

POKROČILÝ SYSTÉM SOUBOJE V RPG HŘE

Jakub Kraus

SEKCE – MECHATRONIKA

Fakulta mechatroniky, informatiky a mezioborových studií, 3. ročník
Bakalářský studijní program – INFORMAČNÍ TECHNOLOGIE

Abstrakt: Práce představuje návrh a implementaci pokročilého soubojového systému pro real-time arénové RPG. Řešený problém spočívá ve vytvoření souboje, který je pro hráče čitelný, ale současně vyžaduje reakci na různé role a vybavení nepřátel. Navržený systém kombinuje sdílený model pravidel pro hráče a nepřátele, datově řízené vybavení, Utility AI, Behaviour Tree, Blackboard a Combat Director. Vybavení ovlivňuje nejen poškození a obranu, ale také dostupné akce, optimální vzdálenost, poise a přiřazení skupinové role. Implementace vznikla v enginu Godot 4.5 s využitím GDScriptu a pluginu Beehave. Výsledkem je funkční arénový prototyp se systémem kol, obchodem, generováním vybavení nepřátel a kvalitativním uživatelským testováním. Hlavním přínosem je propojení soubojových mechanik, vybavení a koordinovaného skupinového chování nepřátel do jednoho hratelného technického dema.

Klíčová slova: RPG hra, soubojový systém, Godot, Utility AI, Behaviour Tree, Blackboard, Combat Director, umělá inteligence nepřátel, poise

1 Úvod

Real-time souboj v RPG hře musí hráči poskytovat srozumitelnou zpětnou vazbu, ale zároveň nesmí působit mechanicky a předvídatelně. U arénového souboje je tento požadavek výrazný: hráč v krátkém čase sleduje vlastní výdrž, vzdálenost, animace, blokování, úhyby i chování více nepřátel. Jednoduchá AI založená pouze na přiblížení a útoku by sice byla snadno implementovatelná, ale při větším počtu protivníků by rychle vedla buď k zahlcení hráče, nebo k pasivním a málo zajímavým nepřátelům.

Cílem práce bylo navrhnout a implementovat funkční prototyp arénového RPG souboje v reálném čase. Důraz nebyl kladen na rozsáhlý herní obsah, ale na technický systém, ve kterém spolu souvisí pravidla souboje, vybavení, rozhodování nepřátel a řízení skupiny. Výsledný prototyp proto používá sdílený model mechanik pro hráče i nepřátele, hybridní AI architekturu a koordinační modul Combat Director.

Návrh vychází z obecných principů herní AI, zejména z kombinace stromů chování, utility rozhodování a sdílených datových struktur [1, 2, 3].

2 Návrh soubojového systému

Soubojový systém je navržen pro uzavřenou 3D arénu, ve které hráč postupuje po kolech. Hráč i nepřátelé používají stejný základní model pravidel: zdraví, výdrž, poise, blokování, úhyb, typy poškození a negace vybavení. Díky tomu nepřátelé nepůsobí jako oddělené skriptované objekty, ale jako entity podléhající stejným pravidlům jako hráč.

Výdrž omezuje opakování silných akcí. Útok, blokování i úhyb spotřebovávají zdroj, který se musí obnovovat, takže hráč i AI musí pracovat s krátkodobým rytmem souboje. Poise odděluje

poškození zdraví od přerušení akce. Slabší zásah může cíl zranit bez přerušení, zatímco útok s vyšším *poise damage* může vyvolat stagger. Při vyčerpání výdrže během bloku vzniká guard break, tedy silnější přerušení obranné akce.

Poškození je rozděleno na sečné, bodné a tupé složky. Zbroj neudává pouze jednu obrannou hodnotu, ale negace pro jednotlivé typy poškození a hodnotu *poise*. Zbraně naopak určují poškození, cenu ve výdrži, dosah, stabilitu při blokování, *poise damage* a dostupný *moveset*. Vybavení tím ovlivňuje nejen statistiky, ale i chování postavy v souboji. Přehled hlavních částí soubojového modelu uvádí tabulka 1.

