VAULT, SLIDE, MANTLE

BUILDING BRINK'S SMART SYSTEM

Arne Olav Hallingstad
Lead Gameplay Programmer
Splash Damage

SPLASH DAMAGE

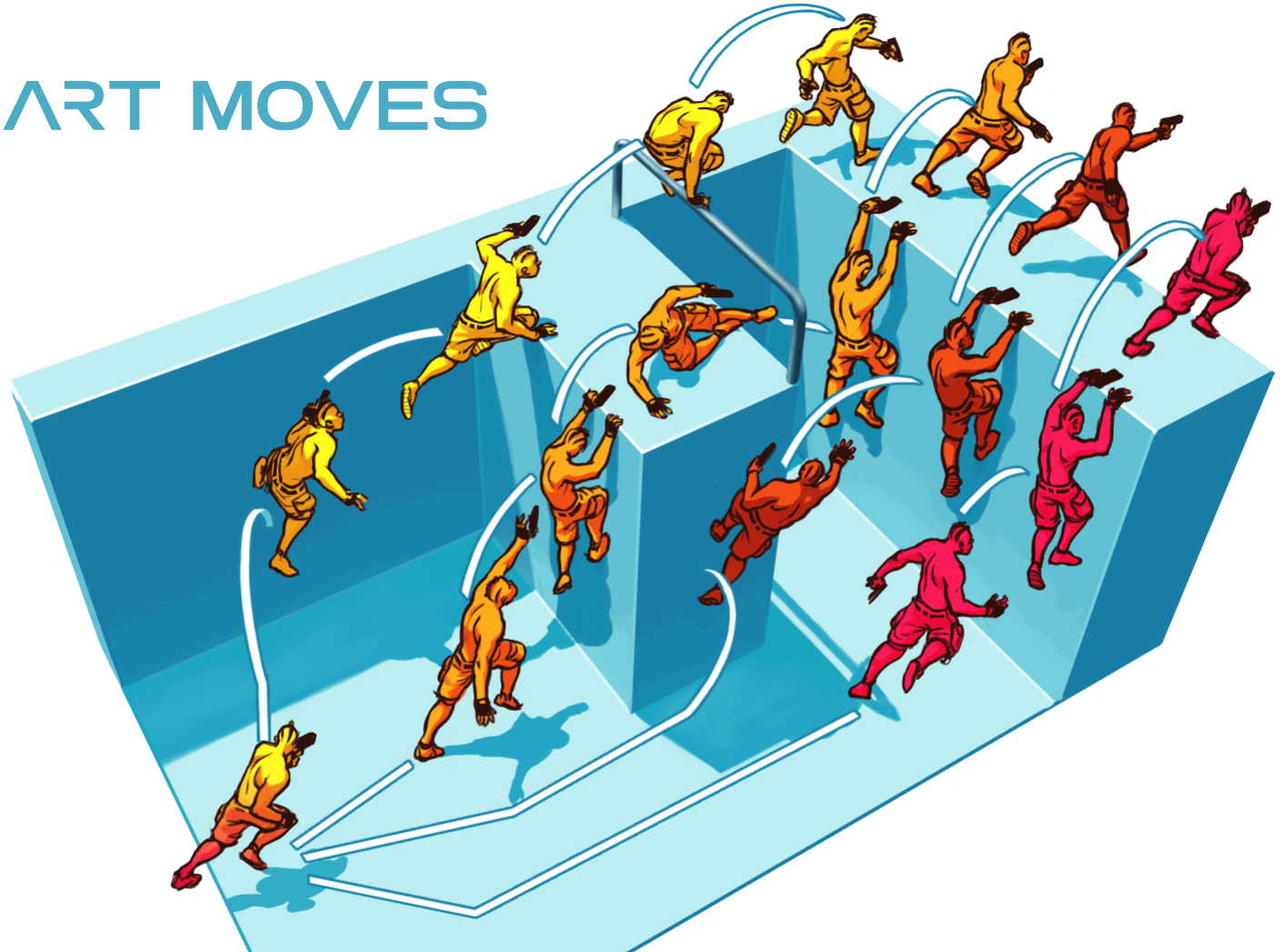
- Multiplayer team & objective based FPS games
- Relatively small team
- Evolve multiplayer shooters

