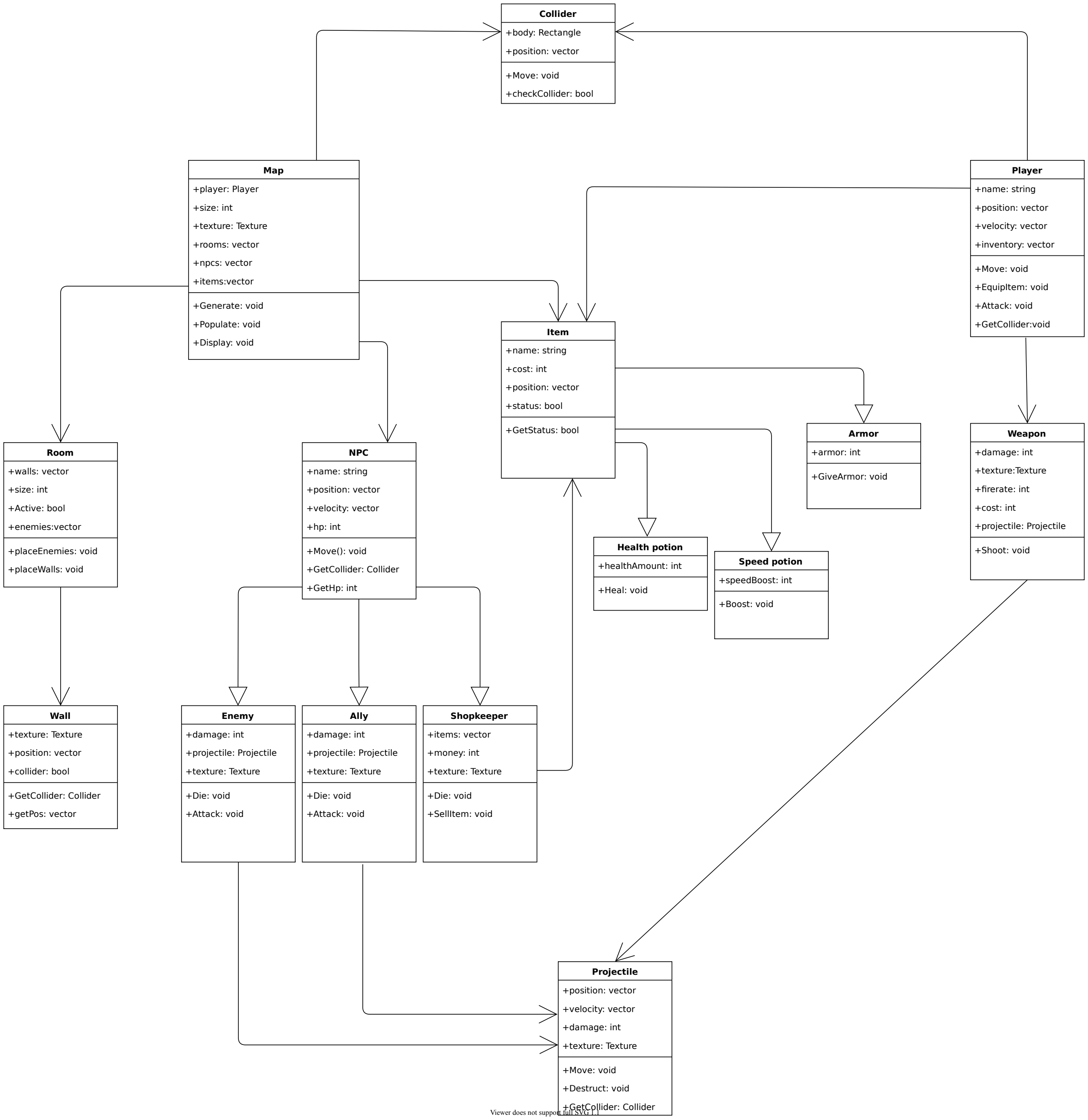
Additional features

- **Randomly generated dungeons**
- **Randomly generated monsters and items**
- Character leveling (skills & abilities)
- Ability to craft / modify items
- Quests or other events
- **Other kinds of non-player characters (shopkeepers, allied combat, etc.)**
- **Additional UI elements (dungeon map, interactive inventory, etc.)**
- **Sound effects**

