Period: 2

Group Member(s): Kassidy Tang

Group Name: kt

Project Title: Kill Time

Project Description:

        The user starts the game near the middle of the screen as an avatar with a sword in hand. As monsters start appearing from the right side of the character, the user is able to fight back using their mouse click. The sword will only hurt the monster if the monster is within range. If the user is not able to kill 3 monsters without losing all their health, the game will restart. However, if the user is able to kill 3 monsters within that level, they will progress and fight even more difficult monsters. The user will be able to upgrade their sword or get more health in between levels.

Current Functionalities:

- The avatar's body can be drawn
- The monster's body can be drawn
- The sword can be drawn
- The sword can move with the click of the mouse
- Avatar can move using the keys "a" and "d"
- The monster position also moves with the avatar
- The clouds also move with the avatar
- Game information like the number of kills is displayed on the bottom
- The sword can rotate in place
- The information at the bottom updates
- The player can click the mouse to deal damage
- The monster can deal damage
- The game can reset when the player dies
- Background changes to different times of the day
- The monster changes color after the player hits it

Planned Functionalities By Next Meeting:

- Starting screen
- Upgrades
- Monster will be drawn in more detail

Trouble:

I took some time getting the stick to rotate but I got it to work after watching some tutorials on rotation and translation. It was difficult getting the monster to change colors every time the player hit it so I changed it so that the monster only changed colors once after the initial hit. It was difficult getting the exact coordinates of the stick after it rotated to calculate when the player was doing damage, but I figured out that using coordinates based on the avatar's position solved the problem.

UML:

Driver

rotation: float
mDim: int
aPos: PVector;
mPos: PVector
cPos: PVector
sPos: PVector
curSPos: PVector
dir: PVector
monDir: PVector
backwards: PVector

kills: int
pHealth: int
monHealth: int
curHealth: int
level: int
damage: int
strength: int
deaths: int

normCol: color
normLine: color

times: StringList
timesPos: int
actualTime: String

---

setup(): void
draw(): void
updateCharacters(): void
reset(): void
mouseClicked(): void
keyPressed(): void

cloud

1 cloudPos: PVector

---

cloud(p1: PVector): cloud object
draw(): void

score

kills: int
1 pHealth: int
level: int

---

score(p1: int, p2: int, p3: int): score object
draw(): void

bg

1..* curTime: string

---

bg(p1: string): background object
draw(): void
back(p1: color, p2: color): void

avatarr

1 bodyW: int
bodyH: int
avatarPos: PVector

---

avatar(p1 : PVector): avatar object
draw(): void

sword

1 sThickness: int
sLength: int
rotation: float
swordPos: PVector

---

sword(p1: PVector): sword object
draw(): void

1..*

monster

dim: int
health: int
fCol: color
lCOl: color
monPos: PVector

---

monster(p1 : PVector, p2: int, p3: color, p4: color, p5: int): monster object
draw(): void