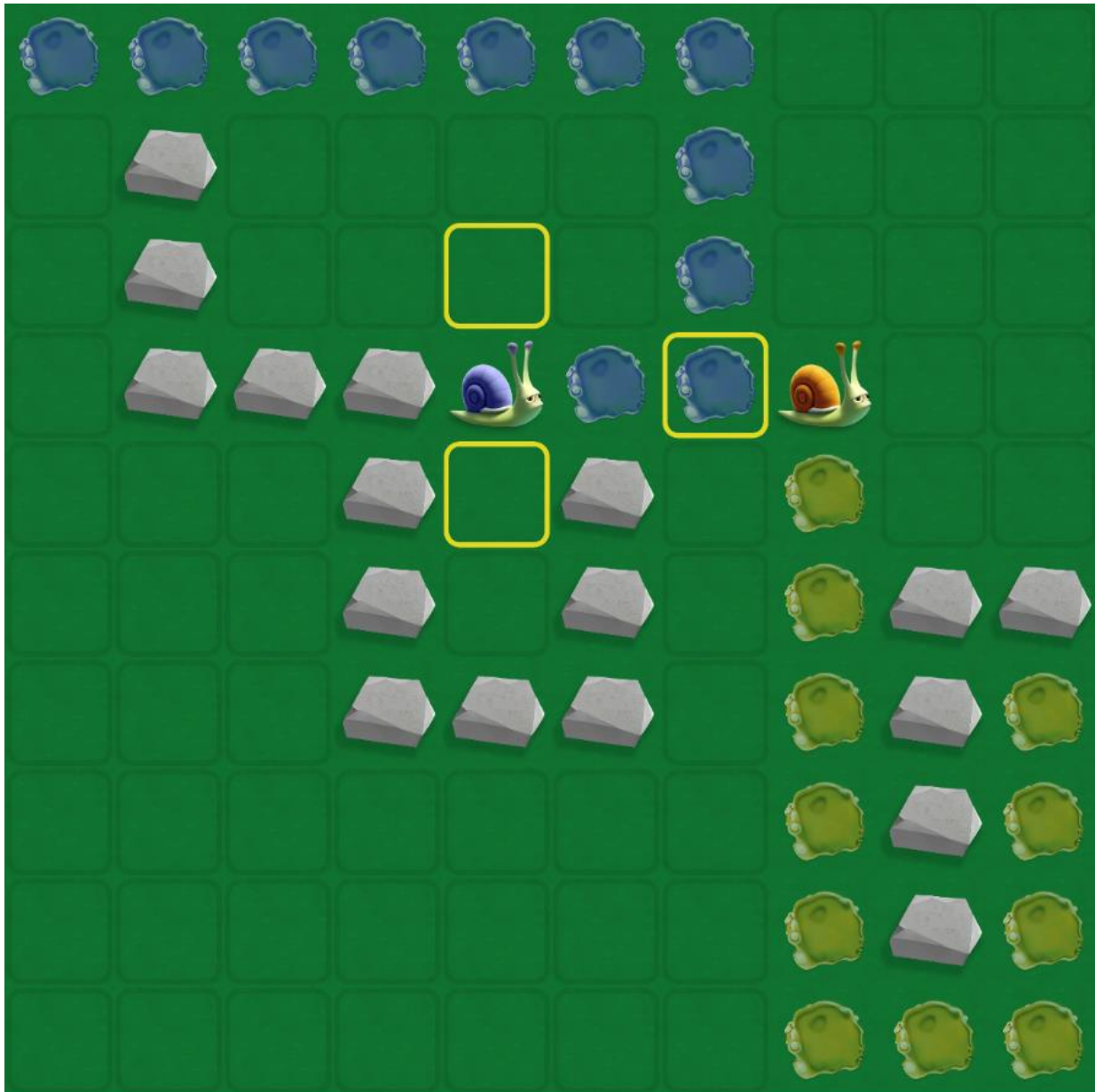


Projektarbeit B-GD-ALL-GPG-23SS: Snails



Technische Spezifikationen

- Unity-Version: 2021.3.22 (LTS)
- Empfohlenes Unity-Template: 2D

Spielprinzip und Spielregeln

- In diesem rundenbasierten Zweispieler-Brettspiel bewegen sich die Spieler, zwei Schnecken, abwechselnd über ein Spielfeld aus quadratischen Kacheln, um diese durch Betreten mit ihrem Schleim in der Spielerfarbe zu markieren und einzunehmen
- Pro Zug führt ein Spieler eine der folgenden Aktionen aus:
 - Bewegung in Richtung eines der horizontal oder vertikal angrenzenden, begehbaren Felder
 - Die normale Schrittweite beträgt ein Feld
 - Betritt eine Schnecke ihre eigenen Schleimspur, so rutscht sie entlang der Bewegungsrichtung über angrenzende Felder hinweg bis zum Ende der Spur
 - Aussetzen (keine Bewegung)
 - Aufgeben, wobei unabhängig vom Punktestand der gegnerische Spieler gewinnt
- Ein Spieler erhält einen Punkt für jedes durch ihn eingenommene Feld
- Eingenommene Felder können nicht mehr vom Gegner betreten werden
- Nimmt ein Spieler kein neues Feld innerhalb der festgelegten Anzahl an aufeinanderfolgenden Zügen¹ ein, so muss er alle verbleibenden Züge aussetzen
- Das Spiel endet, sobald
 - alle leeren Felder eingenommen wurden
 - oder
 - beide Spieler für den Rest des Spiels aussetzen müssen
- Sofern kein Spieler vorzeitig aufgegeben hat, gewinnt der Spieler mit der höchsten Gesamtpunktzahl

Spielobjekte

	Begehrbar durch Spieler	Begehrbar durch Gegner
Leeres Feld ²	Ja	Ja
Eigene Schnecke ³	Ja	Nein
Gegnerische Schnecke ⁴	Nein	Ja
Eigene Schleimspur ⁵	Ja	Nein
Gegnerische Schleimspur ⁶	Nein	Ja
Steinblock ⁷	Nein	Nein

Player-Agents