GOALS

- Improve player movement
- Be consistent
- Be accessible
- Support different body types
- Shouldn’t require extra LD work
- Must be usable by AI

JACK MONAHAN
GAUSSWERKS.COM

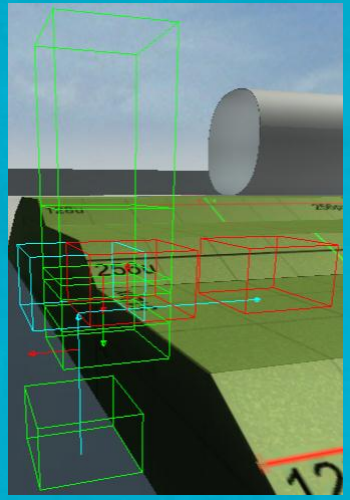
SMART MOVES

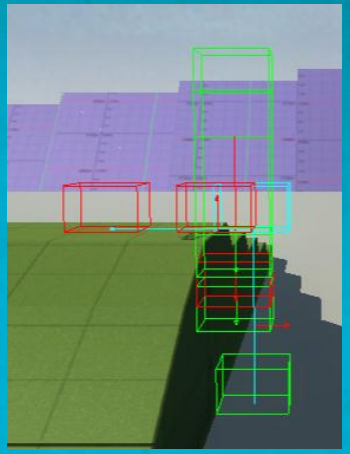


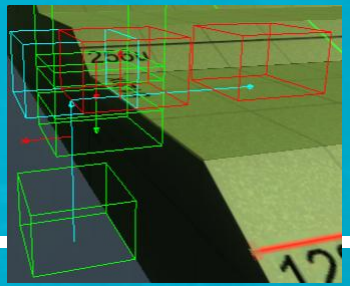
WHAT WE'LL COVER

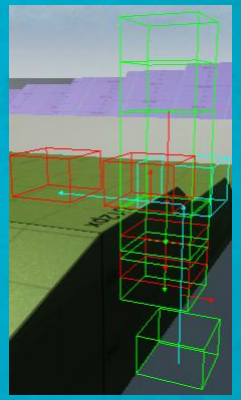
- Prototype
- Precomputation
- Runtime Detection
- Runtime Execution
- Lessons Learned

PROTOTYPE







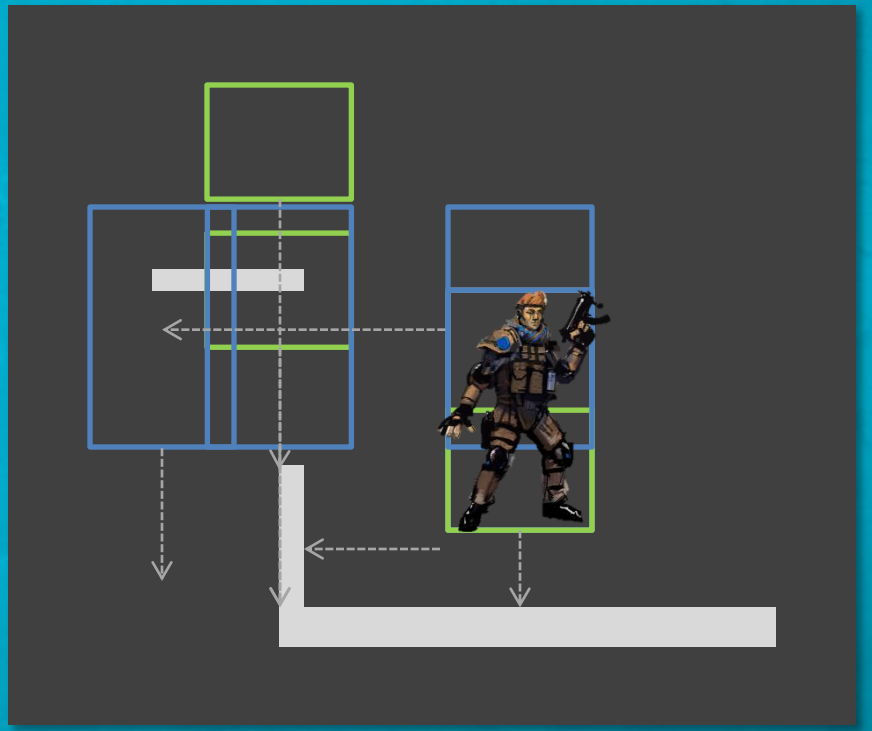


PROTOTYPE

- Prove viability of SMART movement
  - Multiplayer game
  - Impact on level design and gameplay
- Prototyped using run-time collision traces
- Refined over 6 months

