Complete Antigravity Prompt + AEO Implementation Guide

Build "Escape Velocity: The Digital Employee Journey" Interactive Experience

Target: Deploy production-ready interactive learning game with full AEO optimization
Platforms: HostGator or Vercel
Timeline: 3-5 days with Antigravity agents
Goal: Drive Skool sign-ups, demonstrate Plinko Solutions value, dominate answer engines

---

PART 1: COMPLETE ANTIGRAVITY PROMPT

Copy this entire prompt into your Antigravity workspace:

PROJECT: Escape Velocity - The Digital Employee Journey

Interactive Browser-Based Learning Experience for Plinko Solutions

MISSION

Build a gamified interactive website that teaches AI automation concepts through gameplay inspired by Google Interland. Users progress through 3 scenes, learning about "Business Antigravity" (removing operational weight through digital employees). The experience replaces traditional landing pages with educational gameplay that drives Skool community sign-ups.

TECHNICAL REQUIREMENTS

Core Stack

- **Pure HTML/CSS/JavaScript** (no framework dependencies)
- **Phaser.js 3.x** for game engine (CDN link)
- **Responsive design** (mobile-first, works on all devices)
- **LocalStorage** for progress tracking
- **No backend required** (static site deployment)
- **Zero external dependencies** beyond Phaser CDN
- **Optimized for performance** (< 3 second load time)

Browser Support

- Chrome/Edge (latest 2 versions)
- Safari (iOS + macOS)
- Firefox (latest)
- Mobile Safari (iOS 14+)
- Chrome Mobile (Android)

Deployment Targets

- **HostGator**: Upload via FTP/cPanel File Manager
- **Vercel**: GitHub integration or drag-drop deploy
- Must include `.htaccess` (HostGator) or `vercel.json` (Vercel) config

SCENE 1: THE GRAVITY WELL (Problem State)

Visual Design

Theme: Cluttered office environment drowning in manual tasks
Perspective: 2D side-scrolling view (like classic platformer)
Color Palette:

- Background: Deep space blue (#0B0D17)
- UI Elements: Neon cyan (#00F0FF)
- Task Cards: Orange/red gradient (stress indicator)
- Character: Simple sprite with subtle animation

Game Mechanics

1. **Character Movement**
    - Arrow keys: Left/Right movement
    - Space bar: Jump (low gravity feel)
    - Click/Tap: Mobile touch controls
2. **Task Card Collection**
    - 10 floating task cards scattered across scene
    - Each card shows: Icon + "5 hrs/week" time cost
    - Collection triggers animation: Card flies to "Gravity Meter"
    - Sound effect: Whoosh + weight increase tone
3. **Gravity Meter (Top Right)**
    - Visual: Circular gauge filling with red
    - Text: "Operational Weight: X hours/week"
    - Increments: +5 hours per task collected
    - Goal: Collect all 10 tasks (50 hours total weight)
4. **Environmental Storytelling**
    - Animated background: Papers flying, emails pinging
    - Parallax scrolling: Multiple depth layers
    - Clock animation: Hands spinning fast (time pressure)
    - "Weight" visual: Screen slowly tilts/warps as meter fills

Task Card Examples (10 Total)

1. "Lead Follow-Up" - 8 hrs/week
2. "Data Entry" - 12 hrs/week
3. "Email Triage" - 5 hrs/week
4. "Calendar Management" - 3 hrs/week
5. "CRM Updates" - 7 hrs/week
6. "Reporting" - 6 hrs/week
7. "Invoice Processing" - 4 hrs/week
8. "Meeting Notes" - 3 hrs/week
9. "Contract Tracking" - 5 hrs/week
10. "Research Tasks" - 7 hrs/week

Completion Trigger

- **Condition**: All 10 tasks collected
- **Animation**: Screen "crushes" under weight, fade to black
- **Transition**: "But what if you could remove the weight?" text appears
- **CTA**: "Press SPACE to Launch" (proceeds to Scene 2)

Code Structure

```
// Scene1.js - Phaser Scene Class
class GravityWell extends Phaser.Scene {
constructor() { super({ key: 'GravityWell' }); }

preload() {
// Load sprites, sounds, backgrounds
}

create() {
// Initialize physics, player, task cards, gravity meter
// Set up collision detection
// Configure mobile touch controls
}

update() {
// Handle player movement
// Check task collection
// Update gravity meter
// Monitor completion state
}

collectTask(player, task) {
// Animate collection
// Update meter
// Save progress to localStorage
// Check if all tasks collected
}
}
```