- Jede Spielfigur wird von einem der folgenden Player-Agents gesteuert:
 - Benutzergesteuerter Player-Agent
 - Klick mit primärer Maustaste auf eines der möglichen Zielfelder zum Ausführen der Aktion (Bewegen)
 - Klick mit primärer Maustaste auf den Spieler oder Klick mit sekundärer Maustaste auf beliebige Position zum Aussetzen

¹ Die maximale erlaubte Anzahl aufeinanderfolgender Züge ohne Einnahme eines neuen Felds ist in den Spieleinstellungen definiert

² Sprite „empty_256x256.png“ im Asset-Pack enthalten

³ Sprite „snail_blue_256x256.png“ im Asset-Pack enthalten

⁴ Sprite „snail_orange_256x256.png“ im Asset-Pack enthalten

⁵ Sprite „trail_blue_256x256.png“ im Asset-Pack enthalten

⁶ Sprite „trail_orange_256x256.png“ im Asset-Pack enthalten

⁷ Sprite „impassable_256x256.png“ im Asset-Pack enthalten

- ESC-Taste zum Aufgeben
- Computergesteuerter Player-Agent
 - Die Aktionen dieses Spielers werden unter Verwendung der bereitgestellten Bibliothek⁸ berechnet

Spielvorbereitung

- Der Benutzer wählt für beide Spieler die Art des Player-Agents (Dropdown)
 - Beliebige Kombinationen sind möglich, auch Computer vs. Computer
- Der Benutzer kann optional pro Spieler einen Spielernamen eingeben
 - Wird kein Spielernamen vergeben, heißt der Spieler Player 1 bzw. Player 2
- Der Benutzer wählt eine der verfügbaren Maps aus (Dropdown)
 - Das Spiel sollte alle Maps, die sich im Map-Verzeichnis befinden, erkennen und die Dateinamen ohne Endung alphabetisch absteigend sortiert anzeigen
 - Sollte eine Map nicht geladen werden können, sollte dies durch eine Fehlermeldung im GUI angezeigt werden
- Das Spiel wird bei Betätigung des Play-Buttons gestartet
- Ein zufällig ausgewählter Spieler beginnt

