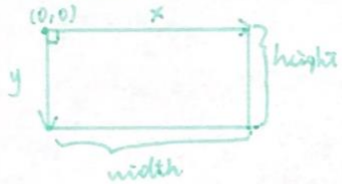
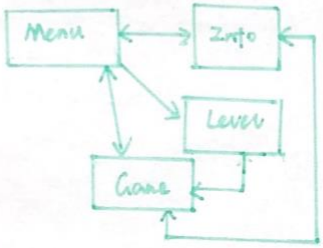
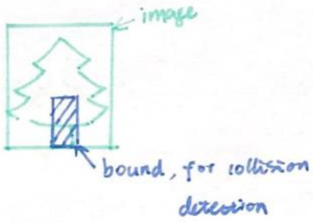
→ java coordinates:



→ states:



→ Bound: (for entity)

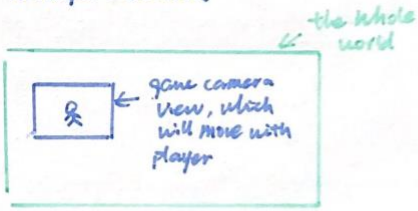


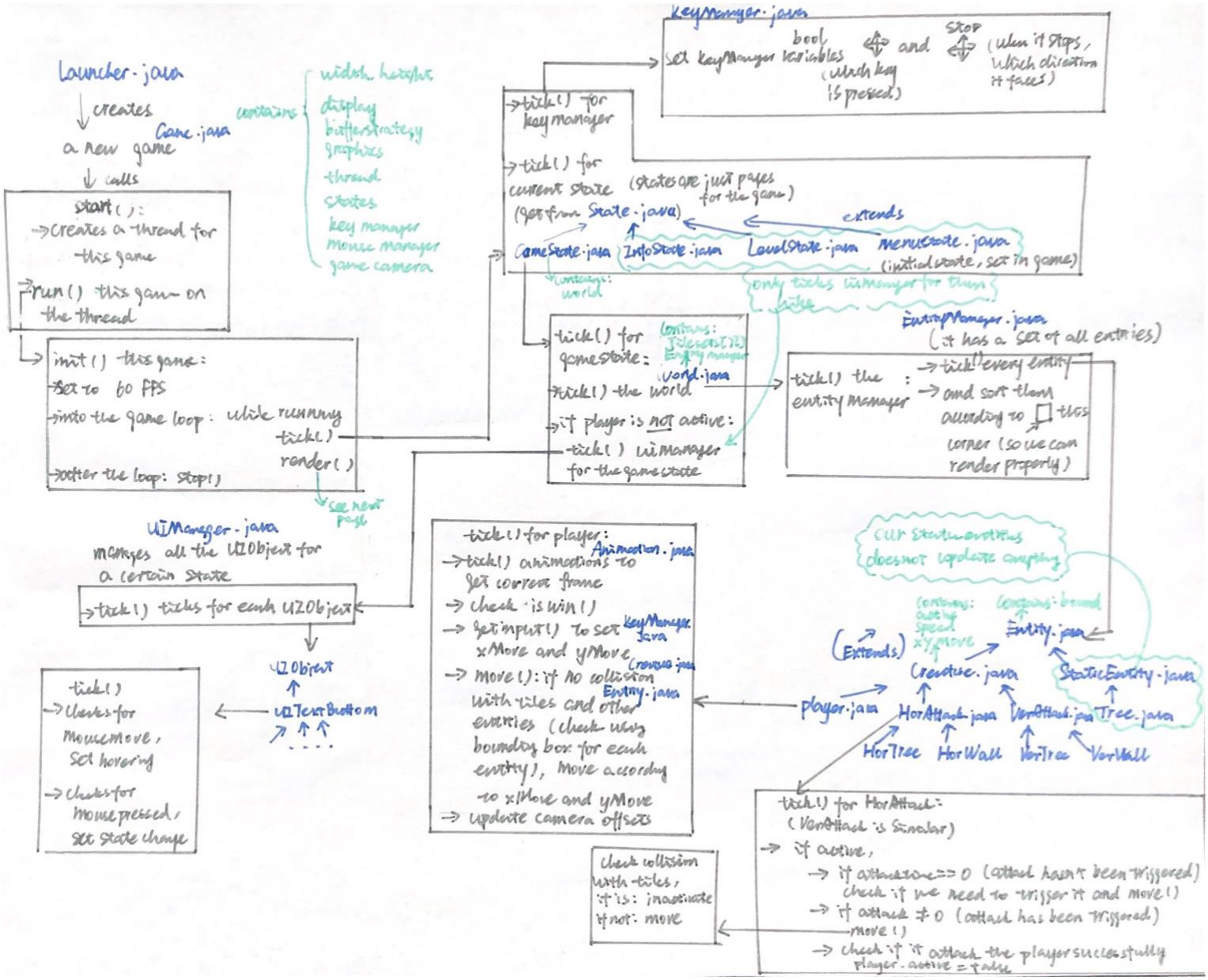
→ format for the .txt file that loads the world