Tabulka 1: Vybrané části soubojového systému a jejich význam

Systém	Herní funkce	Technický význam
Výdrž	Omezuje útok, blok a úhyb.	Brání opakování nejsilnějších akcí bez rizika.
Poise a stagger	Rozlišuje zranění a přerušení akce.	Umožňuje odlišit lehké a těžké zbraně.
Typy poškození	Sečné, bodné a tupé složky.	Propojuje zbraň, zbroj a negace.
Datové vybavení	Zbraně a zbroje jako samostatná data.	Ovlivňuje statistiky, animace i rozhodování AI.

3 Architektura AI nepřátel

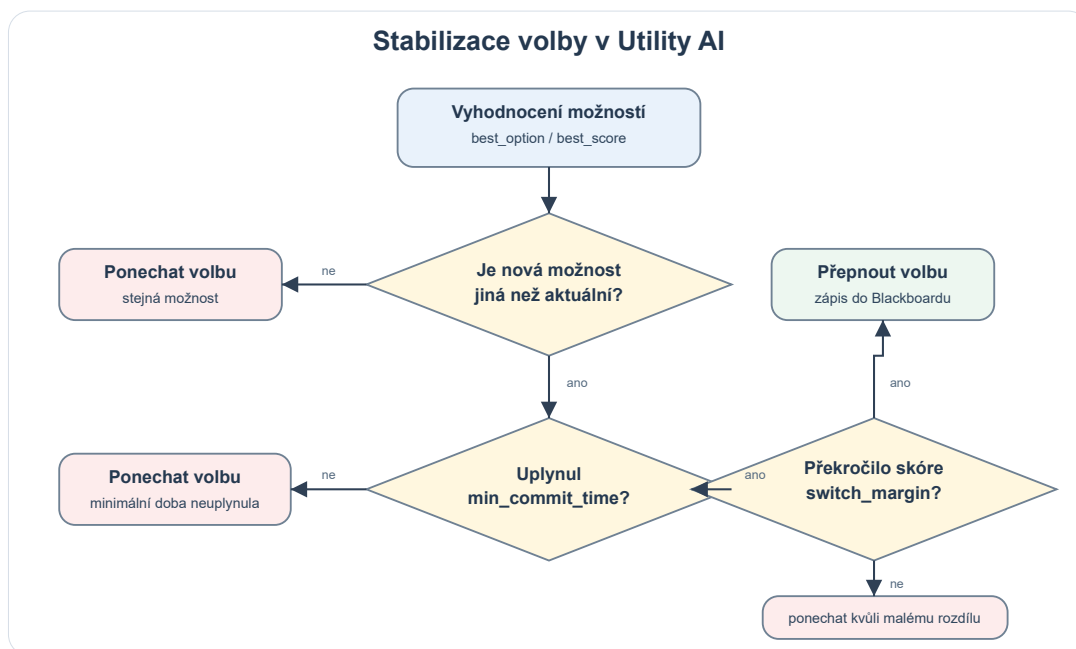
Nepřátelé používají hybridní architekturu složenou z Utility AI, Behaviour Tree a Blackboardu. Utility AI vybírá záměr, vhodnou zbraň a bezprostřední bojovou akci. Behaviour Tree tyto volby provádí ve scéně, například pohybem, útokem, blokem, úhybem nebo výměnou zbraně. Blackboard funguje jako sdílený datový prostor nepřítele, ve kterém jsou uloženy informace o vlastním stavu, cíli, bojovém kontextu i výstupech rozhodování.

Toto rozdělení omezuje nevýhody čistého stavového automatu. Soubojová situace obsahuje mnoho spojitých veličin, například vzdálenost, dostupnou výdrž, vhodnost zbraně, aktuální roli nebo hrozbu útoku. Utility AI tyto faktory převádí na skóre a vybírá nejvhodnější možnost [3]. Behaviour Tree je použit pro vlastní průběh akce, protože útok nebo úhyb nejsou jednorázová rozhodnutí, ale sekvence s podmínkami, animací a ukončením.