LEDGE DETECTION + VAULTING

- Find ground
- Find wall
- Find low edge/high edge
- Trace clip to ledge height
- Trace clip over ledge
- Trace clip down
- Trace down on ledge



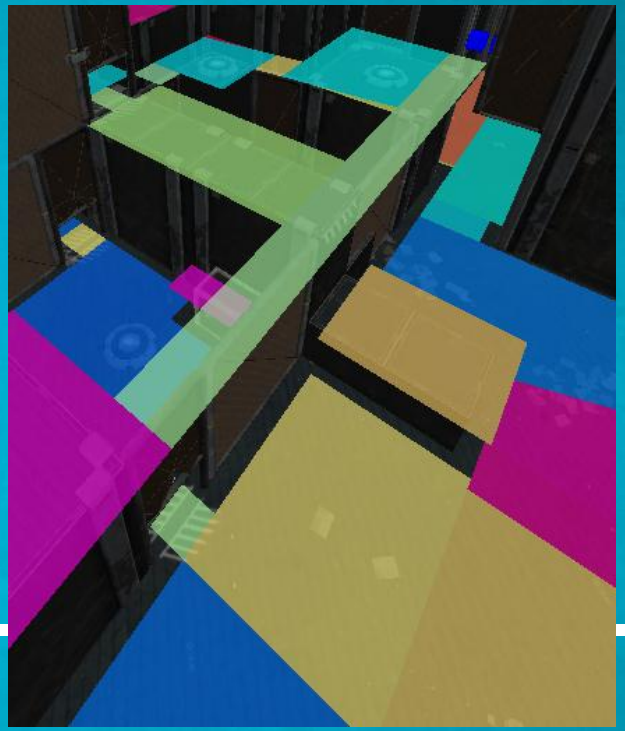
SUCCESSES

- Easy to implement
- No LD placed hint objects
- Works on any map
- Standardized map metrics

ISSUES

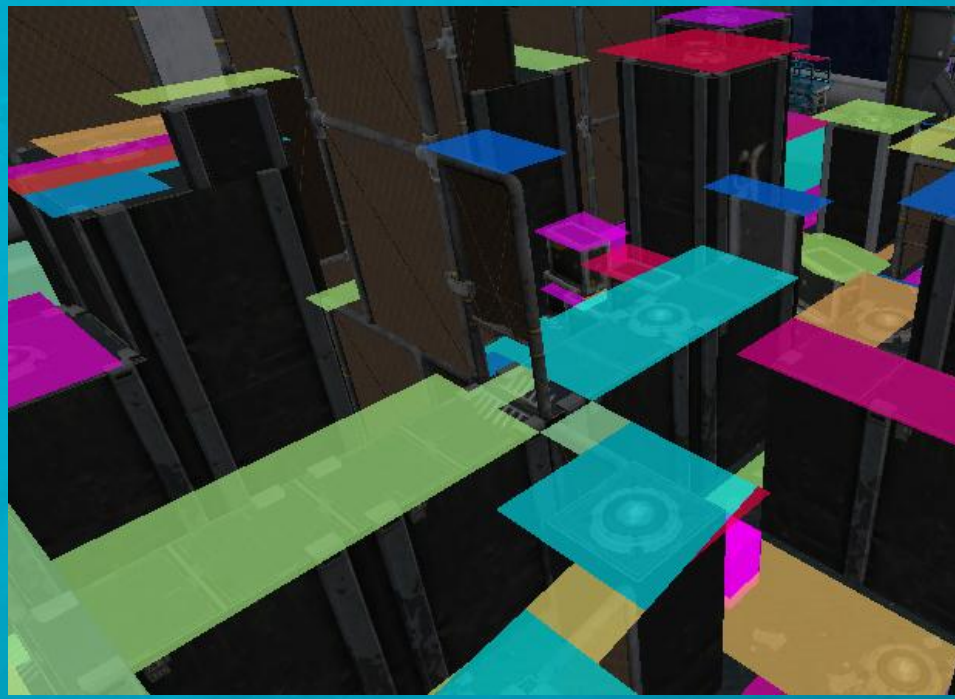
- At least 1 trace every frame
- Worst case 8 traces per player

PRECOMPUTATION



NAV MESH SYSTEM

- Used for AI path-finding
- Map-compile step
- Areas connected by reachabilities
- Potential use for SMART?