Spielablauf

- Spieler und der Spielzug werden durch eine Texteinblendung im Format „Round {turnNumber} - {playerName}“ und einem Sound⁹ angekündigt
- Alle möglichen nächsten Zielpositionen werden grafisch¹⁰ hervorgehoben
 - Bei benutzergesteuerten Player-Agents wird auf die Eingabe gewartet, bei Computergegnern werden die Zielfelder mindestens für die in den Spieleinstellungen definierte Dauer angezeigt
 - Befindet sich der Mauszeiger über einem hervorgehobenen Feld, so sollte die Hervorhebung leicht größer werden und ein Klick-Sound¹¹ abgespielt werden
- Der Spieler führt eine Aktion aus
 - Bei Überschreiten des Zeitlimits¹² endet der Zug mit der Aktion „Aussetzen“
- Die ausgeführte Aktion wird durch eine entsprechende Texteinblendung mit Sound angekündigt:
 - Slide-Aktion¹³: „{playerName}: Slide to ({x}, {y})“
 - Einnehmen eines neuen Feldes¹⁴: „{playerName}: Capture ({x}, {y})“
 - Aussetzen¹⁵: „{playerName}: Miss“
- Gleichzeitig wird die Spielfigur von ihrer derzeitigen Position mit animiertem Übergang zum Zielfeld bewegt

⁸ Die Bibliothek enthält hierzu die Klasse `Snails.Agent.NegamaxAgent`; Dieser Player-Agent berechnet die nächste Aktion basierend auf dem aktuellen Spielzustand (`Snails.IWorldState`). Die Bibliothek stellt hier zusätzlich die Klasse `Snails.WorldState` bereit, welche die Schnittstelle `Snails.IWorldState` implementiert. Diese kann optional verwendet werden.

⁹ Sound „user-interface-menu-select-click-01“ von weloveindies.com

¹⁰ Sprite „empty_256x256.png“ und pulsierender Sprite-Shader im Asset-Pack enthalten

¹¹ Sound „ui-click-high-modern-click-06“ von weloveindies.com

¹² Die Maximaldauer eines Zuges ist in den Spieleinstellungen definiert

¹³ Sound „ui-modern-confirmation-ascending-03“ von weloveindies.com

¹⁴ Sound „ui-modern-confirmation-ascending-03“ von weloveindies.com

¹⁵ Sound „ui-modern-confirmation-descending-01“ von weloveindies.com

Spielende

- Der Results-Screen wird angezeigt

Maps

- Mapdaten werden in einer Textdatei im JSON-Format definiert
 - Größe des Spielfeldes¹⁶ als „size“ mit zwei Eigenschaften „x“ und „y“
 - Position und Art der Spieleobjekte¹⁷ auf dem Spielfeld als Array „contents“. Die Array-Elemente sind dabei nach dem folgenden Schema geordnet

0	1	2
3	4	5
6	7	8

- Die verwendeten Spieler-IDs sind 1 für Spieler 1 und 2 für Spieler 2
- Die im Asset-Pack enthaltene Map „default.json“ der Größe 5x5 sollte geladen werden können und wie folgt aussehen



- Das Spiel beinhaltet zwei weitere Maps¹⁸ der Größen 7x7 und 10x10 Kacheln
- Jede Map wird in einer eigenständigen Textdatei definiert
- Alle Mapdateien liegen im Verzeichnis „{Application.streamingAssetsPath}/maps“
- Nur Maps, die die in der Konfigurationsdatei festgelegten Vorgaben zur Kantenlänge und zum Format des Spielfeldes erfüllen, sollen vom Spiel geladen werden

Konfiguration

- Die Spieleinstellungen sind in einer Textdatei „settings.gcf“ im Verzeichnis {Application.streamingAssetsPath} im JSON-Format definiert und werden beim Start der Anwendung geladen
- Das Spiel unterstützt mindestens die folgenden Einstellungen:
 - Maximaldauer eines Zuges in Sekunden als „maxTurnDuration“ (Default: 30s)
 - Maximale Anzahl erlaubter aufeinanderfolgender Züge ohne Markierung als „maxTurnsWithoutCapture“ (Default: 5)
 - Maximale Rechenzeit für computergesteuerte Player-Agents pro Spielzug in Millisekunden als „maxComputationTimePerTurn“ (Default: 500ms)

¹⁶ Zur (De-)Serialisierung der Größe soll die in der Bibliothek enthaltene Klasse Snails.Vector2Int verwendet werden, die sich mit dem System.Text.Json.JsonSerializer (de-)serialisieren lässt

¹⁷ Zur (De-)Serialisierung der Tile-Inhalte soll die in der Bibliothek enthaltene Klasse Snails.TileData verwendet werden, die sich mit dem System.Text.Json.JsonSerializer (de-)serialisieren lässt.