Konkrétní průběh rozhodování lze ukázat na nepříteli v roli Tank. Pokud je blízko hráče a hráč právě útočí nebo míří střelnou zbraní, zvýší se v Blackboardu hodnota obranného tlaku. Utility AI v takové situaci dočasně sníží atraktivitu přímého útoku a zvýhodní blok nebo úhyb podle toho, zda aktuální zbraň nebo štít umožňuje spolehlivou obranu. Po odeznění hrozby se skóre opět posune směrem k původnímu záměru držet tlak zepředu. Nepřítel tak nemusí mít ručně napsaný pevný scénář pro každou situaci; rozhodnutí vzniká z kombinace role, vzdálenosti, výdrže, vybavení a stavu hráče.

Utility AI přitom neprovádí akce přímo. Její výstupem jsou hodnoty zapsané do Blackboardu, například zvolený záměr, požadovaná zbraň nebo bezprostřední bojová akce. Behaviour Tree tyto hodnoty čte a převádí je na konkrétní práci ve scéně: navigaci k cíli, spuštění animace, aktivaci hitboxu, blokování, úhyb nebo výměnu zbraně. Tím je oddělena vrstva rozhodování od vrstvy provedení a jednotlivé části lze ladit samostatně.

Důležitou částí vlastní implementace je stabilizace rozhodování, znázorněná na obrázku 1. Nepřítel nevyhodnocuje novou volbu pouze podle okamžitě nejvyššího skóre, ale používá inter-



Obrázek 1: Stabilizace rozhodování v Utility AI. Přepnutí akce nastane až po splnění časové podmínky a dostatečného rozdílu skóre.

val vyhodnocení, minimální dobu držení volby a přepínací rezervu. Tím se omezuje chaotické přepínání mezi útokem, blokem, úhybem a změnou zbraně při malých změnách situace.

4 Skupinové chování a Combat Director

Při boji s více nepřáteli není vhodné, aby všichni protivníci útočili současně ze stejného směru. Takové chování je nepřehledné a často působí nespravedlivě. V prototypu je proto nad jednotlivými nepřáteli vytvořen Combat Director. Ten eviduje aktivní nepřátele, vyhodnocuje jejich vhodnost pro role a zapisuje výsledek do Blackboardu.

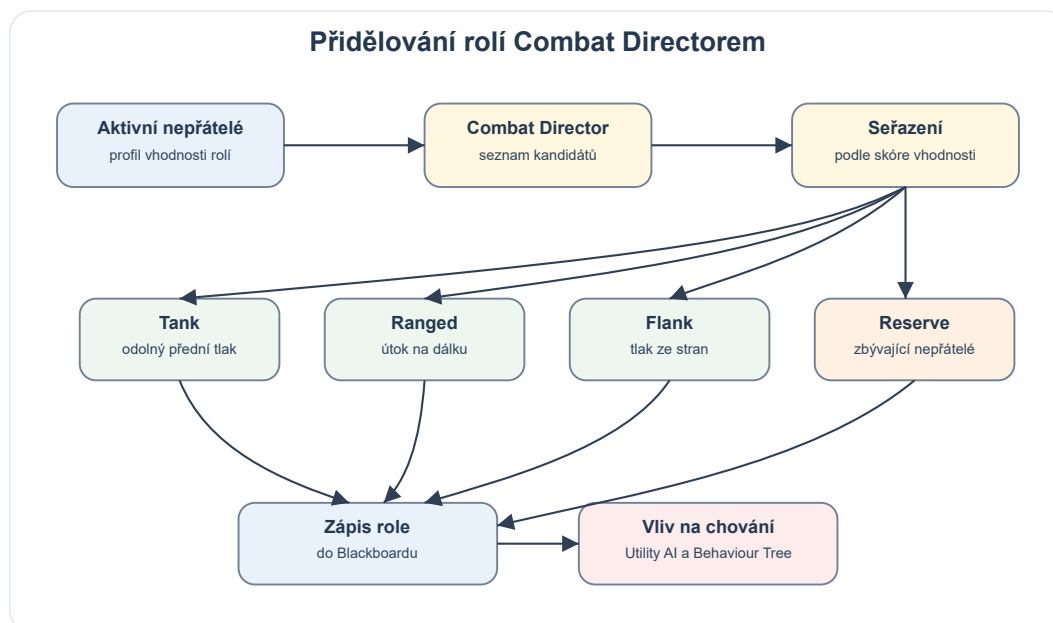
Použité role jsou Tank (přední odolný útočník), Flank (útočník ze stran), Ranged (útočník na dálku) a Reserve (záložní nepřítel). Podobné koordinační principy se používají i v komerčních hrách, kde samostatná vrstva omezuje konfliktní chování skupiny a přiděluje role podle situace [4]. V této práci je role určována hlavně podle vybavení nepřítele. Nepřítel se štítem a vyšší odolností je vhodnější pro Tank, střelná zbraň zvyšuje vhodnost pro Ranged a rychlejší zbraň může podporovat Flank. Přehled rolí, typického vybavení a jejich účelu shrnuje tabulka 2.