SCENE 2: THE LAUNCHPAD (Automation Builder)

Visual Design

Theme: Mission control dashboard + drag-drop workflow builder
Perspective: Top-down interface (dashboard view)
Layout: Split-screen

- Left: Draggable task blocks (from Scene 1)
- Center: Automation flowchart canvas
- Right: "Digital Employee" preview panel

Game Mechanics

1. **Drag-and-Drop Workflow Builder**
    - Drag task blocks from left palette
    - Drop onto canvas to create automation nodes
    - Connect nodes with visual "data flow" lines
    - Each connection removes "weight" from meter
2. **Automation Categories** (Color-coded)
    - **Lead Management** (Blue): Follow-ups, CRM updates, outreach
    - **Data Processing** (Green): Entry, validation, enrichment
    - **Communication** (Purple): Emails, scheduling, notifications
    - **Reporting** (Orange): Dashboards, analytics, exports
3. **Weight Removal Mechanic**
    - Each automation created: Visual "weight lifting off"
    - Gravity meter transitions from red → orange → yellow → green
    - Counter shows: "Weight Removed: X hours/week"
    - Celebratory particles when milestones hit (10 hrs, 25 hrs, 40 hrs)
4. **Digital Employee Preview**
    - Right panel shows animated bot "executing" workflows
    - Real-time visualization: Tasks flowing through bot
    - Speed comparison: Manual (slow) vs Automated (instant)
    - Success animations: Checkmarks, confetti, progress bars

Automation Templates (Pre-built for quick wins)

1. **"The Sales Associate"**: Lead capture → Enrichment → CRM entry → Follow-up sequence
2. **"The Accountant"**: Invoice sent → Payment tracked → Reminder automated → Books updated
3. **"The Recruiter"**: Resume parsed → Candidate scored → Interview scheduled → Notes compiled
4. **"The Marketer"**: Content scheduled → Published → Analytics tracked → Report generated

Gamification Elements

- **Achievement System**:
  - "First Automation" - Create any workflow
  - "Time Saver" - Remove 10 hours/week
  - "Escape Velocity" - Remove all 50 hours
  - "Power User" - Use all 4 templates
- **Progress Bar**: Visual journey from "Gravity Well" to "Orbit"
- **Leaderboard Hook**: "Join Skool to see how others optimized their workflows"

Completion Trigger

- **Condition**: Remove 40+ hours (80% of total weight)
- **Animation**: Rocket launch sequence (workflow "lifts off")
- **Transition**: "Achieving orbit... witness the transformation"
- **CTA**: "Launch to Mission Control" (proceeds to Scene 3)

Code Structure

```
// Scene2.js - Phaser Scene Class with interact.js integration
class Launchpad extends Phaser.Scene {
constructor() { super({ key: 'Launchpad' }); }

create() {
// Initialize drag-drop system (interact.js)
// Create task palette
// Set up canvas grid
// Initialize digital employee animations
// Load automation templates
}

handleDrop(taskBlock, dropZone) {
// Create automation node
// Update weight meter
// Trigger digital employee animation
// Check for achievement unlocks
}

applyTemplate(templateName) {
// Auto-build workflow from template
// Animate creation process
// Calculate weight removed
}
}
```

SCENE 3: MISSION CONTROL (Results Dashboard)

Visual Design

Theme: Split-screen before/after comparison + client case studies
Perspective: Dashboard view with interactive data viz
Layout: Three panels

- Top: Time-lapse comparison (Manual vs Automated week)
- Middle: Client "satellites" (clickable case studies)
- Bottom: Quiz module + certificate generation