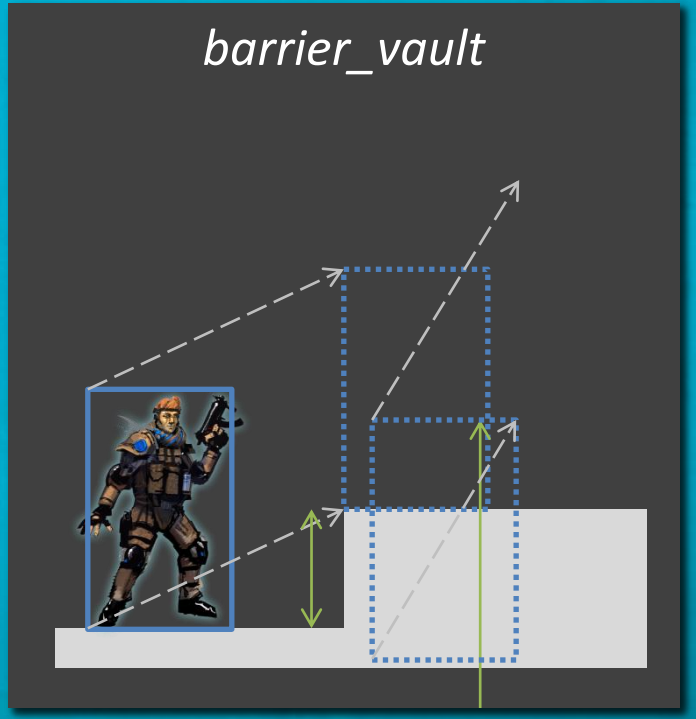


REACHABILITIES

- Get all edges between two areas
- Edges overlap vertically we may create a reachability
- Stores edge segment
- *travel_flags*

REACHABILITY TYPES

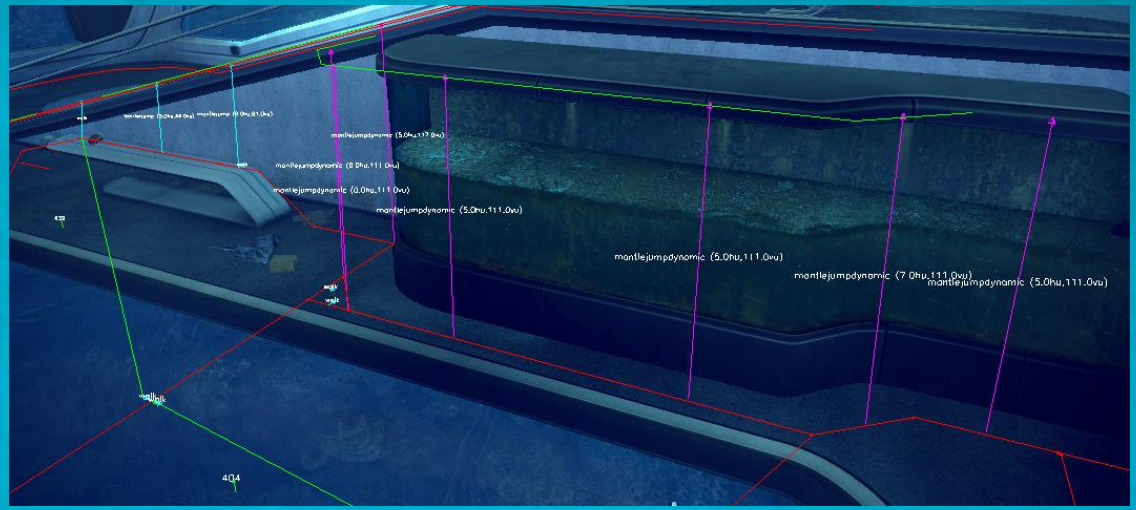
- *barrier_vault*
- *barrier_mantle*
- AI pathfinding
- Used by players & bots
- *barrier_dynamic*
  - Used by players only
  - Vault/Mantle move decided at runtime
  - Explosion in number of reachabilities