Tabulka 2: Role nepřátel používané Combat Directorem

Role	Vhodné vybavení	Chování v aréně	Účel pro hráče
Tank	Štít, stabilní zbraň, vyšší poise.	Drží tlak zepředu a častěji využívá blok.	Vytváří přímou hrozbu, kterou nelze dlouho ignorovat.
Flank	Zbraň na blízko s vhodným dosahem.	Hledá boční nebo zadní pozici vůči hráči.	Nutí hráče sledovat prostor, ne pouze jednoho protivníka před sebou.
Ranged	Luk, kuše nebo jiná zbraň na dálku.	Udržuje odstup a útočí projektily.	Trestá pasivitu a nutí hráče měnit pozici v aréně.
Reserve	Libovolné vybavení.	Má nižší aktivitu, čeká nebo vytváří omezený tlak.	Snižuje zahlcení hráče při větším počtu nepřátel.

Podstatné je, že Combat Director nevytváří pevné typy nepřátel. Stejná základní scéna nepřítele může být v jednom kole vyhodnocena jako Tank a v jiném jako Flank nebo Ranged podle toho, jakou výbavu dostane. Role tedy nevzniká ručním skriptováním samostatných tříd protivníků, ale jako výsledek aktuálního vybavení a jeho profilu vhodnosti. Tím vzniká variabilita bez nutnosti vyrábět velké množství oddělených nepřátelských archetypů.

Vazba mezi systémy má postup: vybavení určí profil vhodnosti role, Combat Director podle něj přidělí roli, role se zapíše do Blackboardu, Utility AI podle role upraví skórování možností a Behaviour Tree následně provede jiné chování ve scéně. Například štít zvyšuje stabilitu bloku a současně posiluje vhodnost pro Tank; zbraň na dálku naopak zvýší vhodnost pro Ranged a změní preferovanou vzdálenost. Vybavení je proto v návrhu nositelem herního i rozhodovacího významu.



Obrázek 2: Přiřazení rolí Combat Directorem podle profilu aktivních nepřátel a zápis role do Blackboardu.

Obrázek 2 ukazuje, že Combat Director neřídí každý pohyb nepřítele. Jeho úkolem je dodat skupinovou strukturu. Konkrétní pohyb a akce nadále řeší lokální AI nepřítele. Tato kombinace dovoluje, aby stejná základní scéna nepřítele mohla v různých kolech zastávat jinou roli podle vygenerované výbavy. Výsledný rozdíl rolí v aréně je znázorněn na obrázku 3.



Obrázek 3: Ukázka arény s vyznačenými rolemi nepřátel. Role mění typ tlaku, který nepřítel na hráče vytváří.

5 Implementace prototypu

Prototyp byl implementován v enginu Godot 4.5 s využitím jazyka GDScript. Godot byl zvolen kvůli scénovému modelu, dostupné práci s 3D scénou, animacemi, zdroji a signály [5, 6]. Behaviour Tree vrstva je realizována pomocí pluginu Beehave [7]; Utility AI, Blackboard a Combat Director byly vytvořeny jako vlastní systémy.

Projekt je rozdělen na scény hráče, nepřítel, zbraní, zbrojí, projektilů, arény, obchodu a uživatelského rozhraní. Vybavení je datově řízené: jednotlivé zbraně a zbroje nesou parametry poškození, negací, poise, ceny, dostupnosti a vazeb na animace. Při vybavení zbraně se mění nejen vizuální model, ale také moveset a hodnoty používané při rozhodování AI.