Game Mechanics

1. **Time-Lapse Comparison**
    - Split screen: Left (Manual workflow) vs Right (Automated)
    - Week passes in fast-forward (7-second animation)
    - Manual side: Chaotic, errors, delays, stress indicators
    - Automated side: Smooth, perfect execution, happy indicators
    - Real metrics display: Hours saved, errors prevented, revenue impact
2. **Client Satellites (Interactive Case Studies)**
    - 5 orbiting "satellites" representing real Plinko clients
    - Click satellite → Modal opens with case study
    - Each satellite shows:
        - Industry icon
        - Result headline ("$24K in 3 weeks")
        - Client quote (video thumbnail or text)
        - Specific automations built
        - ROI metrics
3. **Satellite Data** (Pull from Google Sheets via Antigravity integration)

```
const clientResults = [
{
name: "SAP Recruitment Firm",
industry: "Recruiting",
result: "$24K Additional Revenue",
timeframe: "3 weeks",
automation: "Lead enrichment + CRM integration",
quote: "Plinko turned our data chaos into a revenue machine.",
hoursaved: 15
},
{
name: "Events Company",
industry: "Event Management",
result: "$18K Contract Value Recovered",
timeframe: "1 month",
automation: "Invoice tracking + payment reminders",
quote: "We recovered contracts we thought were dead.",
hoursaved: 8
},
{
name: "Real Estate Team",
industry: "Real Estate",
result: "$50K Revenue Growth",
```

timeframe: "90 days",
automation: "Lead follow-up + showing scheduler",
quote: "Our agents focus on selling, not admin.",
hoursaved: 25
},
{
name: "Healthcare Practice",
industry: "Healthcare",
result: "60% Admin Reduction",
timeframe: "2 months",
automation: "Patient intake + appointment reminders",
quote: "Staff can finally focus on patient care.",
hoursaved: 20
},
{
name: "Professional Services",
industry: "Consulting",
result: "40 Hours/Month Freed",
timeframe: "6 weeks",
automation: "Reporting + client communication",
quote: "Like hiring a full-time admin for $500/mo.",
hoursaved: 40
}
];

4. **Learning Quiz (5 Questions)**
    - Multiple choice format
    - Questions test understanding of:
        - What is "Business Antigravity"?
        - When do you achieve "Escape Velocity"?
        - What's a "Digital Employee"?
        - How does performance-based pricing work?
        - Which tasks automate best?
    - Instant feedback: Correct (green) / Incorrect (red with explanation)
    - Must score 4/5 to unlock certificate
5. **Certificate Generation**
    - Canvas API draws personalized certificate
    - Includes: User's name, date, "Certified Antigravity Operator"
    - Plinko Solutions branding + logo
    - Download as PNG
    - Social share buttons (LinkedIn, Twitter)

Completion CTAs

1. **Primary**: "Join the Antigravity Lab" (Skool community link)
    - Button style: Glowing cyan, pulsing animation
    - Copy: "Continue learning with 200+ citizen developers"
2. **Secondary**: "Build My First Digital Employee" (Agency booking)
    - Button style: Solid blue, professional
    - Copy: "Get your first automation live in 14 days"
3. **Tertiary**: "Download the Antigravity Toolkit" (Lead magnet)
    - Button style: Ghost outline

- Copy: "Free templates + ROI calculator"

Code Structure

```
// Scene3.js - Results & Quiz
class MissionControl extends Phaser.Scene {
constructor() { super({ key: 'MissionControl' }); }

create() {
// Initialize time-lapse comparison
// Load client satellite data
// Set up quiz module
// Prepare certificate canvas
}

showCaseStudy(satelliteData) {
// Create modal overlay
// Animate data presentation
// Show video or quote
// Add "Learn More" CTA
}

generateCertificate(userName, quizScore) {
// Use Canvas API to draw cert
// Add user data + timestamp
// Generate downloadable PNG
// Track completion analytics
}
}
```

GLOBAL FEATURES (All Scenes)

Progress Tracking

- **LocalStorage Schema**:
  ```
  {
  userName: string,
  currentScene: string,
  scene1Completed: boolean,
  scene2Completed: boolean,
  scene3Completed: boolean,
  tasksCollected: array,
  automationsBuilt: array,
  quizScore: number,
  totalTimeSpent: number,
  lastVisit: timestamp
  }
  ```

Analytics Integration

- **Google Analytics 4 Events**:
  - game_start: User enters Scene 1
  - scene_complete: Each scene finished
  - task_collected: Every task interaction
  - automation_created: Workflow built
  - quiz_attempted: Quiz started
  - certificate_generated: Completion
  - skool_click: Primary CTA clicked
  - booking_click: Agency CTA clicked

