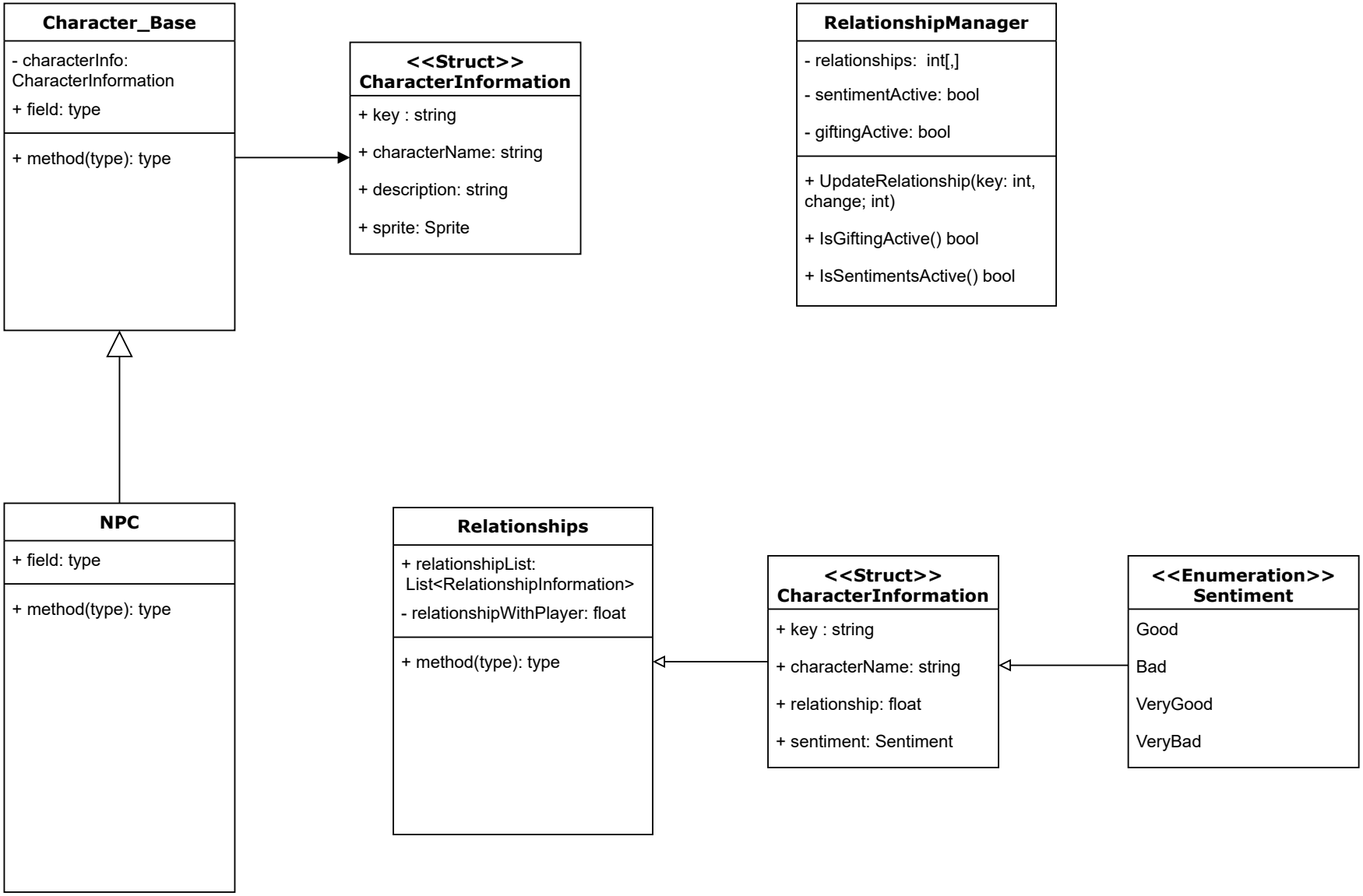
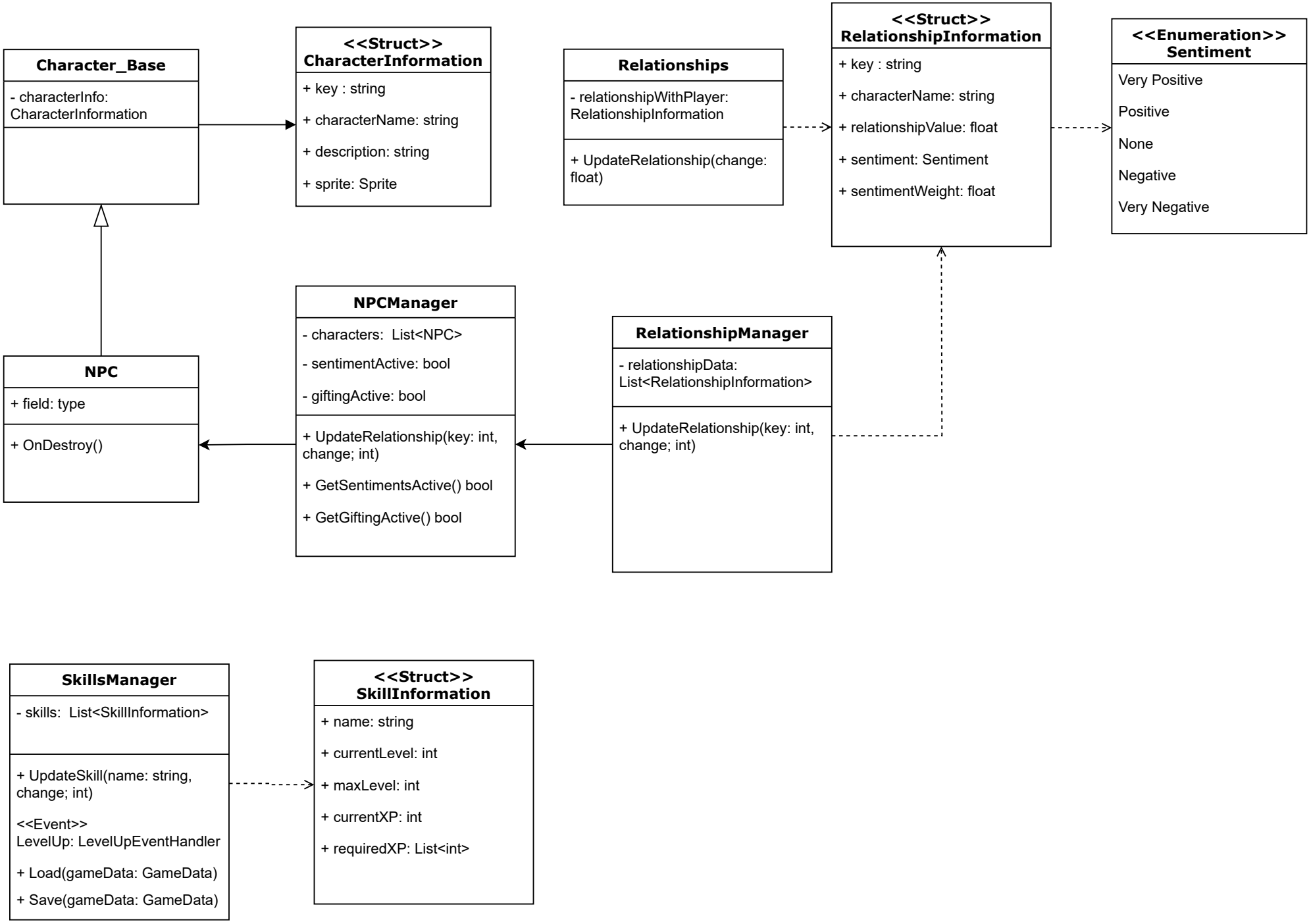

Character_Base
- characterInfo: CharacterInformation
+ field: type
+ method(type): type
<<Struct>> CharacterInformation
+ key : string
+ characterName: string
+ description: string
+ sprite: Sprite
RelationshipManager
- relationships: int[,]
- sentimentActive: bool
- giftingActive: bool
+ UpdateRelationship(key: int, change; int)
+ IsGiftingActive() bool
+ IsSentimentsActive() bool
NPC
+ field: type
+ method(type): type
Relationships
+ relationshipList: List<RelationshipInformation>
- relationshipWithPlayer: float
+ method(type): type
<<Struct>> CharacterInformation
+ key : string
+ characterName: string
+ relationship: float
+ sentiment: Sentiment
<<Enumeration>> Sentiment
Good
Bad
VeryGood
VeryBad


Character_Base
- characterInfo: CharacterInformation
<<Struct>> CharacterInformation
+ key : string
+ characterName: string
+ description: string
+ sprite: Sprite
Relationships
- relationshipWithPlayer: RelationshipInformation
+ UpdateRelationship(change: float)
<<Struct>> RelationshipInformation
+ key : string
+ characterName: string
+ relationshipValue: float
+ sentiment: Sentiment
+ sentimentWeight: float
<<Enumeration>> Sentiment
Very Positive
Positive
None
Negative
Very Negative
NPC
+ field: type
+ OnDestroy()
NPCManager
- characters: List<NPC>
- sentimentActive: bool
- giftingActive: bool
+ UpdateRelationship(key: int, change; int)
+ GetSentimentsActive() bool
+ GetGiftingActive() bool
RelationshipManager
- relationshipData: List<RelationshipInformation>
+ UpdateRelationship(key: int, change; int)
SkillsManager
- skills: List<SkillInformation>
+ UpdateSkill(name: string, change; int)
<<Event>> LevelUp: LevelUpEventHandler
+ Load(gameData: GameData)
+ Save(gameData: GameData)
<<Struct>> SkillInformation
+ name: string
+ currentLevel: int
+ maxLevel: int
+ currentXP: int
+ requiredXP: List<int>