

Slutprojekt - JavaScript grundkurs
Handledare: David Jensen

ANIMAL WARZONE

Fia Andersson, **Max** Andersson, **Hazem** Kavas, **Edwin** Körner,
Miranda Nilhag, **Jonatan** Nyberg

Spelbeskrivning

1 vs 1, 2d top-down shooter spel.

Eliminera den andra spelaren...

...genom att skjuta bananer mot varandra.

Välj mellan fyra livsfarliga karaktärer

Vid spelstart placeras spelarna på
varsin sida av spelplanen

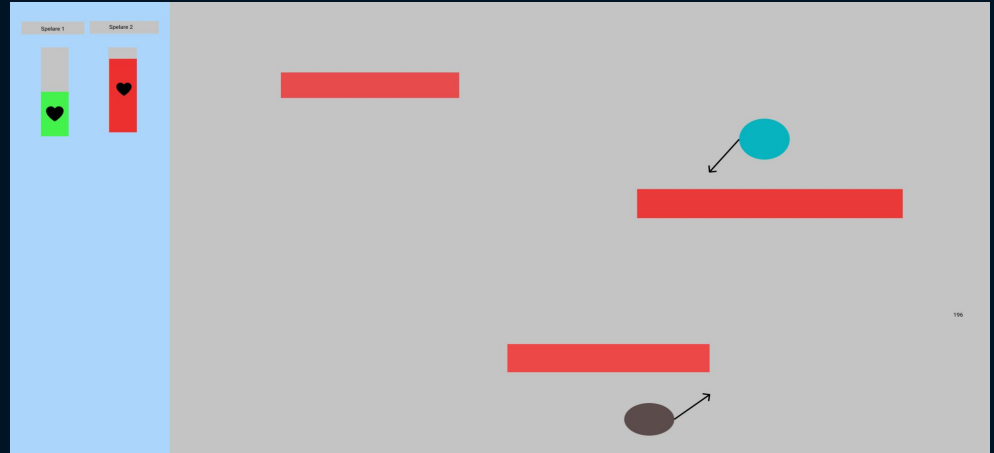
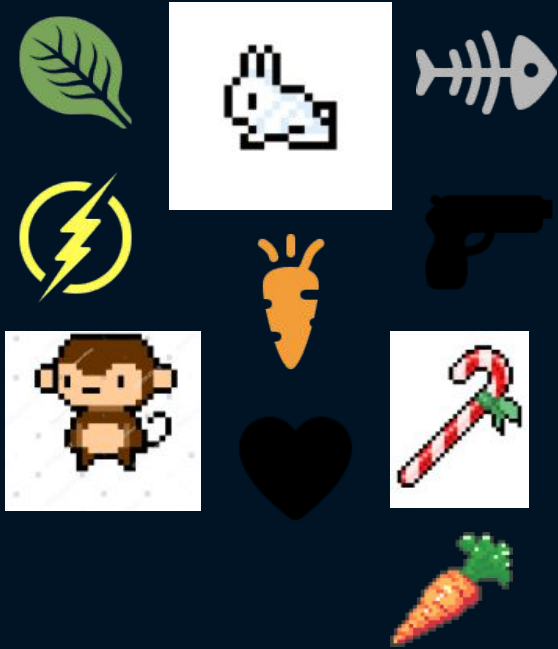
Barrikader att ta skydd mot genereras...

...och powerups för att öka dina chanser att
eliminera eller inte bli eliminerad

Varje spelare har 4 liv och den som blir
träffad och förlorar sina liv först förlorar