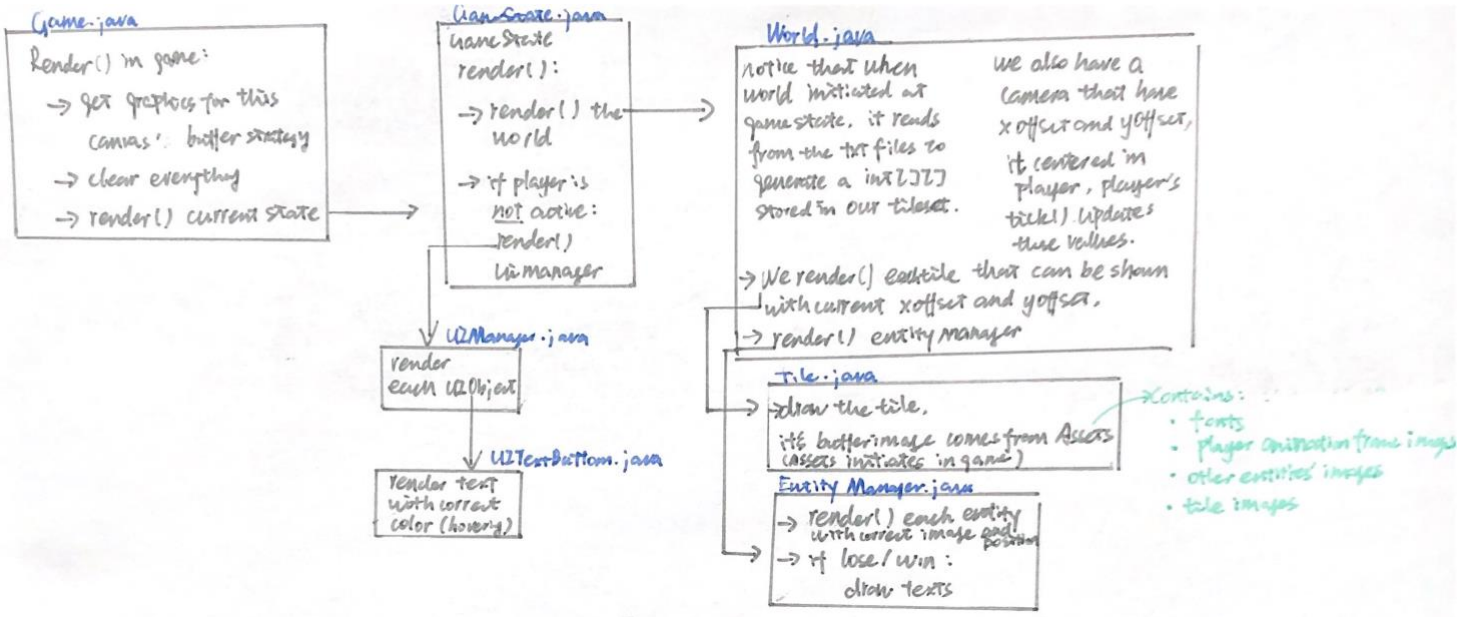
width height
2 3
start positions for player
0 0
tile types
0 0
0 1
2 0

→ Spritesheet
An image that contains all of our needed image

→ GameCamera
When entities move, their relative position to tiles changes, this has nothing to do with game camera




KeyManager.java
set keyManager variables (which key is pressed)
bool
and
stop
(when it stops, which direction it faces)
Launcher.java
creates a new game
Game.java
contains
width, height
display
bufferStrategy
graphics
thread
states
key manager
mouse manager
game camera
start():
→creates a thread for this game
→run() this game on the thread
init() this game:
→set to 60 FPS
→into the game loop: while running tick() render()
→after the loop: stop()
→tick() for key manager
→tick() for current state (get from State.java)
(states are just pages for the game)
extends
GameState.java
InfoState.java
LevelState.java
MenuState.java
(initial state, set in game)
contains: world
only ticks UIManager for them
tick() for gamestate:
(contains: EntityManager, World.java)
→tick() the world
→if player is not active:
→tick() UIManager for the gamestate
EntityManager.java
(it has a set of all entries)
tick() the entity manager:
→tick() every entity
→and sort them according to ☐ the top corner (so we can render properly)
UIManager.java
manages all the UIObject for a certain State
→tick() ticks for each UIObject
see next page
UIObject
UITextButton
tick()
→checks for mouseMove, set hovering
→checks for mousePressed, set state change
tick() for player:
Animation.java
→tick() animations to get correct frame
→check isWin()
→getInput() to set xMove and yMove
KeyManager.java
→move(): if no collision with tiles and other entities (check using bounding box for each entity), move according to xMove and yMove
Creature.java
Entity.java
→update camera offsets
check collision with tiles,
if is: inactivate
if not: move
our static entities doesnot update anything
(contains: health, speed, xy move)
(contains: bound)
(Extends)
Entity.java
Creature.java
StaticEntity.java
player.java
HorAttack.java
VerAttack.java
Tree.java
HorTree
HorWall
VerTree
VerWall
tick() for HorAttack:
(VerAttack is similar)
→ if active,
→ if attackTime == 0 (attack hasn't been triggered) check if we need to trigger it and move()
→ if attack ≠ 0 (attack has been triggered) move()
→ check if it attack the player successfully
player.active = false


Game.java
Render() in game:
-> get graphics for this canvas' buffer strategy
-> clear everything
-> render() current state
GameState.java
GameState render():
-> render() the world
-> if player is not active: render() UIManager
World.java
notice that when world initiated at gameState, it reads from the txt files to generate a int[][] stored in our tileset.
we also have a camera that have xOffset and yOffset, it centered in player, player's tick() updates these values.
-> We render() each tile that can be shown with current xoffset and yoffset.
-> render() entity manager
UIManager.java
render each UIObject
UITextButton.java
render text with correct color (hovering)
Tile.java
-> draw the tile.
its buffer image comes from Assets (assets initiates in game)
contains:
• fonts
• player animation frame images
• other entities' images
• tile images
EntityManager.java
-> render() each entity with correct image and position
-> if lose / win: draw texts