為模擬 AI 在 Super Mario 中尋找最短通關策略的核心挑戰，我首先建立了一個一維跑關模型(見model.nypb第一部分)。環境共有 10 個位置（0–9），0 為起點、9 為終點。格 3 為坑洞、格 5 有敵人。每一幀可採取兩種動作：run 或 jump。若以 run 落在陷阱或敵人格上則死亡（回報 −100），以 jump 則可安全越過。每步行動消耗 1 點時間（回報 −1），抵達終點給予 +100 獎勵。這是個確定性 MDP（Markov Decision Process），可完整定義狀態轉移與報酬函數。

求解方法是使用Value Iteration求得最優策略。其核心方程為$V_{k+1}(s) = \max[r(s,a) + \gamma V_k(s')]$，在此$\gamma = 1$，因任務有限步即結束。反覆更新直到所有狀態的價值收斂。接著依每個狀態選取使$r + V(s')$最大的動作作為最優策略。

最後模擬結果顯示，需在tile2與tile4選擇"jump"決策，其餘使用"run"，AI共用9步（幀）安全抵達終點，總報酬 92 分。

然而若只用上述簡化模型，極難反映出實際的情況，主要原因是此模型排除了glitch(漏洞)的影響，因此我又設計了另一個模型(見model.nypb第二部分)，以Bullet Bill漏洞現象為例子，讓AI系統探索這類「非預期但可重現」的遊戲漏洞。

在該模型中，AI 以隨機搜尋（Random Explorer）模擬探索過程。探索器會維護一個「archive」用以儲存已發現的狀態，並從中隨機挑選一個儲存點繼續嘗試新動作。若發現新的記憶體狀態，則將其加入 archive。

當環境偵測到warp_byte改變時，即代表觸發了Bullet Bill，系統會將該動作序列記錄為成功範例。這種「回溯式探索」能逐步擴張已知空間，並利用保存點（savestate）重複嘗試新的行為組合，模擬強化學習中的「探索（exploration）」階段。

該模型的核心使用了強化學習，本質是一個馬可夫決策過程，每個動作都會改變環境狀態與回饋獎勵。AI 需在未知環境中透過試錯學習最有效率的策略，最終目標是觸發Bullet Bill（最大化最終獎勵）。這正是強化學習的典型架構。在本模型中，State為位置、子像素、RAM。Action為控制按鍵輸入。Reward為每幀 −1 表示時間成本，觸發Bullet Bill或到達終點則給予高額正獎勵。

在這個模型中，AI 會找到Bullet Bill的唯一情況是：它剛好在一個有特殊物件的frame、角色位置正好重疊該物件、子像素位置 = 0.5、當下動作是「右移＋跳躍」。觸發條件相當嚴苛。

最終模擬的結果為ai於19071次迭代才找到glitch，表示平均要嘗試約20000次才會遇到一次 frame-perfect 的碰撞。每回合最多2000幀，約4千萬幀的嘗試中只有一次成功。換算起來成功率約為0.0000025%。在這個簡化模型裡，Bullet Bill屬於極低機率罕見事件。

總結來說，此模型展示 AI 如何在複雜系統中自我發現非預期現象的縮影，卻也凸顯出AI 在實際尋找最優解時面臨的核心困難，在延遲回饋且搜尋空間指數爆炸的情況下，在未來必須有更好的算法，才能更有效率地搜尋解答。