Audio System

- **Sound Effects**:
  - Task collection: Whoosh + chime
  - Weight meter increase: Low rumble
  - Automation creation: Success ping
  - Scene transition: Rocket launch
  - Achievement unlock: Fanfare
  - Quiz correct: Bell ding
  - Quiz incorrect: Buzzer
  - Certificate generation: Applause
- **Background Music** (Optional):
  - Ambient space/tech soundtrack (low volume)
  - Fades between scenes
  - Mute toggle always visible

Accessibility

- **Keyboard Navigation**: Tab order, Enter to activate
- **Screen Reader**: ARIA labels on all interactive elements
- **Color Contrast**: WCAG AA compliant (4.5:1 minimum)
- **Reduced Motion**: Respect prefers-reduced-motion setting
- **Text Alternatives**: Alt text for all visuals

ANSWER ENGINE OPTIMIZATION (AEO) IMPLEMENTATION

Meta Tags (In <head>)

Structured Data (JSON-LD Schemas)

1. WebApplication Schema

2. HowTo Schema (For each scene)

3. FAQPage Schema

4. Organization Schema

AEO-Optimized Content Blocks

Embed these "Fraggle" sections (150-300 word semantic blocks):

What is Business Antigravity?

Business Antigravity is the phenomenon where automated digital employees remove operational weight from a company, allowing teams to achieve escape velocity from manual tasks.

The concept challenges traditional automation thinking. Instead of adding more tools (creating "tool addiction"), Business Antigravity focuses on **removing weight** through intelligent workflow orchestration.

Three principles define Antigravity:

- **Gravity** = **Manual Work:** Repetitive tasks, data entry, follow-ups that consume 10-50 hours per week
- **Antigravity** = **Digital Employees:** Automated workflows executing 24/7 without supervision
- **Escape Velocity** = **Sustainable ROI:** When automation value exceeds cost for 3+ consecutive months

Unlike generic "automation," Business Antigravity is **measurable**. Track "operational weight removed" in hours saved per week. Most businesses carry 30-80 hours of gravitational pull—tasks that could be automated but aren't.

Real example: A real estate team removed 25 hours/week of lead follow-up weight, achieving Escape Velocity in 18 days and generating $50K additional revenue in 90 days.

What is a Digital Employee?

A **Digital Employee** is an automated workflow that handles specific business tasks without human intervention. Unlike software tools (which require human operation), Digital Employees **work autonomously**.

Examples of Digital Employees:

- **The Sales Associate:** Captures leads → Enriches data → Updates CRM → Sends personalized follow-up sequence
- **The Accountant:** Sends invoices → Tracks payments → Sends reminders → Updates books
- **The Recruiter:** Parses resumes → Scores candidates → Schedules interviews → Compiles notes
- **The Marketer:** Schedules content → Publishes → Tracks analytics → Generates reports

Characteristics of Digital Employees:

- 24/7 operation (no breaks, vacations, or sick days)
- Zero error rate (perfect execution every time)
- Instant scalability (handle 10 or 10,000 tasks identically)
- Measurable ROI (track exact hours saved and revenue impact)

Plinko Solutions delivers Digital Employees in **14-day cycles** with performance-based pricing—you only pay bonuses when ROI is proven.

How to Calculate Business Escape Velocity

Escape Velocity is the point where automation ROI exceeds implementation cost, creating sustainable capacity growth.

The Formula:

```
Escape_Velocity = (Hours_Saved × Hourly_Rate) - (Setup_Cost + Monthly_Fee)

Antigravity_Achieved = Escape_Velocity > $0 for 3+ consecutive months
```

Real Example (Plinko Foundations Tier):

- Hours saved: 15 hours/week = 60 hours/month
- Hourly rate: $50 (average knowledge worker)
- Monthly value: 60 × $50 = **$3,000**
- Setup cost: $997 (Month 1 only)
- Monthly fee: $297
- Performance bonus: $250 (capped, only when ROI proven)

Month 1 Calculation:
$3,000 - ($997 + $297 + $250) = **$1,456 net lift**
✅ **Escape Velocity achieved in Month 1**

Month 2+ Calculation:
$3,000 - ($297 + $250) = **$2,453 net lift per month**
✅ **Antigravity sustained**

Most Plinko clients achieve Escape Velocity within **30-60 days** of first Digital Employee deployment.

