# CreepyTown — 2D Horror Escape (Phaser 3)

**What this is:** a polished, single-player 2D horror/escape game built with Phaser 3 (plain HTML + JS). It’s structured so you can copy files into a GitHub repo and run it as a static site.

## Project structure

CreepyTown/  
├─ index.html  
├─ README.md  
├─ main.js  
├─ package.json (optional)  
├─ assets/  
│ ├─ player.png  
│ ├─ enemy.png  
│ ├─ tileset.png  
│ ├─ sfx/footsteps.wav  
│ ├─ sfx/scream.wav  
│ ├─ music/ambience.mp3  
│ └─ ...  
└─ data/  
 └─ map.json (optional)

**Note:** I include placeholder filenames only (you must add your own art / audio assets). The code is written to work with simple sprites so you can test immediately.

## Quick install & run

1. Create a new GitHub repo and copy the files from this document into it (same structure).
2. Push to GitHub.
3. In GitHub repo -> Settings -> Pages -> select the branch and / (root) and publish (or use GitHub Actions to build if you add bundling). The site will be available as a static page.

Locally, you can run a static server:

# using Python 3  
python -m http.server 8000  
# then visit http://localhost:8000

## How the code is organized in this document

Below are the main files — copy each into the project root (create assets/ and add assets as named).

### index.html

<!doctype html>  
<html lang="en">  
<head>  
 <meta charset="utf-8" />  
 <meta name="viewport" content="width=device-width,initial-scale=1" />  
 <title>CreepyTown — Escape</title>  
 <style>  
 html,body { height:100%; margin:0; background:#000; }  
 #game { width:100%; height:100vh; display:block; }  
 .notice { position:fixed; left:8px; top:8px; color:#ccc; font-family:Inter, system-ui, sans-serif; z-index:999; }  
 </style>  
</head>  
<body>  
 <div class="notice">WASD/Arrows to move • Space to toggle flashlight • Shift to sprint</div>  
 <div id="game"></div>  
 <script src="https://cdn.jsdelivr.net/npm/phaser@3.60.0/dist/phaser.min.js"></script>  
 <script src="main.js"></script>  
</body>  
</html>

### main.js

/\* main.js — Phaser 3 game. Single-file for simplicity. \*/  
  
class BootScene extends Phaser.Scene {  
 constructor() { super('Boot'); }  
 preload(){  
 // small placeholder to keep user informed  
 this.load.image('pixel','assets/pixel.png');  
 }  
 create(){ this.scene.start('Preload'); }  
}  
  
class PreloadScene extends Phaser.Scene {  
 constructor(){ super('Preload'); }  
 preload(){  
 // UI  
 this.load.image('ui\_panel','assets/ui\_panel.png');  
 // Player & enemy  
 this.load.spritesheet('player','assets/player.png',{ frameWidth:32, frameHeight:48 });  
 this.load.spritesheet('enemy','assets/enemy.png',{ frameWidth:32, frameHeight:48 });  
 // tiles + light mask  
 this.load.image('tiles','assets/tileset.png');  
 this.load.image('bg','assets/bg.png');  
 // audio  
 this.load.audio('ambience','assets/music/ambience.mp3');  
 this.load.audio('scream','assets/sfx/scream.wav');  
 this.load.audio('foot','assets/sfx/footsteps.wav');  
 }  
 create(){  
 this.scene.start('Town');  
 }  
}  
  
class TownScene extends Phaser.Scene {  
 constructor(){ super('Town'); }  
 create(){  
 // world size  
 const W = 2400, H = 1600;  
 // background  
 this.add.tileSprite(0,0,W,H,'bg').setOrigin(0).setScrollFactor(1);  
  
 // simple static world boundaries  
 this.physics.world.setBounds(0,0,W,H);  
  
 // player  
 this.player = this.physics.add.sprite(400, 300, 'player', 0);  
 this.player.setCollideWorldBounds(true);  
 this.player.speed = 140;  
 this.player.sprintSpeed = 240;  
 this.player.isHidden = false;  
  
 // enemy group  
 this.enemies = this.physics.add.group();  
 for(let i=0;i<5;i++){  
 const x = Phaser.Math.Between(600, W-200);  
 const y = Phaser.Math.Between(200, H-200);  
 const e = this.enemies.create(x,y,'enemy',0);  
 e.patrolAngle = Phaser.Math.Between(0,360);  
 e.detectionRadius = 220;  
 e.speed = Phaser.Math.Between(30,65);  
 e.setCollideWorldBounds(true);  
 }  
  