Herní smyčka prototypu má podobu kolo – souboj – odměna – obchod – další kolo. Hráč nejprve spustí kolo v aréně. Systém odstraní předměty z předchozí obchodní fáze, vytvoří skupinu nepřátel a každému z nich vygeneruje vybavení podle aktuálního kola. Z této výbavy se odvodí profil vhodnosti rolí, Combat Director rozdělí role a teprve poté proběhne vlastní souboj. Po poražení nepřátel hráč získá zlato, otevře se obchod a hráč může koupit nebo vyměnit vybavení před dalším kolem; příklad této fáze je na obrázku 4.

Tato smyčka propojuje souboj, ekonomiku, generování vybavení, obtížnost a AI role. Následující kolo nezvyšuje tlak pouze vyššími čísly, ale také dostupností jiných předmětů, vyšším počtem nepřátel a větší variabilitou jejich vybavení. Stejný systém dostupnosti se používá pro obchod i pro generování nepřátel, takže hráč i protivníci postupují v rámci jedné datově řízené progresse.

Vybavení není pouze vizuální prvek připojený k modelu. Zbraň mění moveset, typy poškození, poise damage, cenu útoku ve výdrži, optimální dosah a možnost blokování. Zbroj mění negace proti sečnému, bodnému a tupému poškození a přidává poise. Obchodní parametry určují cenu, raritu, minimální kolo výskytu a váhu ve výběru. Stejně datové vlastnosti používá obchod, hráčovo vybavení i generátor výbavy nepřátel, což snižuje duplicitu a dělá z vybavení společný jazyk pro více částí prototypu.

Obtížnost je parametrická. Předvolby mění hodnoty zdraví, výdrže, regenerace, poise, po-



Obrázek 4: Obchodní fáze prototypu. Předměty zobrazují cenu a bojové statistiky, které přímo ovlivňují další kola.

škození a také behaviorální parametry nepřátel, například agresivitu, defenzivitu a reaktivitu. S postupem kol se zvyšuje počet nepřátel, dostupnost vybavení a variabilita jejich slotů. Součástí prototypu jsou také vlastní modely, textury, animace, zvuky a prvky uživatelského rozhraní.

6 Testování a výsledky

Prototyp byl ověřován neformálním uživatelským testováním. Část testování probíhala na dálku přes Discord, kde hráči sdíleli obrazovku nebo poskytovali okamžitou zpětnou vazbu. Druhá část proběhla fyzicky na kolejích přibližně se šesti hráči. Cílem nebylo statistické vyhodnocení, ale praktické ověření, zda je souboj čitelný, zda role nepřátel dávají smysl a zda systém kol, odměn a obchodu tvoří funkční celek.

Testování sledovalo především čitelnost souboje, práci s výdrží, význam blokování a úhybů, chování více nepřátel, srozumitelnost obchodu a pocit z postupného růstu obtížnosti. Důležité bylo také ověřit, zda hráč chápe dopad vybavení: proč některý nepřítel vydrží přerušení akce, proč jiný drží odstup a proč se po výměně zbraně mění dostupné akce. Právě tyto situace ukazovaly, zda propojení mechanik a AI funguje i bez vysvětlování implementace.

Běžný průchod na obtížnost easy trval přibližně 10 minut. Nejdělsí průchody trvaly 30 až 40 minut a hráči se dostali téměř do 40. herního kola. Testování tím ověřilo nejen první střety, ale i pozdější fázi arénové smyčky, ve které už hraje roli generování vybavení, ekonomika a narůstající počet nepřátel.

Delší průchody jsou pro hodnocení prototypu důležité. Ukazují, že systém kol, odměn a obchodu funguje i mimo první minuty hry a že hráč má důvod pokračovat do dalších střetů. Pozdější kola současně ověřila generování výbavy nepřátel a růst obtížnosti: pokud by se vybavení odemykalo příliš rychle, hráč by narazil na nespravedlivý nárůst síly; pokud příliš pomalu, pozdější kola by působila jednotvárně.