Designskisser 1.0

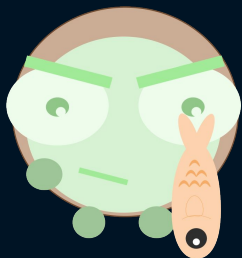


Designskisser 2.0

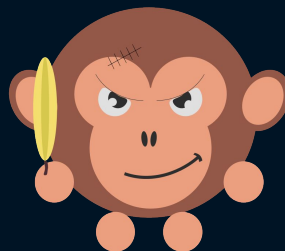
Karaktärerna



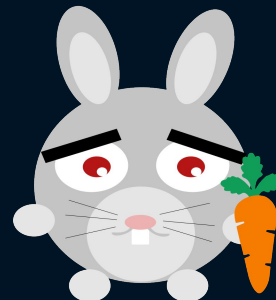
Grisen



Sköldpaddan



Apan



Kaninen

Designskisser 2.0

Entiteterna



Fart



Liv



Sköld



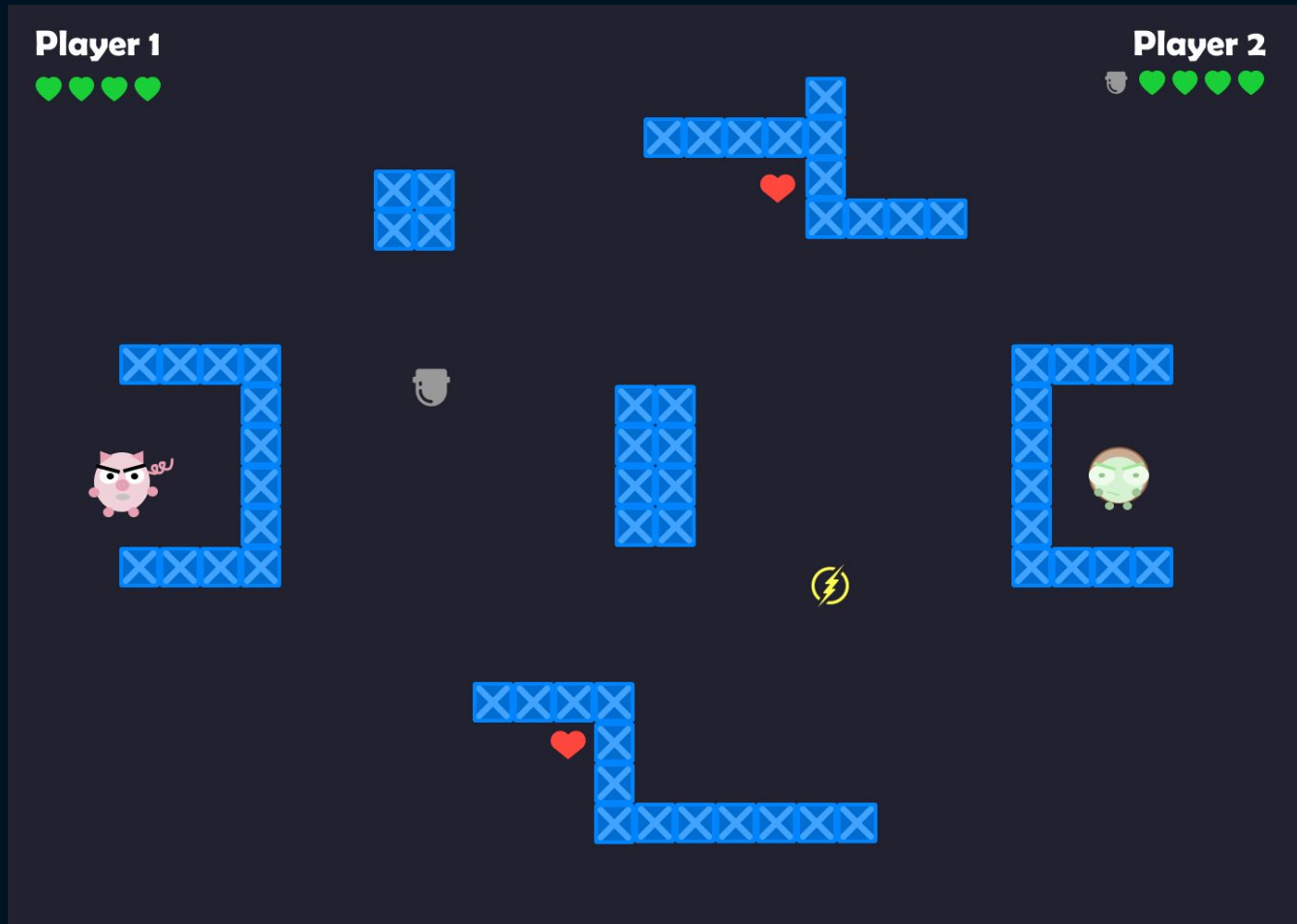