 // camera  
 this.cameras.main.startFollow(this.player, true, 0.12, 0.12);  
 this.cameras.main.setBounds(0,0,W,H);  
 this.cameras.main.setBackgroundColor('#000000');  
  
 // light & shadow - using a light mask  
 this.lightTexture = this.make.renderTexture({ width: W, height: H, add: true });  
 this.darkness = this.add.rectangle(0,0,W,H,0x000000,1).setOrigin(0);  
 this.darkness.setDepth(50);  
  
 // flashlight properties  
 this.flashOn = true;  
 this.flashRadius = 220;  
  
 // controls  
 this.cursors = this.input.keyboard.createCursorKeys();  
 this.keys = this.input.keyboard.addKeys('W,A,S,D,SPACE,SHIFT');  
  
 // collisions  
 this.physics.add.overlap(this.player, this.enemies, ()=>{ this.playerCaught(); }, null, this);  
  
 // objective  
 this.escapeZone = this.add.zone(W-120, H-120, 200, 200).setOrigin(0).setRectangleDropZone(200,200);  
 this.physics.world.enable(this.escapeZone);  
 this.escapeZone.body.setAllowGravity(false);  
 this.escapeZone.setDepth(0);  
 this.escapeRect = this.add.rectangle(W-120, H-120, 200,200).setStrokeStyle(2,0x66ff66,0.6).setOrigin(0).setAlpha(0.15);  
  
 this.physics.add.overlap(this.player, this.escapeZone, ()=>{ this.win(); }, null, this);  
  
 // audio  
 this.amb = this.sound.add('ambience',{ loop:true, volume:0.5 });  
 this.amb.play();  
  
 // HUD  
 this.hud = this.add.text(10,10, 'Objective: Reach the green zone to escape', { font: '16px monospace', fill:'#ddd' }).setScrollFactor(0).setDepth(100);  
  
 // create simple animations  
 this.createAnims();  
 }  
  
 createAnims(){  
 this.anims.create({ key:'walk', frames: this.anims.generateFrameNumbers('player', {start:0, end:3}), frameRate:8, repeat:-1 });  
 this.anims.create({ key:'idle', frames: [{ key:'player', frame:0 }], frameRate:1 });  
 this.anims.create({ key:'enemy\_walk', frames: this.anims.generateFrameNumbers('enemy', {start:0,end:3}), frameRate:6, repeat:-1 });  
 }  
  
 update(t,dt){  
 // player movement  
 let vx = 0, vy = 0;  
 if (this.keys.W.isDown || this.cursors.up.isDown) vy = -1;  
 if (this.keys.S.isDown || this.cursors.down.isDown) vy = 1;  
 if (this.keys.A.isDown || this.cursors.left.isDown) vx = -1;  
 if (this.keys.D.isDown || this.cursors.right.isDown) vx = 1;  
 const isSprinting = this.keys.SHIFT.isDown;  
 const speed = (vx !==0 || vy!==0) ? (isSprinting ? this.player.sprintSpeed : this.player.speed) : 0;  
 if (speed>0){  
 const len = Math.sqrt(vx\*vx + vy\*vy) || 1;  
 this.player.body.setVelocity((vx/len)\*speed, (vy/len)\*speed);  
 this.player.anims.play('walk', true);  
 } else { this.player.body.setVelocity(0,0); this.player.anims.play('idle', true); }  
  
 // flashlight toggle  
 if (Phaser.Input.Keyboard.JustDown(this.keys.SPACE)) this.flashOn = !this.flashOn;  
  
 // simple enemy behavior  
 this.enemies.getChildren().forEach(e=>{  
 // vector from enemy to player  
 const dx = this.player.x - e.x;  
 const dy = this.player.y - e.y;  
 const dist = Math.sqrt(dx\*dx+dy\*dy);  
 if (dist < e.detectionRadius && !this.player.isHidden){  
 // chase  
 const vx = dx/dist \* (e.speed + (dist<120?20:0));  
 const vy = dy/dist \* (e.speed + (dist<120?20:0));  
 e.body.setVelocity(vx,vy);  
 e.anims.play('enemy\_walk', true);  
 // scary audio when very close  
 if (dist<80 && !this.\_screamPlayed){ this.sound.play('scream', { volume:0.6 }); this.\_screamPlayed=true; }  
 } else {  
 // patrol slowly  
 e.body.setVelocity(Math.cos(e.patrolAngle)\*e.speed\*0.2, Math.sin(e.patrolAngle)\*e.speed\*0.2);  
 e.patrolAngle += 0.002;  
 e.anims.play('enemy\_walk', true);  
 }  
 });  
  