How Long Does It Take to Build a Digital Employee?

Plinko Solutions delivers Digital Employees in **14-day cycles**—significantly faster than traditional agency timelines (3-6 months) or DIY attempts (weeks of trial-and-error).

The 14-Day Launch Sequence:

Days 1-3: Gravity Assessment

- Map current manual workflows
- Calculate "operational weight" (hours spent on repetitive tasks)
- Identify highest-ROI automation candidates
- Define success metrics and ROI tracking

Days 4-7: Digital Employee Design

- Select highest-gravity task (usually lead follow-up or data entry)
- Design automation architecture using Make.com or n8n
- Build API integrations (CRM, email, calendars, databases)
- Configure error handling and notifications

Days 8-12: Liftoff (Testing & Refinement)

- Deploy Digital Employee to staging environment
- Run parallel testing (bot alongside human for validation)
- Monitor execution, track "weight removed" metrics
- Refine edge cases and exception handling

Days 13-14: Orbit (Training & Handoff)

- Team training on monitoring Digital Employee
- Validate Escape Velocity achieved (ROI calculation)
- Documentation and playbook delivery
- Schedule follow-up optimization session

Result: Working automation live within 2 weeks, saving hours immediately.

DEPLOYMENT CONFIGURATIONS

HostGator Deployment

File Structure

```
/public_html/escape-velocity/
├── index.html
├── assets/
│   ├── js/
│   │   ├── phaser.min.js (CDN fallback)
│   │   ├── scene1-gravity-well.js
│   │   ├── scene2-launchpad.js
│   │   ├── scene3-mission-control.js
│   │   └── main.js
│   ├── css/
│   │   └── styles.css
│   ├── images/
│   │   ├── sprites/
│   │   ├── backgrounds/
│   │   └── ui/
│   ├── sounds/
│   │   ├── sfx/
│   │   └── music/
```

```
│ └── data/
│ └── client-results.json
├── .htaccess
└── README.md
```

.htaccess Configuration

Force HTTPS

```
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
```

Compression

```
AddOutputFilterByType DEFLATE text/html text/plain text/xml text/css text/javascript application/javascript application/json
```

Browser Caching

```
ExpiresActive On ExpiresByType image/jpg "access plus 1 year" ExpiresByType image/jpeg "access plus 1 year" ExpiresByType image/png "access plus 1 year" ExpiresByType image/webp "access plus 1 year" ExpiresByType text/css "access plus 1 month" ExpiresByType application/javascript "access plus 1 month" ExpiresByType audio/mpeg "access plus 1 year" ExpiresByType audio/ogg "access plus 1 year"
```

Security Headers

```
Header set X-Content-Type-Options "nosniff" Header set X-Frame-Options "SAMEORIGIN" Header set X-XSS-Protection "1; mode=block" Header set Referrer-Policy "strict-origin-when-cross-origin"
```

Error Pages

```
ErrorDocument 404 /escape-velocity/404.html
ErrorDocument 500 /escape-velocity/500.html
```

Upload Instructions

1. Connect via FTP (FileZilla) or cPanel File Manager
2. Navigate to `/public_html/`
3. Create `/escape-velocity/` directory
4. Upload all files maintaining structure
5. Set permissions: 755 for directories, 644 for files
6. Test at `https://plinkosolutions.com/escape-velocity/`

Vercel Deployment

File Structure

```
/
├── index.html
├── public/
│   ├── js/
│   ├── css/
│   ├── images/
│   ├── sounds/
│   └── data/
├── vercel.json
├── package.json (optional, for build step)
└── README.md
```

vercel.json Configuration

```
{
"version": 2,
"builds": [
{
"src": "index.html",
"use": "@vercel/static"
}
],
"routes": [
{
"src": "/(.)",
"dest": "/$1"
}
],
"headers": [
{
"source": "/(.)",
"headers": [
{
"key": "X-Content-Type-Options",
"value": "nosniff"
},
{
"key": "X-Frame-Options",
"value": "SAMEORIGIN"
},
{
"key": "X-XSS-Protection",
"value": "1; mode=block"
},
{
"key": "Referrer-Policy",
"value": "strict-origin-when-cross-origin"
}
```

