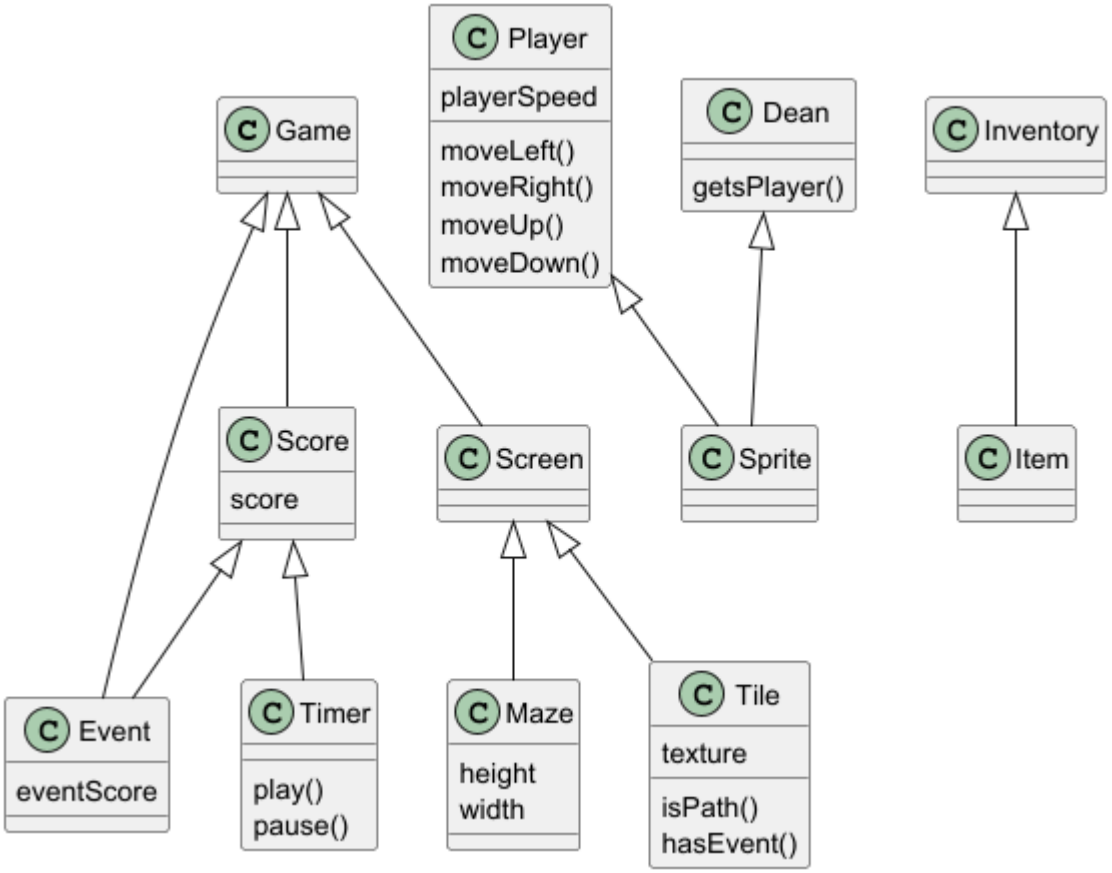
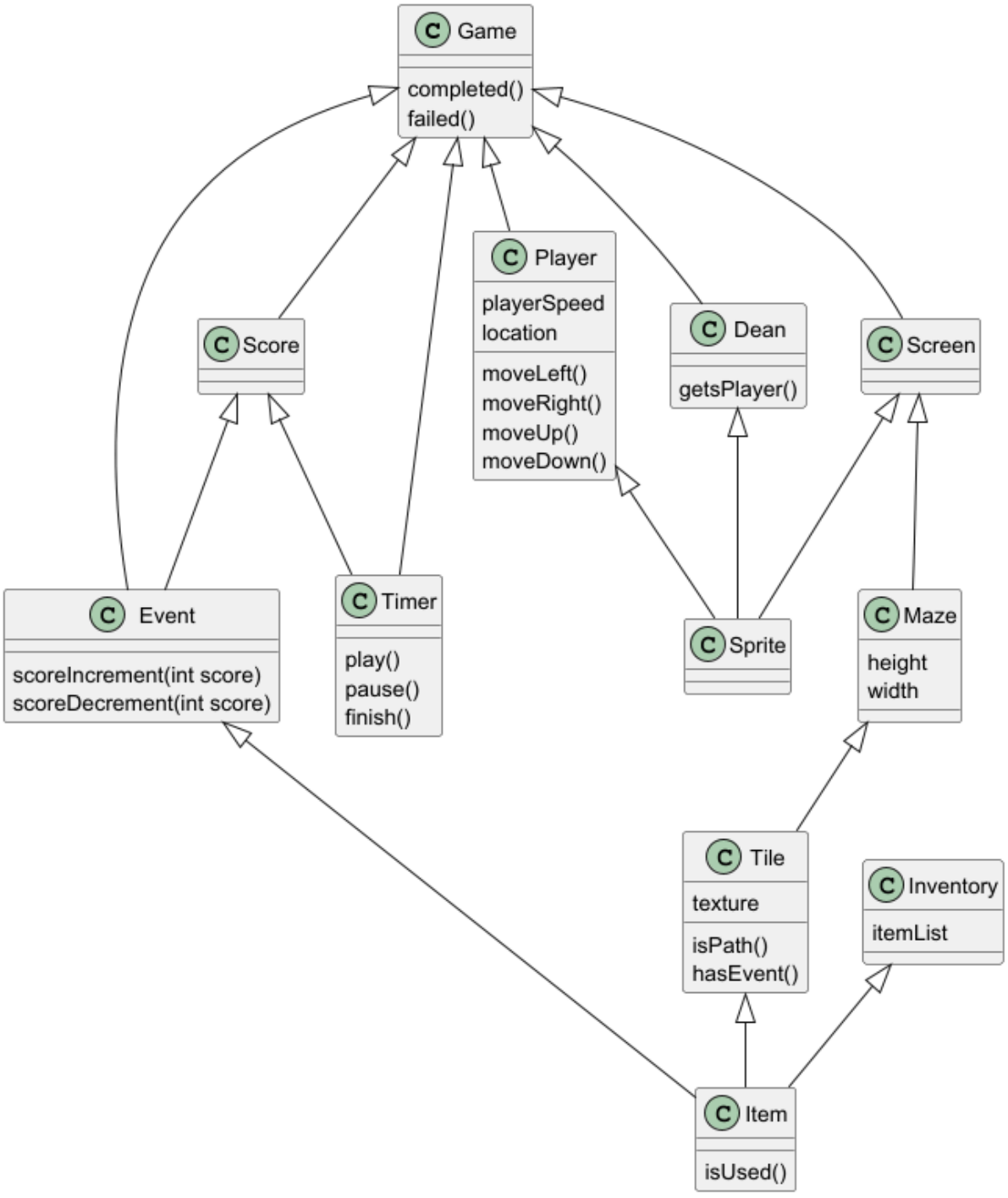
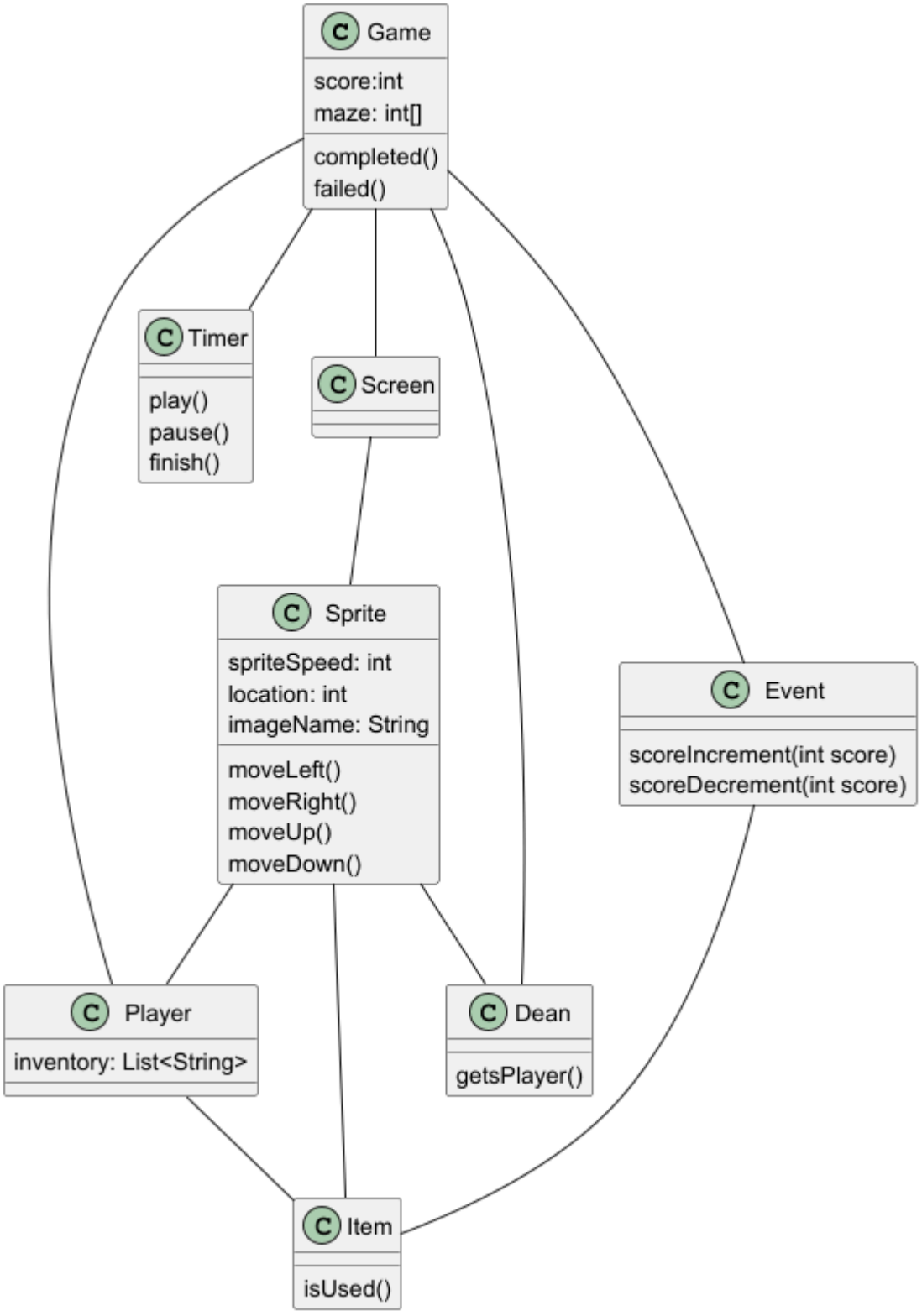
Class diagram (structural) - gives a good overview of system, how classes interact




Game
completed()
failed()
Player
playerSpeed
location
moveLeft()
moveRight()
moveUp()
moveDown()
Score
Dean
getsPlayer()
Screen
Event
scoreIncrement(int score)
scoreDecrement(int score)
Timer
play()
pause()
finish()
Sprite
Maze
height
width
Tile
texture
isPath()
hasEvent()
Inventory
itemList
Item
isUsed()



Game - main class
Score - influenced by events and score
Event - when player at location of tile, event can be called by game. Events influence score
Timer - can be paused and play again. May affect score.
Item - used by events. Can be on tile. Items are put in inventory.
Inventory - holds items.

Player - user controls movements. Player's location will be used by Game to determine if event.
Dean - appears at end of game if out of time.
Screen - contains all visual stuff
Sprite - used to hold sprite info from library
Maze - not sure yet, but keep it there if want procedural generation
Tile - can be walls/paths with different textures. Some tiles start events. Game can access tiles (not shown for abstraction)

State diagram (behavioural) - shows how the game changes with different events, how decisions influence game