 // draw lighting mask  
 this.drawLighting();  
 }  
  
 drawLighting(){  
 // faster: render a black rectangle then cut a circle around player to simulate flashlight  
 const cam = this.cameras.main;  
 const width = this.physics.world.bounds.width;  
 const height = this.physics.world.bounds.height;  
 this.lightTexture.clear();  
 // fill with full black  
 this.lightTexture.fill(0x000000, 0.95);  
  
 if (this.flashOn){  
 const px = this.player.x;  
 const py = this.player.y;  
 // gradient circle  
 const g = this.make.graphics({ x:0, y:0, add:false });  
 const r = this.flashRadius;  
 const steps = 24;  
 for(let i=steps;i>0;i--){  
 const alpha = 0.04 \* (i);  
 g.fillStyle(0xffffff, alpha);  
 g.fillCircle(px, py, r \* (i/steps));  
 }  
 this.lightTexture.draw(g, 0, 0);  
 g.destroy();  
 }  
  
 // reduce darkness where enemies are in the light (so they are visible)  
 this.enemies.getChildren().forEach(e=>{  
 const dx = e.x - this.player.x; const dy = e.y - this.player.y;  
 const dist = Math.sqrt(dx\*dx+dy\*dy);  
 if (dist < this.flashRadius){  
 const g = this.make.graphics({ add:false });  
 g.fillStyle(0xffffff, 0.12);  
 g.fillCircle(e.x, e.y, 50);  
 this.lightTexture.draw(g,0,0);  
 g.destroy();  
 }  
 });  
  
 // apply mask to darkness rectangle  
 this.darkness.setMask(new Phaser.Display.Masks.BitmapMask(this, this.lightTexture));  
 this.darkness.setDepth(50);  
 }  
  
 playerCaught(){  
 // simple caught behavior  
 this.cameras.main.flash(600,255,0,0);  
 this.amb.stop();  
 this.sound.play('scream');  
 this.scene.restart();  
 }  
  
 win(){  
 this.amb.stop();  
 this.add.text(this.cameras.main.midPoint.x - 180, this.cameras.main.midPoint.y, 'You escaped! — Refresh to play again', { font:'28px monospace', fill:'#fff' }).setScrollFactor(0).setDepth(200);  
 this.scene.pause();  
 }  
}  
  
const config = {  
 type: Phaser.AUTO,  
 parent: 'game',  
 width: 960,  
 height: 640,  
 physics: { default: 'arcade', arcade: { debug:false } },  
 scene: [ BootScene, PreloadScene, TownScene ]  
};  
  
const game = new Phaser.Game(config);  
  
// OPTIONAL: expose for debugging  
window.\_\_game = game;

## README.md (copy into repo root)

# CreepyTown — 2D Horror Escape  
  
A small polished 2D horror escape game built with Phaser 3. Drop your assets into `assets/` and serve the folder as a static site.  
  
## Controls  
- WASD / Arrow keys: Move  
- Shift: Sprint (makes noise)  
- Space: Toggle flashlight  
  
## Notes  
- Replace placeholder assets in `assets/` for visuals and audio.  
- The game uses a render texture as a light mask to create flashlight/shadow effect — tweak `flashRadius` in `main.js` to change.  
  
## Deploy  
- Push to GitHub and enable GitHub Pages to serve as a static site.

## Asset suggestions

* Player/enemy: 32x48 sprite sheets (4 frames) with simple walking frames.
* tileset.png: 64x64 tilesheet (optional)
* bg.png: subtle tiled background (2000x1200+)
* Sounds: free ambient loops, close-up scream.wav, footsteps.

## Want me to:

* 1. Turn this into a multi-scene game with building interiors and a tilemap? (I can expand the code.)
  2. Add pathfinding to enemies? (A\*)
  3. Create GitHub-ready project with package.json, build scripts, and GitHub Actions?

Tell me which and I’ll add it.

*Copy any of the code blocks above into files in your repo and the game will run.*