]
},
{
"source": "/(.)\.(jpg|jpeg|png|webp|gif|svg|ico)",
"headers": [
{
"key": "Cache-Control",
"value": "public, max-age=31536000, immutable"
}
]
},
{
"source": "/(.)\.(css|js)",
"headers": [
{
"key": "Cache-Control",
"value": "public, max-age=2592000, stale-while-revalidate"
}
]
}
]
}

Deploy Methods

Method 1: GitHub Integration (Recommended)

1. Push code to GitHub repository
2. Connect Vercel to GitHub
3. Import repository
4. Auto-deploy on every push
5. Preview URLs for each commit

Method 2: Vercel CLI

```
npm install -g vercel
cd escape-velocity
vercel login
vercel --prod
```

Method 3: Drag-and-Drop

1. Visit vercel.com/new
2. Drag project folder
3. Configure domain (plinkosolutions.com/escape-velocity)
4. Deploy

TESTING & VALIDATION CHECKLIST

Pre-Launch Testing

Functional Testing

- [ ] All 10 task cards collectable in Scene 1
- [ ] Gravity meter updates correctly
- [ ] Scene transitions work smoothly
- [ ] Drag-and-drop functions in Scene 2
- [ ] Automation templates apply correctly
- [ ] Weight removal calculations accurate
- [ ] Client satellites clickable in Scene 3
- [ ] Quiz questions display correctly
- [ ] Certificate generates with user's name
- [ ] LocalStorage saves/loads progress
- [ ] Analytics events fire properly

Cross-Browser Testing

- [ ] Chrome (latest)
- [ ] Firefox (latest)
- [ ] Safari (macOS + iOS)
- [ ] Edge (latest)
- [ ] Chrome Mobile (Android)

Mobile Responsiveness

- [ ] Touch controls work (no keyboard dependency)
- [ ] UI elements scale properly
- [ ] Text readable without zoom
- [ ] Buttons/interactive elements touchable (44px minimum)
- [ ] Landscape and portrait modes functional

Performance Testing

- [ ] Initial load < 3 seconds (3G network)
- [ ] No memory leaks after 15 min gameplay
- [ ] Audio doesn't cause lag
- [ ] Animations smooth (60 FPS)
- [ ] LocalStorage doesn't exceed 5MB

AEO Validation

- [ ] All schema markup validates (schema.org validator)
- [ ] Meta tags render correctly (Facebook Debugger, Twitter Card Validator)
- [ ] Structured data appears in Google Search Console
- [ ] FAQ content indexable (robots.txt check)
- [ ] Canonical URLs correct
- [ ] Open Graph images display properly

Accessibility Audit

- [ ] Keyboard navigation complete (no mouse-only interactions)
- [ ] Screen reader announces all elements (NVDA/VoiceOver test)
- [ ] Color contrast meets WCAG AA (4.5:1 minimum)
- [ ] Focus indicators visible
- [ ] Alt text for all images
- [ ] ARIA labels on interactive elements

ANALYTICS & SUCCESS TRACKING

GA4 Event Configuration

```
// Track game events
function trackEvent(eventName, parameters) {
gtag('event', eventName, parameters);
}

// Example implementations:
trackEvent('game_start', {
timestamp: Date.now(),
returning_user: localStorage.getItem('userName') ? true : false
});

trackEvent('scene_complete', {
scene_name: 'Gravity Well',
tasks_collected: 10,
time_spent: 180 // seconds
});

trackEvent('automation_created', {
automation_type: 'Sales Associate',
weight_removed: 8 // hours/week
});

trackEvent('quiz_attempted', {
score: 4,
total_questions: 5,
passed: true
});

trackEvent('certificate_generated', {
user_name: userName,
completion_time: totalTimeSpent
});

trackEvent('cta_clicked', {
cta_type: 'skool_primary',
scene_location: 'Mission Control'
});
```

Success Metrics Dashboard

Track weekly:

- Unique visitors
- Completion rate (% reaching Scene 3)
- Average time spent
- Skool CTA click-through rate
- Booking CTA click-through rate
- Certificate downloads
- Social shares

Target benchmarks (Month 1):

- 500+ unique visitors
- 35%+ completion rate
- 12+ minute avg time spent
- 20%+ Skool CTA clicks
- 5%+ booking clicks
- 100+ certificates generated

AGENT EXECUTION INSTRUCTIONS

Deploy Multiple Agents for Parallel Development

Agent 1: Scene 1 - Gravity Well
Build Scene 1 (Gravity Well) for Escape Velocity game:

- 2D side-scrolling platformer with Phaser.js
- 10 collectable task cards
- Gravity meter UI (top right)
- Character sprite with arrow key movement
- LocalStorage integration for progress
- Sound effects for collection
- Transition to Scene 2 on completion
  Test in browser and capture video walkthrough.