SLIDE

- Areas marked low ceiling
- Bots & players required to crouch
- Players can auto slide into these areas

RUNTIME DETECTION



PLAYER PHYSICS LOOP

- Step 1: Is player on ground?
- Step 2: Query player body type for available movement modes
- Step 3: Detect high moves (vault, mantle, wall hop)
- Step 4: Detect low moves (slides)
- Step 5: Choose active move
- Step 6: Update player state machine

STEP 3: DETECT HIGH MOVE

- 3.1: Player checks
- 3.2: Nav mesh query
- 3.3: Evaluate high moves

STEP 3.1: PLAYER CHECKS

- Cannot be in active state
  - Vaulting
  - Mantling
  - Sliding
  - Iron-sighting
  - Knocked down

STEP 3.2: NAV MESH QUERY

- Search bounds 6x player b-box width & 2.5x player height
- Areas = GetBoundsAreas(searchBounds)
  - Areas in BSP-tree
- For each Area
  - For each Reachability
    - barrier_vault, barrier_mantle or barrier_dynamic
    - Append to list

STEP 3.3: EVALUATE HIGH MOVES

- Iterate all reachabilities
  - Player must look at the ledge
  - Distance within 2.5x player b-box width
- Vault: ledge height is 0.4x-0.8x player height
- Mantle: ledge height is 0.8x-1.4x player height
- Auto wall hop: Ledge height is mantle height + player's jump height

STEP 3.3: EVALUATE HIGH MOVES

- Reachability list
- Exclude wall hop if vault/mantle in list
- Mutual exclusion
  - Allow mantle if within 1.5x player b-box width
  - Otherwise: Allow vault
- Sort potential moves by closest ledge

PLAYER PHYSICS LOOP

- Step 1: Is player on ground?
- Step 2: Query player body type for available movement modes
- Step 3: Detect high moves (vault, mantle, wall hop)
- Step 4: Detect low moves (slide)
- Step 5: Choose active move
- Step 6: Update player for delta time

