

Návrh řešení

Autor: Marek Zalubil

Název práce: 2D RTS hra v Godotu — návrh řešení

1. Stručné shrnutí projektu

Cílem maturitní práce je vytvořit návrh a prototyp 2D real-time strategy (RTS) hry vyvíjené v enginu Godot. Hra bude obsahovat: správu surovin, stavbu budov, výcvik jednotek, rozhodovací AI pro nepřítele, uživatelské rozhraní pro příkazy (klikání + drag selection), a jednoduchý level se scénářem kampaně.

2. Model dat (ER model / třídy)

Níže je popsán základní datový model a návrh tříd/zdrojů používaných ve hře.

2.1 Hlavní entity (ER-přehled)

- **Player** (id, name, team_id, resources)
- **Resource** (id, type, amount, position)
- **Building** (id, type, owner_id, hp, position, production_queue)
- **Unit** (id, type, owner_id, hp, position, state, target)
- **UnitType** (id, name, cost, hp_max, speed, sight_range, attack)
- **MapTile** (x,y, terrain_type, walkable, resource_id)
- **GameEvent** (timestamp, type, payload)

2.2 Datové reprezentace v Godotu

- Ukládání konstantních dat (UnitType, BuildingType) jako **JSON** nebo **.tres** resources (Godot Resource).
 - Herní stav (GameState) jako singleton (AutoLoad) třída: uchovává seznam hráčů, jednotek, budov a mapu.
 - Uložení mapy: TileMap + doplňkové Tile data (Dictionary s indexem na resource).
-

3. Klíčové procesy a náročné části

Vyberu nejnáročnější části (AI + pathfinding + selection/command system + synchronizace herního stavu) a popíšu je.

3.1 Pathfinding (A* + lokální vyhýbání)

Proč náročné: mnoho jednotek pohybujících se současně, nutnost plynulých pohybů bez zahlcení.

Řešení: - Globální navigace: A nad gridem (tile map), heuristika Manhattan/Octile dle zrnitosti mapy. Lokální vyhýbání: jednoduché lokální posuny (steering) + velocity obstacles nebo boids-style separation, případně použití navmesh pro 2D (Godot Navigation2D). - Cacheování tras: ukládat výsledky A pro často používané cíle; invalidovat při změně mapy (stavby).

Optimalizace: použití D-heap priority queue, limitované vyhledávání pro blízké cíle, run length encoding mapy.

3.2 Unit command & selection system

Požadavky: multi-select (drag rectangle), right-click kontextové akce (move, attack, harvest), command queue per unit.

Návrh: - UI: Capture mouse drag vypočítat AABB v herních souřadnicích test kolize s Unit collision rect. - Pro vydaný příkaz: pokud target je enemy přidej `AttackCommand(target_id)`, jinak `MoveCommand(position)`. - Command scheduler: jednotka zpracovává frontu příkazů; některé příkazy jsou přerušitelné.

Edge cases: rozkazy při stání na místě; skupinové formace (offsety) – při move vytvořit rozptýl (formation) kolem cíle.

3.3 AI protivníka (strategická a taktická vrstva)

Rozdělení: - **Taktická AI:** bojuje s jednotkami (mikromanagement), používá pathfinding a cílení. **Strategická AI:** rozhoduje o výrobě jednotek, stavbě, expanzi, útocích. Řešení: jednoduchý stavový stroj nebo behavior tree.

4. Návrh uživatelského rozhraní (skicy / wireframes)

Níže jsou uvedeny návrhy hlavních obrazovek hry.

4.1 Hlavní herní obrazovka

Horní lišta: Panel zdrojů (Gold / Wood / Population), čas hry (paused / speed), tlačítka menu.

Levý dolní roh: Mini-map (s názvy týmů a ikonkami jednotek).

Pravý dolní roh: Unit panel (ikony, health bars, abilities) — při více jednotkách zobrazit skupinový panel.

Centrální oblast: herní svět (TileMap) s Tile Grid, budovami a jednotkami.

- ### 4.2 Dialog stavby / výcviku
- Context menu při kliknutí na budovu: seznam dostupných jednotek/upgrade, cost a build time, tlačítko „Queue“.

5. Návrh grafické podoby

5.1 Styl

- **Styl:** čistě 2D pixel art se středně realistickou paletou (nekaždý pixel je karikatura), rozlišení sprites 32×32 nebo 48×48.
- **Animace:** 4 směry (up/right/down/left) nebo 8 směrů pro plynulejší dojem; každá jednotka 3–6 fází chůze + attack frames.

5.2 Barvy a UI

- Použít omezenou paletu (např. 16–32 barev) pro soudržnost.
- UI prvky: ploché karty, jemný drop-shadow, kontrastní barvy pro selekci (cyan outline pro vybrané jednotky).

6. Nástroje / frameworky / řešení — volby a zdůvodnění

Shrnutí hlavních voleb a zdůvodnění:

6.1 Herní engine: Godot (doporučeno)

- **Proč:** open-source, dobrá podpora 2D, rychlé prototypování, GDScript snadno čitelný.
- **Alternativy:** Unity (silné, ale složitější), Defold (malá komunita), LÖVE (více ruční práce).
- **Výhody Godotu:** autoloady, integrovaný TileMap, rychlý editor.
- **Nevýhody:** menší ekosystém assetů než Unity.

6.2 Jazyk: GDScript (doporučeno)

- Rychlý vývoj, těsná integrace. Alternativa C# pro pokročilejší optimalizace.

6.3 Grafika: Aseprite / Krita

- Aseprite pro pixel art a animace. Krita pro UI a koncepty.

6.4 Verzeování: Git + GitHub / GitLab

- Důvody: historie, zálohování, veřejný odkaz pro hodnocení.

6.5 Audio: Audacity + freesound.org

7. Scénář hry / návrh levelů, postav a předmětů

7.1 Hlavní koncept / příběh

Konflikt dvou frakcí na hranici říše. Hráč jako velitel buduje základnu, trénuje jednotky a bojuje o suroviny.

7.2 Levely

1. **Tutoriál:** základní ovládání, těžba, stavba.
2. **Obrana:** omezené zdroje, obrana základny.
3. **Expanze:** mapa s více základnami a AI soupeřem.

7.3 Typy jednotek a budov

- Worker, Infantry, Archer, Siege.
- Headquarters, Barracks, Tower, Resource Depot.