Zpětná vazba vedla k úpravám vah Utility AI, stabilizačních parametrů rozhodování, odměn, generování vybavení a vyvážení výdrže. Pokud byli nepřátelé příliš agresivní, hráč neměl

prostor číst animace a reagovat. Pokud byli naopak pasivní, ztrácely význam role Combat Directoru. Největším výsledkem testování proto nebyla jedna izolovaná změna, ale ladění vazeb mezi vybavením, AI, ekonomikou a postupem kol.

V praxi se ukázalo, že cílem ladění není maximalizovat obtížnost, ale najít rovnováhu mezi tlakem a čitelností. Nepřítel má hráče nutit k rozhodování, ale zároveň musí být z jeho animace, role a pozice patrné, proč se chová určitým způsobem. Úpravy agresivity, odměn, výdrže a generování vybavení proto směřovaly k tomu, aby hráč měl prostor reagovat, ale nemohl souboj bezpečně ignorovat nebo obejít jednou dominantní strategií.

7 Závěr

Výsledkem práce je funkční technické demo arénového RPG souboje vytvořené v Godotu 4.5. Prototyp obsahuje sdílený model pravidel pro hráče a nepřátele, výdrž, blokování, úhyb, poise, stagger, guard break, typy poškození, negace podle vybavení, datově řízené zbraně a zbroje, systém kol, obchod, generování vybavení nepřátel a parametrickou obtížnost.

Hlavní vlastní přínos spočívá v propojení soubojových mechanik s hybridní AI architekturou. Utility AI vybírá záměr a akce, Behaviour Tree je provádí ve scéně, Blackboard sdílí stav mezi moduly a Combat Director přiřazuje role skupině nepřátel. Vybavení tak neurčuje pouze číselné hodnoty, ale mění i chování nepřítele a jeho roli v aréně.

Testování ukázalo, že prototyp je hratelný v opakovaných průchodech a že arénová smyčka souboj – odměna – obchod – další kolo dokáže ověřit navržené systémy v delším běhu. Další rozvoj by měl směřovat k rozšíření obsahu, přesnějšímu sběru testovacích dat, dalším typům nepřátel, bohatším movesetům, hlubšímu ladění AI a aktivnějšímu využití kontextu arény.

Literatura

- [1] Dawe, Michael; Gargolinski, Steve; Dicken, Luke; Humphreys, Troy; Mark, Dave. *Behavior Selection Algorithms: An Overview*. In: Rabin, Steve, ed. *Game AI Pro*. CRC Press, 2013. Dostupné z: https://www.gameapro.com/GameAIPro/GameAIPro_Chapter04_Behavior_Selection_Algorithms.pdf
- [2] Dill, Kevin. *Structural Architecture – Common Tricks of the Trade*. In: Rabin, Steve, ed. *Game AI Pro*. CRC Press, 2013. Dostupné z: https://www.gameapro.com/GameAIPro/GameAIPro_Chapter05_Structural_Architecture_Common_Tricks_of_the_Trade.pdf
- [3] Graham, David. *An Introduction to Utility Theory*. In: Rabin, Steve, ed. *Game AI Pro*. CRC Press, 2013. Dostupné z: https://www.gameapro.com/GameAIPro/GameAIPro_Chapter09_An_Introduction_to_Utility_Theory.pdf
- [4] McIntosh, Travis. *Human Enemy AI in The Last of Us*. In: Rabin, Steve, ed. *Game AI Pro 2: Collected Wisdom of Game AI Professionals*. CRC Press, 2016, s. 419--429. Dostupné z: https://www.gameapro.com/GameAIPro2/GameAIPro2_Chapter34_Human_Enemy_AI_in_The_Last_of_Us.pdf
- [5] Godot Engine contributors. *Nodes and Scenes*. Godot Engine 4.5 documentation, 2026. [cit. 2026-05-04]. Dostupné z: https://docs.godotengine.org/en/4.5/getting-started/step_by_step/nodes_and_scenes.html

- [6] Godot Engine contributors. *Resource*. Godot Engine 4.5 documentation, 2026. [cit. 2026-05-04]. Dostupné z: https://docs.godotengine.org/en/4.5/classes/class_resource.html
- [7] bitbrain. *Beehave Documentation*. 2026. [cit. 2026-05-05]. Dostupné z: <https://bitbrain.com/beekeep/>