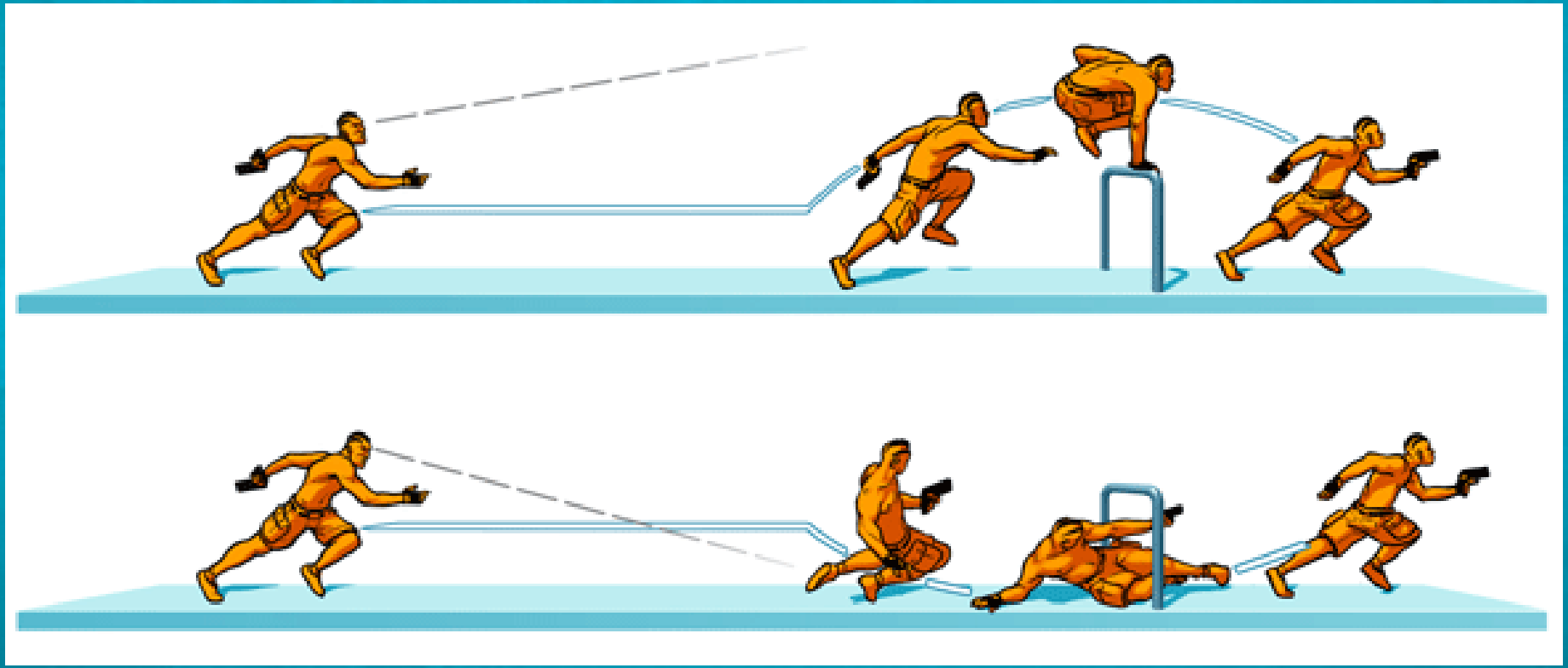
STEP 4: DETECT SLIDE

- 4.1: Player checks
- 4.2: Nav mesh query
- 4.3: Evaluate low moves

STEP 4.3: EVALUATE LOW MOVES

- Iterate all areas
- Area within height of 0.4x player height
- Area marked as low ceiling
- Auto crouch: distance < small number
- Auto slide: distance within 1.5x player b-box width
- Mutual exclusion
  - Allow auto slide if sprint held
  - Otherwise: Allow auto crouch