Barrikad



Vapen

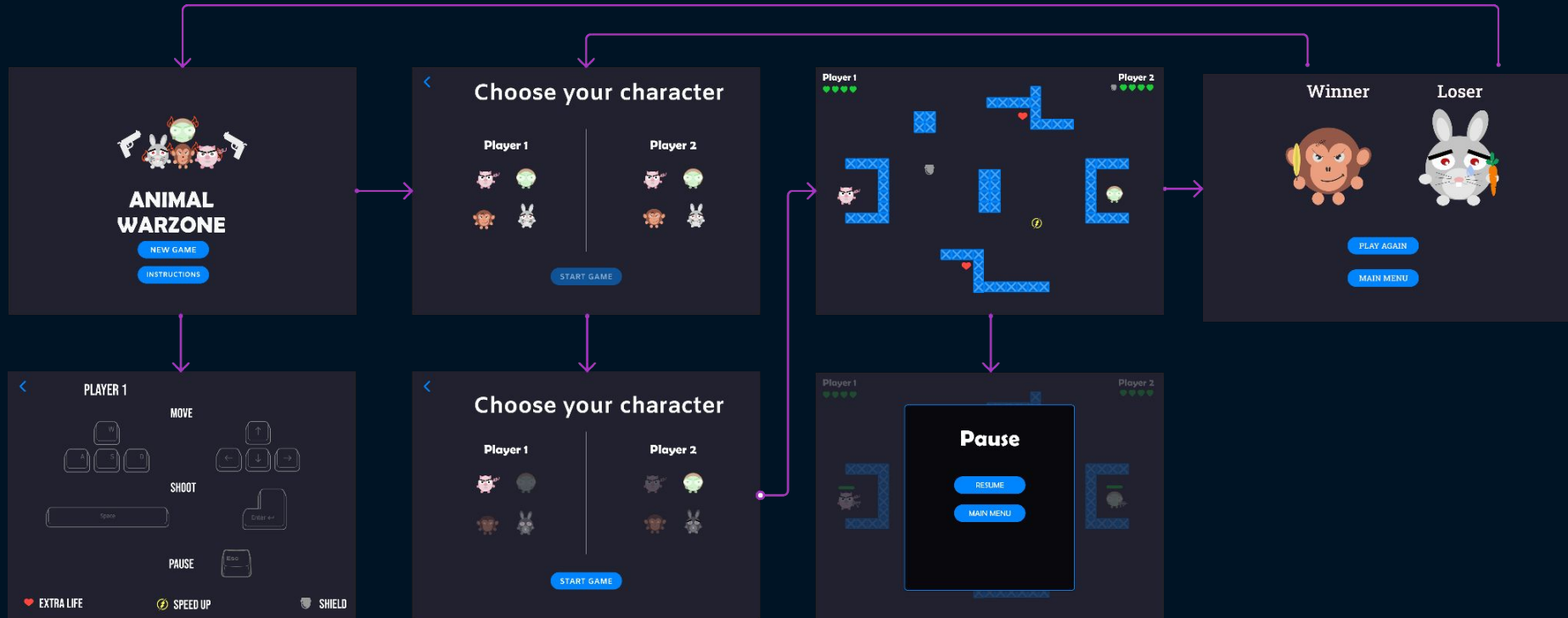
Designskisser 2.0

Spelplan

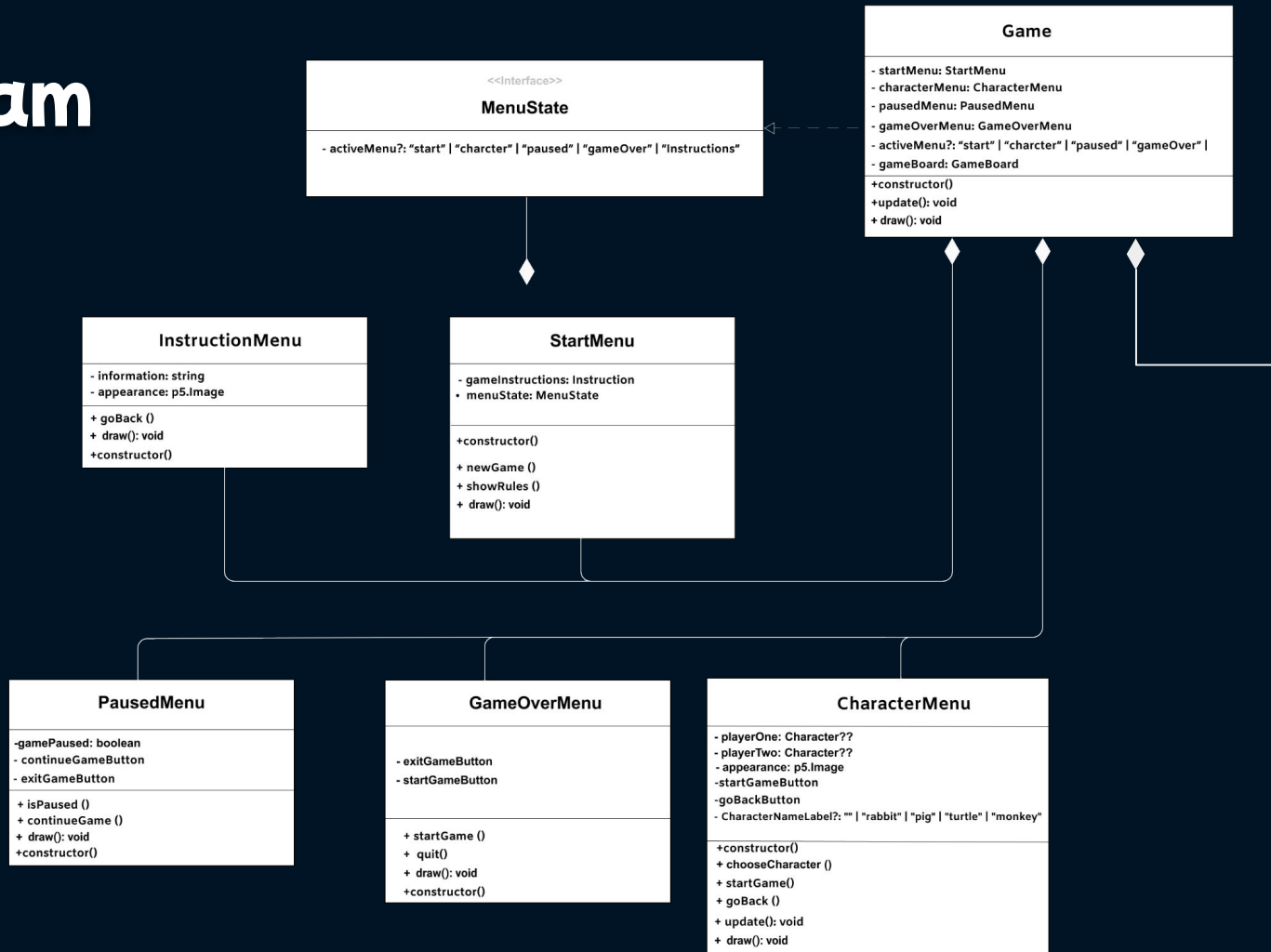


Designskisser 2.0