Agent 2: Scene 2 - Launchpad
Build Scene 2 (Launchpad) for Escape Velocity game:

- Drag-and-drop automation builder (interact.js)
- 4 pre-built templates (Sales, Accountant, Recruiter, Marketer)
- Weight removal animations
- Digital employee preview panel
- Achievement system
- Transition to Scene 3 when 40+ hours removed
  Test all drag-drop interactions and capture screenshots.

Agent 3: Scene 3 - Mission Control
Build Scene 3 (Mission Control) for Escape Velocity game:

- Time-lapse comparison animation (Manual vs Automated)

- 5 client satellites with case study modals
- Quiz module (5 questions, multiple choice)
- Certificate generation (Canvas API)
- Skool and booking CTAs
- Social share buttons
  Test quiz logic, certificate download, and CTA tracking.

Agent 4: AEO Implementation
Implement all Answer Engine Optimization for Escape Velocity:

- Add all schema markup (WebApplication, HowTo, FAQPage, Organization)
- Insert AEO content blocks (Business Antigravity definition, Digital Employee explanation, Escape Velocity formula, Implementation timeline)
- Configure meta tags (Open Graph, Twitter Cards)
- Validate with schema.org validator
  Generate report of all structured data added.

Agent 5: Deployment Configuration
Set up deployment configs for HostGator and Vercel:

- Create .htaccess (HostGator version)
- Create vercel.json (Vercel version)
- Organize file structure for both platforms
- Add README with upload instructions
- Test deployment on Vercel preview
  Document deployment process with screenshots.

Manager Agent Orchestration

Coordinate 5 parallel agents building Escape Velocity game:

Agents 1-3: Build Scenes 1-3 simultaneously
Agent 4: Implement AEO (can start immediately)
Agent 5: Set up deployment configs (can start immediately)

Once Agents 1-3 complete:

- Integrate all 3 scenes into single index.html
- Test scene transitions
- Validate LocalStorage works across scenes
- Merge with Agent 4's AEO implementation
- Apply Agent 5's deployment configs

Final validation:

- Run full playthrough (Scene 1 → 2 → 3 → Certificate)
- Check analytics events fire
- Validate all schemas
- Test on mobile
- Deploy to Vercel preview
- Generate final QA report

Estimated completion: 4-6 hours with parallel execution.

POST-LAUNCH OPTIMIZATION

Week 1: Gather Data

- Monitor completion rates by scene
- Track drop-off points
- Analyze time spent per scene
- Review CTA click patterns
- Check mobile vs desktop performance

Week 2: Iterate Based on Feedback

- A/B test CTA copy
- Adjust difficulty if completion rate < 30%
- Optimize loading speed if > 3 seconds
- Fix any reported bugs
- Add user-requested features

Week 3: AEO Monitoring

- Check Google Search Console for indexing
- Track Featured Snippet appearances
- Monitor ChatGPT/Perplexity citations (Relixir or manual)
- Measure organic traffic growth
- Analyze search queries driving traffic

Week 4: Conversion Optimization

- Review Skool sign-up rate from game completers
- Analyze booking conversion path
- Test different certificate designs
- Experiment with social share copy
- Add testimonials from early players

EXPECTED OUTCOMES

Performance Targets (90 Days)

Traffic:

- 2,000+ unique visitors (organic + referral + social)
- 40%+ from AEO (answer engine citations)
- 25%+ from social shares (LinkedIn, Twitter)

Engagement:

- 40%+ completion rate (reach Scene 3)
- 15+ minute avg time spent
- 3.5+ scenes per session (revisits)

Conversion:

- 20%+ Skool CTA clicks (400+ community joins)
- 5%+ booking CTA clicks (100+ strategy calls booked)
- 30%+ certificate downloads (600+ generated)