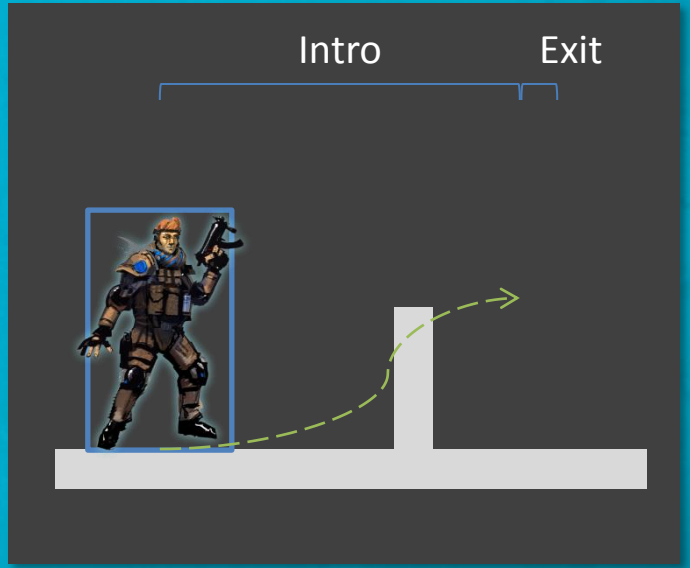
STEP 5: CHOOSE ACTIVE MOVE



RUNTIME EXECUTION

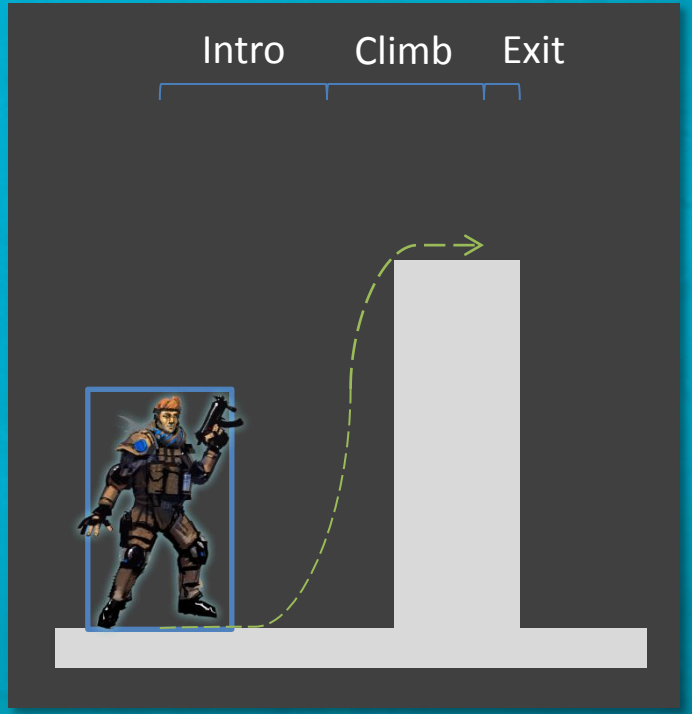
VAULT PHYSICS STATES

- Intro and exit states
- Intro
  - Duration: Distance and player velocity
  - Spline from player position to ledge
- Exit
  - Calculates momentum
  - Calculates direction
  - Clear momentum if drop too high (trace)



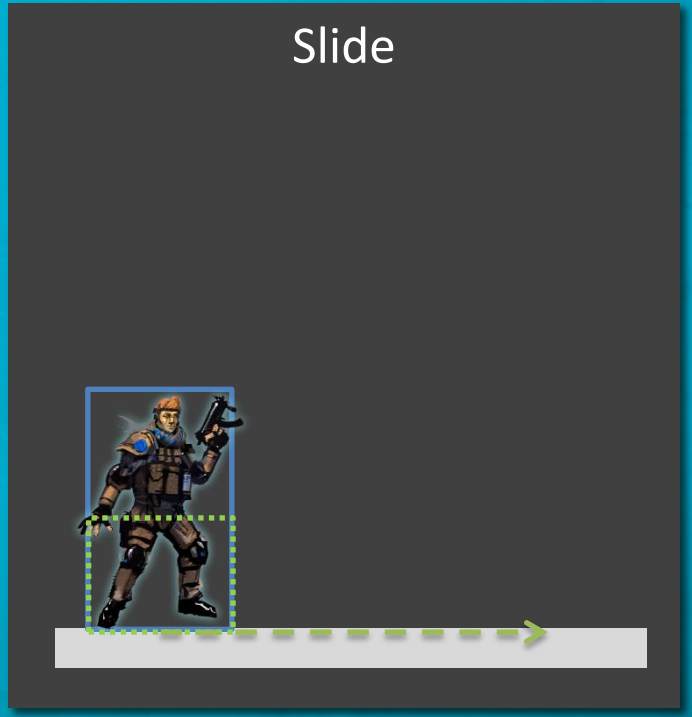
MANTLE PHYSICS STATES

- Intro, climb and exit states
- Spline calculation same as vault
- Intro
  - Duration calculation same as vault
  - Push player to correct position
- Climb
  - Duration scaled up during climb
  - Pulls the player on top of ledge
- Exit - Clear momentum



SLIDE + WALL HOP PHYSICS STATES

- Sliding
  - Force crouch
  - Constant velocity in direction of travel for a second
- Wall Hop
  - Single frame state
  - Only light body type may wall hop



THIRD PERSON ANIMATIONS

- Animations driven by physics state
- Slide - Play slide animation
- Vault - Intro plays vault animation
- Mantle
  - Intro plays grab animation
  - Climb plays mantle animation as root motion

THIRD PERSON ANIMATIONS

- Animations driven by physics state
- Slide - Play slide animation
- Vault - Intro plays vault animation
- Mantle
  - Intro plays grab animation
  - Climb plays mantle animation as root motion

CONCLUSIONS

- More fluid movement
- SMART Button
- Generated during map-compile step
- Free flow restricted by body types

LESSONS LEARNED

splash damage

- Prototyping allows quick iterations
- Systems can successfully be used beyond their original intention
- Could give client authority over physics
- Consolidating physics states saves network bandwidth

ARNE OLAV HALLINGSTAD

WWW.SPLASHDAMAGE.COM
AO@SPLASHDAMAGE.COM

@SPLASHDAMAGE
@ARNEOLAVHAL

QUESTIONS?