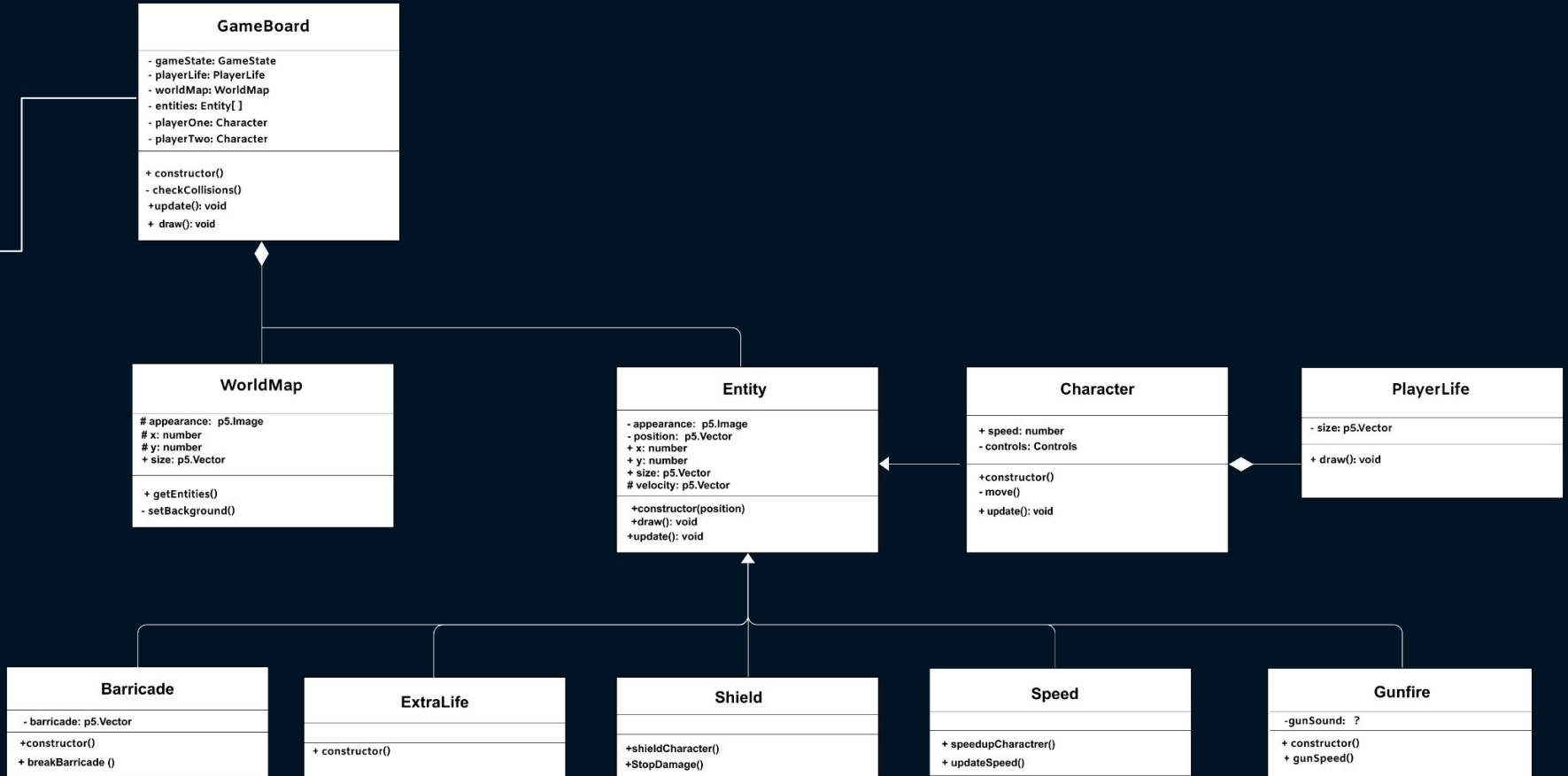
Flödet



Klassdiagram



Klassdiagram

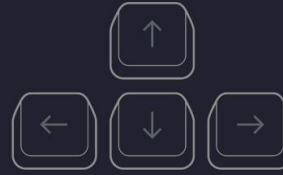
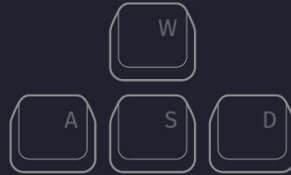