Bold: Planned features, Normal: Possible features

We will implement at least all the basic features listed. The plan is to create a dungeon crawler inspired by the Binding of Isaac. The dungeon crawler will have a player character, enemies, and shopkeepers. The map will consist of randomly generated rooms that contain randomly generated enemies or a shopkeeper. The player character will be in a single room at a time. The screen will be fixed on the room. The player character will use ranged weapons. The enemy damages the player when the player gets too close to the enemy. The player will have n-amount of hp, damage results in loss of hp. Loss of all hp results in game over. The game is won by reaching the end of the dungeon. Movement is done with keyboard (WASD) and shooting with SPACE to direction where the character is facing. Other keyboard keys can be used, for example inventory management and weapon selection.

- High-level structure of the software: main modules, main classes (according to current understanding)



Main classes:

- A player class that has the following functionality: Moving, attacking, inventory management, skill points and level management, health modification (taking damage, taking armor damage and healing), currency modification (collecting currency and buying things)
- Weapon class for determining what the weapon looks like in the players inventory (the texture of weapon), the projectile that the weapon will be firing, fire rate of weapon (for ex. 5 projectiles per some time unit), special attributes such as whether the weapon has spread fire or whether projectiles penetrate walls etc...
- Projectile class for determining the damage, size, moving direction and velocity of a projectile, what the projectile looks like (texture of projectile) and possibly additional features such as whether or not the projectile penetrates NPCs or walls etc...
- Abstract item classes for items such as health potion, speed potion, grenade/bomb, teleport, ability/spell to distract enemies, invisibility cloak, armor to temporarily "increase" max hp, currency (gems, gold).
- Abstract NPC class. The classes that inherit from this class would be the following classes: Ally, Enemy, Shopkeeper
- A class for several types of walls, this class would determine what the wall looks like (the wall texture), the location of the wall in the room and additional features such as wall durability (the amount of damage the wall takes before being destroyed, if destructible) etc...
- A room class for rooms that has the following functionality: Randomization for determining what enemies and items a room contains and where in the room they are located, randomizing wall placement inside of rooms, randomizing what kind of walls are used, some functionality that connects a room to another room so that the class knows which room is a neighboring room, determining whether the room is a dungeon room or a shopkeeper's room
- A class for a map that contains an arbitrary amount of rooms
- A class for a collider handles collisions between projectiles, NPCs, player and walls
- A class for animations (movement of player, enemies, projectiles)

- Planned use of external libraries
    - We will use SFML (Simple and Fast Multimedia Library) and (possibly) IMGUI-SFML. https://www.sfml-dev.org/documentation/2.5.1/

- Division of work and responsibilities between the group

Every member has their own branch. The table contains every member's main responsibility. While these are the main responsibilities of the group members, group members may help other members with their responsibilities. Progress of each member is discussed in weekly meetings and workload is distributed equally during the project. Weekly goals will also be discussed in the meetings. The responsibility areas don't necessarily cover the whole project and a more detailed to-do-list will be used to cover gaps between responsibility areas.

Group member	Responsibilities
Elias	Enemy, collider, items
Sampo	Map, rooms, walls
Joona	SFML library use, makefile, animation
Urho	Player, weapon, Projectile

- Planned schedule and milestones before the final deadline of the project

We will have a meeting weekly, every Wednesday at 14.00.

Date	Goal
W44	-Project plan ready and commit to git. -Initial workload decided for next week
W45	-Game can be compiled and runs -Included libraries added to Git repository -Main structure of the game works (Player movement, rooms, weapons)
W46	-Map, enemies, damage, weapon -Randomly generated map is accessible, movement through rooms work

	-Enemies spawn to rooms and can take damage from player (projectiles and collision detection) -Possibility of different weapons for player
W47	-Enemies can attack player -Inventory system implemented -Items from inventory are usable (potions, bombs, etc.) -Weapons and upgrades can be purchased from shopkeepers
W48	-Additional content -Smarter enemies and allies -More weapons and potions -Bosses
W49	-Finishing the project... -Project documentation ready