AEO Performance:

- 1,500+ AI citations across ChatGPT, Perplexity, Gemini
- Featured Snippets for "Business Antigravity", "Digital Employee", "Escape Velocity"
- Ranking for 20+ long-tail automation queries

Business Impact:

- 50+ new Skool community members from game
- 10+ new client bookings directly attributed
- 200+ social shares driving brand awareness
- Position as thought leader in "Business Antigravity" concept

---

FINAL DELIVERABLES

Upon agent completion, you will have:

1. ✅ **Fully functional 3-scene interactive game**
2. ✅ **Complete AEO implementation** (all schemas + content blocks)
3. ✅ **Deployment-ready code** (HostGator + Vercel configs)
4. ✅ **Analytics tracking** (GA4 events configured)
5. ✅ **Mobile-responsive design** (works on all devices)
6. ✅ **Accessibility compliant** (WCAG AA standard)
7. ✅ **Documentation** (README, deployment guide, maintenance)
8. ✅ **Testing report** (browser compatibility, performance)

Deploy immediately to plinkosolutions.com/escape-velocity/ and begin driving traffic.

END OF COMPLETE ANTIGRAVITY PROMPT

PART 2: EXECUTION WORKFLOW

Step 1: Set Up Antigravity Workspace

1. Install Antigravity (free preview at antigravity.google)
2. Create new project: "Escape Velocity - Plinko Solutions"
3. Enable browser extension (Chrome/Edge required)
4. Connect to Google account (for Sheets integration)

Step 2: Deploy Agents

1. Copy complete prompt above into Antigravity
2. Select "Agent Decides" mode (lets AI determine task breakdown)
3. Deploy 5 parallel agents as described in Manager orchestration
4. Monitor progress in Manager View

Step 3: Review & Iterate

1. Each agent produces artifacts (task plans, code, browser recordings)
2. Comment directly on artifacts for adjustments
3. Agents iterate without stopping workflow
4. Approve final implementations when satisfied

Step 4: Integration Testing

1. Agents merge all scenes into single index.html
2. Test full playthrough in Antigravity browser
3. Validate AEO schemas with schema.org validator
4. Run mobile responsiveness checks
5. Verify analytics events fire correctly

Step 5: Deploy to Production

For HostGator:

1. Download complete project from Antigravity
2. Connect via FTP or cPanel File Manager
3. Upload to /public_html/escape-velocity/
4. Test at plinkosolutions.com/escape-velocity/

For Vercel:

1. Push code to GitHub from Antigravity
2. Connect Vercel to repository
3. Configure custom domain
4. Auto-deploy on every commit

Step 6: Monitor & Optimize

1. Track metrics in GA4 dashboard
2. Monitor AEO citations with Relixir or manual checks
3. Gather user feedback via Skool community
4. A/B test CTAs and messaging
5. Iterate weekly based on data

Expected Timeline

Day	Activity	Agent Work
1	Set up Antigravity, deploy agents	Agents build Scenes 1-3 in parallel
2	Review Scene implementations, provide feedback	Agents iterate based on comments
3	AEO implementation, deployment configs	Agents 4-5 complete schemas and configs
4	Integration testing, mobile QA	Final validation and bug fixes
5	Deploy to production, launch announcement	Live on plinkosolutions.com

Total: 5 days from prompt to production-ready interactive experience.

Success Criteria

✓ Game loads in < 3 seconds
✓ All 3 scenes functional and tested
✓ LocalStorage saves progress correctly
✓ Analytics events fire on all interactions
✓ AEO schemas validate 100%
✓ Mobile-responsive on all devices
✓ Accessibility audit passes WCAG AA
✓ Deployed to both HostGator and Vercel
✓ Documentation complete for maintenance
✓ 40%+ completion rate within first week

Post-Launch Promotion

1. **LinkedIn Post**: "I built an interactive game that teaches AI automation..."
2. **Skool Announcement**: "New free resource: Escape Velocity game"
3. **Email to existing list**: "Play through your automation journey"
4. **Reddit/HN**: "I made Google Interland for business automation"
5. **YouTube Walkthrough**: "Behind-the-scenes building this in 5 days"

This prompt is production-ready. Copy into Antigravity workspace and execute immediately.