Interaktionsskiss

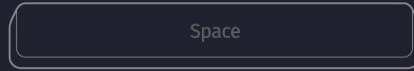


PLAYER 1

MOVE



SHOOT



PAUSE



EXTRA LIFE



SPEED UP



SHIELD

GANTT-Schema

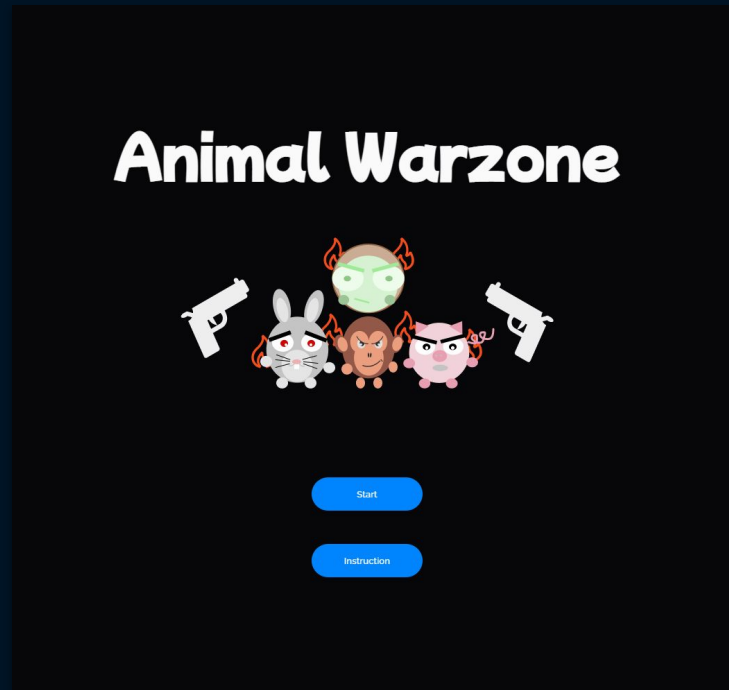


GameState

```

public Game onStartMenu() 'Start' | 'Character' | 'Paused' {
    document.getElementById(this.startMenuDivEl);
}
interfases GameState={new StartMenu(this);
}
activeGameState: GameStateLabel;
public setGameNewGame($state: GameStateLabel) => void;
public this.setGameStatesetGameStateGameStartLabel; => {
    this.startMenuDivEl.remove();
}
class Game implements GameState {
    public set activeGameState: {GameStateLabel;
    private startMenu.openStartMenu();
}

```



Character Menu

Player 1



KOD

Character Menu

```
/** PLAYER 1 */
const playerOneDiv = document.createElement("div");
playerOneDiv.className = "player1";

const playerOneText = document.createElement("h3");
playerOneText.className = "h3";
playerOneText.textContent = "Player 1";

this.characterImg = document.createElement("div");
this.characterImg.className = "choose-characters";

const leftTop = document.createElement("div");
leftTop.className = "left-top";

const leftBottom = document.createElement("div");
leftBottom.className = "left-bottom";

const divPig = document.createElement("div");
divPig.className = "div-animals";
this.pig = document.createElement("img");
this.pig.className = 'img';
this.pig.src = "../assets/images/gameCharacters/pigFront.png";
this.pig.addEventListener("click", () => this.setCharacterName("pig"));

const divTurtle = document.createElement("div");
divTurtle.className = "div-animals";
this.turtle = document.createElement("img");
this.turtle.className = 'img';
this.turtle.src = "../assets/images/gameCharacters/turtleFront.png";
this.turtle.addEventListener("click", () =>
    this.setCharacterName("turtle")
);

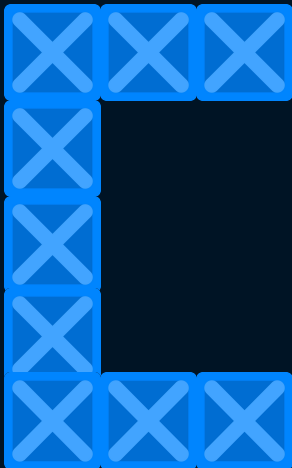
const divMonkey = document.createElement("div");
divMonkey.className = "div-animals";
this.monkey = document.createElement("img");
this.monkey.className = 'img';
this.monkey.src = "../assets/images/gameCharacters/monkeyFront.png";
this.monkey.addEventListener("click", () =>
    this.setCharacterName("monkey")
);
```

```
/** APPEND */
this.mainDiv.append(headerText, mainMenu, playersDiv, buttonDiv);
playersDiv.append(
    playerOneDiv,
    playerOneText,
    verticalLine,
    playerTwoDiv,
    playerTwoText
);

playerOneDiv.append(playerOneText, this.characterImg);
playerTwoDiv.append(playerTwoText, characterImg2);
this.characterImg.append(leftTop, leftBottom);
leftTop.append(divPig, divRabbit);
leftBottom.append(divTurtle, divMonkey);
characterImg2.append(rightTop, rightBottom);
rightTop.append(divPig2, divRabbit2);
rightBottom.append(divTurtle2, divMonkey2);
divPig.append(this.pig);
divRabbit.append(this.rabbit);
divTurtle.append(this.turtle);
divMonkey.append(this.monkey);
divPig2.append(this.pig2);
divRabbit2.append(this.rabbit2);
divTurtle2.append(this.turtle2);
divMonkey2.append(this.monkey2);
buttonDiv.append(this.startGameButton);
}
```