¹⁸ Nicht im Asset-Pack enthalten

- Minimale Anzeigezeit der Zielfelder bei der Aktionsauswahl für computergesteuerte Player-Agents in Sekunden als „minCPUAgentTurnDuration“ (Default: 1.0s)
- Minimale unterstützte Kantenlänge des Spielfeldes als „minMapSize“ (Default: 5)
- Maximale unterstützte Kantenlänge des Spielfeldes als „maxMapSize“ (Default: 20)
- Vorgabe zum Seitenformat als „requireSquareMap“ (Default: false); Falls true, können nur quadratische Maps geladen werden

Screens

Setup-Screen

- Auswahl der Player-Agents für Spieler 1 und Spieler 2
- Eingabe der Spielernamen
- Auswahl der Map
- Alle Einstellungen im Setup-Screen sollten während der Laufzeit der Applikation beibehalten werden und nicht nach jeder Spielrunde neu eingegeben werden müssen
- Reload-Settings-Button, um die Spieleinstellungen neu zu Laden
- Play-Button, um das Spiel zu starten
- Beenden der Applikation durch Quit-Button und mit der ESC-Taste

Game

HUD

- Oben mittig: Nummer des Zugs, verbleibende Zeit des Zuges
- Oben links: Avatar, Name und Punktzahl des Spielers
- Oben rechts: Avatar, Name und Punktzahl des Gegners
- Der Name des Spielers, der gerade am Zug ist, sollte farblich hervorgehoben werden
- Mittig: Spielankündigungen zu Beginn und Ende eines Spielzuges

Results-Screen

- Anzeige des Gewinners, des Punktestandes und der Anzahl gespielter Runden zum Ende einer Partie
- Replay-Button, um direkt eine erneute Partie mit denselben Voreinstellungen zu spielen
- Continue-Button, um zum Setup-Screen zurückzukehren

Erweiterung: Highscore-Liste

- Im Results-Screen soll eine Mini-Highscore-Liste angezeigt werden
 - Eine Highscore-Liste besteht aus maximal drei Einträgen
 - Jeder Eintrag besteht aus dem Namen und der Punktzahl und dem Player-Agent („CPU“ oder „Human“) des Gewinners sowie dem Namen und dem Player-Agent des Gegners
 - Die Einträge sind absteigend nach der Punktzahl sortiert
 - Neuere Einträge mit übereinstimmender Punktzahl erscheinen über älteren Einträgen
 - Ein neuer Eintrag ist farblich hervorgehoben
 - Jede Map hat eine eigene Highscore-Liste
 - Einträge sollten beim Beenden der Applikation nicht verloren gehen

Robustheit

- Das Spiel sollte auch dann funktionsfähig sein, wenn einzelne oder mehrere Einstellungen in der Konfigurationsdatei nicht definiert sind oder wenn die Konfigurationsdatei nicht vorhanden ist
- Das Laden einer Map mit fehlerhaften Daten sollte erkannt und mit einer Fehlermeldung im UI abgebrochen werden. Insbesondere zu bedenken sind:
 - Falsches Format
 - Inkonsistente Daten, z.B.
 - Die angegebene Mapgröße stimmt nicht mit den tatsächlichen Daten überein
 - Die Map enthält nicht exakt einen Spieler mit der ID 1 und einen Spieler mit der ID 2
 - Die Map enthält ungültige Spieler, bspw. einen Spieler mit der ID 3
- Das Spielfeld sollte unabhängig von der Größe vollständig auf dem Bildschirm sichtbar sein
- Kein Spieler-Agent – weder benutzer- noch computergesteuert – sollte in der Lage sein, ungültige Züge durchzuführen