KOD

Barricade



```
//horizontal
for (let x = 0; x < 4; x++) {
  // up right
  entities.push(new Barricade(entites.barricade, imageX * x + width * .5, height * .16, this.size ));
  entities.push(new Barricade(entites.barricade, imageX * x + width * .5 + imageX * 5, height * .16 + imageY * 2, this.size));

  //middle-up left
  entities.push(new Barricade(entites.barricade, imageX * x + width * .08, height * .5 - imageY * 3, this.size));
  //middle-up right
  entities.push(new Barricade(entites.barricade, imageX * x + width * .92 - imageX * 3, height * .5 - imageY * 3, this.size));

  //middle-down left
  entities.push(new Barricade(entites.barricade, imageX * x + width * .08, height * .5 + imageY * 2, this.size));
  // middle-down right
  entities.push(new Barricade(entites.barricade, imageX * x + width * .92 - imageX * 3, height * .5 + imageY * 2, this.size));

  // middle-down
  entities.push(new Barricade(entites.barricade, imageX * x + width * .3, height * .7, this.size));
  entities.push(new Barricade(entites.barricade, imageX * x + width * .3 + imageX * 5, height * .7 + imageY * 3, this.size));
}
// vertical
for (let y = 0; y < 4; y++) {
  // upp right
  entities.push(new Barricade(entites.barricade, width * .5 + imageX * 4, y * imageY + height * .16 - imageY, this.size));

  //middle left
  entities.push(new Barricade(entites.barricade, width * .08 + imageX * 3, y * imageY + height * .5 - imageY * 2, this.size));
  //middle right
  entities.push(new Barricade(entites.barricade, width * .92 - imageX * 3, y * imageY + height * .5 - imageY * 2, this.size));

  // middle
  entities.push(new Barricade(entites.barricade, width * .5 - imageX / 2, y * imageY + height * .5 - imageY * 2, this.size));
  entities.push(new Barricade(entites.barricade, width * .5 + imageX / 2, y * imageY + height * .5 - imageY * 2, this.size));

  // middle-down
  entities.push(new Barricade(entites.barricade, width * .3 + imageX * 4, y * imageY + height * .7 , this.size));
```



KOD

Collision - Between GunFire and Player

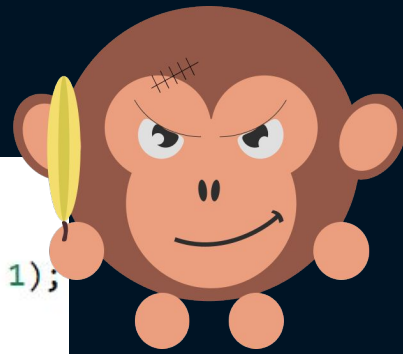
```
if (entity2 instanceof GunFire) {  
    if (entity2.playerNum !== entity1.playerNum ) {  
        sound.shieldHit.play();  
        this.entities.splice(this.entities.indexOf(entity2), 1);  
        if (!entity1.isShielding) {  
            entity1.totalLives = entity1.totalLives -1;  
            sound.drainLifeSound.play();  
            sound.shieldHit.stop();  
        }  
        if (entity1.totalLives === 0) {  
            entity1.isLosing = true;  
            this.gameState.setGameState('GameOver');  
        }  
    }  
}
```



KOD

Collision - Shield / Speed / ExtraLife

```
if (entity2 instanceof Shield) {  
    if (!entity1.isShielding) {  
        this.entities.splice(this.entities.indexOf(entity2), 1);  
        entity1.isShielding = true;  
        sound.powerupSound.play();  
    }  
}  
  
if (entity2 instanceof Speed) {  
    if (!entity1.isSpeeding) {  
        this.entities.splice(this.entities.indexOf(entity2), 1);  
        entity1.isSpeeding = true;  
        sound.powerupSound.play();  
    }  
}  
  
if (entity2 instanceof ExtraLife) {  
    this.entities.splice(this.entities.indexOf(entity2), 1);  
    entity1.totalLives = entity1.totalLives + 1;  
    sound.powerupSound.play();  
}
```



KOD

Character Image

```
public getName() {  
    return this.name;  
}
```

```
public openGameOverMenu(player1: Character, player2: Character) {  
    if (player1.isLosing) {  
        this.setLoser(player1);  
        this.setWinner(player2);  
    } else if (player2.isLosing) {  
        this.setLoser(player2);  
        this.setWinner(player1);  
    }  
  
    document.body.append(this.gameOverMainDiv);  
    music.inGameSound.stop();  
    music.gameOverSound.play();  
}
```

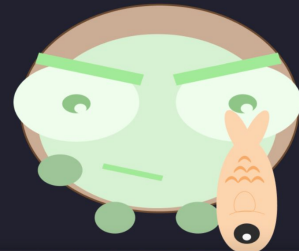
```
public setWinner(winner: Character) {  
    const name = winner.getName();  
    this.imageWinner.src = this.getCharacterImageUrl(name);  
}
```

```
private getCharacterImageUrl(name: CharacterNameLabel) {  
    if (name === 'monkey') {  
        return './assets/images/bigMonkey.png';  
    } else if (name === 'pig') {  
        return './assets/images/bigPig.png';  
    } else if (name === 'turtle') {  
        return './assets/images/bigTurtle.png';  
    } else if (name === 'rabbit') {  
        return './assets/images/bigRabbit.png';  
    } else {  
        return '';  
    }  
}
```

WINNER



LOSER

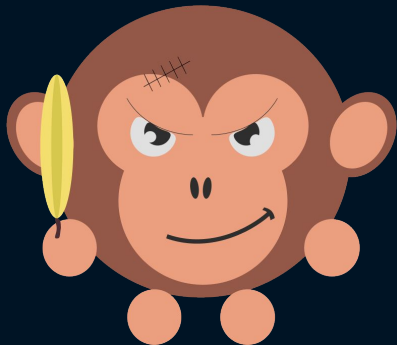


Play Again

Main Menu

DEMO

<https://animalwarzone.netlify.app/>



Reflektion

Bra saker

- Arbeta två och två
- Bra kommunikation i gruppen
- Lyhört, alla fick säga sitt
- Att arbeta remote
- David <33



Mindre bra saker

- Begränsade förkunskaper
- Svårt att dela upp
- Tidsoptimister
- Inte läst tillräckligt med dokumentation
- Satte inga deadlines



TACK FÖR OSS!

Fia Andersson, **Max** Andersson, **Hazem** Kawas, **Edwin** Körner,
Miranda Nilhag, **Jonatan** Nyberg

