Výuková pomůcka - počítačová hra - Klasická hra V Kostky

Bakalářská práce

Vedoucí práce:

doc. Ing. Oldřich Trenz, Ph.D. **Lukáš Hlavatý**

Brno 2017

Rád bych tímto poděkoval doc. Ing. Oldřichu Trenzovi, Ph.D. za připomínky a rady, jež mi dopomohli tuto práci vytvořit.

Čestné prohlášení
Prohlašuji, že jsem bakalářskou práci vytvořil zcela samostatně pod vedením svého vedoucího bakalářské práce a že jsem uvedl všechny použité prameny a literaturu, ze které jsem čerpal.

V Brně dne 4. ledna 2017 _____

Abstrakt

Hlavatý, L. *Teaching tool – computer game – classic game Dice 10,000*. Bachelor Thesis.
Brno, 2017.
In the first part focuses on important relationships of probability and dice games. Later, it presents basic principles of important probability calculations. The second part consists of computer program implementation which simulates a classic game of "Dice 10,000" in a computer vs. human variant. Theoretical knowledge is applied in form of automated decisions based on the selected algorithm in the first part.

Keywords

Bachelor thesis, probability, dice, computer program, automated decisions

Abstrakt

Hlavatý, L. *Výuková pomůcka - počítačová hra - Klasická hra V kostky*. Bakalářská práce.
Brno, 2017.

V první části se práce zaměřuje na důležité vztahy pravděpodobnosti a kostkových her a představuje základní principy důležitých výpočtů. Druhá část se zabývá tvorbou počítačového programu simulujícího klasickou hru "V Kostky" ve variantě počítač versus člověk, kde jsou teoretické poznatky aplikovány jako podpora pro automatické rozhodování počítače.

Klíčová slova

Bakalářská práce, pravděpodobnost, kostky, počítačový program, automatické rozhodování

Obsah

1 Úvod a cíl práce

1.1 Úvod

Jedním z dynamicky se rozvíjejících oborů informatiky je bezesporu umělá inteligence. Její aplikace zasahuje čím dál tím více do našich životů. Počítačové rozhodování dnes využíváme v celé řadě odvětví. Počínaje efektivním řízením továrních linek, přes složité postupy v logistice až po předvídání vývoje burz či měnových kurzů. Počítače nám již pomáhají řídit automobil, stejně jako pilotovat letadlo. Ve všech těchto oborech využíváme rychlosti výpočtů a necháváme počítačové programy více či méně rozhodovat za nás.

Pokud se budeme zabývat vývojem v oblasti umělé inteligence, lze se zde pustit hned do několika disciplín. A jednou z nich jistojistě umělá inteligence v oblasti teorie her. Právě poznatky z této oblasti nám mohou posloužit jako dobrý podklad, pro další práci. A tak se dnes přední počítačové špičky předhánějí, čí superpočítač zahraje lépe šachy, či porazí nejlepší hráče světa ve hře Go.

Ačkoliv tyto příklady se týkají především velkých nadnárodních teamů expertů s vysokým rozpočtem a hrubou výpočetní silou, posloužili zmíněné experimenty jako inspirace pro tuto práci. V té se zaměříme na strojové řešení jedné z tradičních kostkových her.

1.2 Cíl práce

Cílem této práce je výuková aplikace do předmětu Umělá inteligence. Jejím hlavním úkolem je realizace efektivního automatického rozhodování ve známé kostkové hře V kostky. Hru by mělo být možné hrát ve variantě člověk vs. počítač, tak jak je tomu zvykem i v projektech větších, jako byly výše zmiňované šachy, či hra Go. I v tomto případě je pak kladen důraz na co nejvyšší efektivitu hry počítače, s cílem hru proti hráči ovládnout a vyhrát.

Práce se také v úvodu zaměří na některé důležité pojmy z oblasti pravděpodobnosti a teorie her. V podrobnějším popisu hry pak tyto znalosti využijeme, pro identifikaci důležitých principů, jež bude aplikace při výběru optimální strategie využívat. V samotném závěru pak zhodnotíme výsledky a navrhneme další možnosti rozšíření.

2 Pravděpodobnost

Jelikož se budeme v této práci zabývat programováním kostkové hry, je třeba zaměřit se i na výpočty pravděpodobností. Házení kostkou je totiž jedním z typických příkladů, kdy může výpočet pravděpodobností hrát velkou roli.

2.1 Základní pojmy

2.1.1 Pravděpodobnost náhodného jevu

Mezi základní pojmy patří pravděpodobnost náhodného jevu. Ta nám udává, s jak velkou šancí je možné, že daný náhodný jev nastane (MATEMATIKA.CZ, 2016). Vybereme-li si například za náhodný jev případ, kdy padne na kostce hodnota 5, můžeme k tomuto jevu i dopočítat s jakou pravděpodobností tento stav nastane. Pro vyjádření výsledku se pak běžně užívá více možností:

- Číslo z intervalu <0; 1>
- Pravděpodobnost vyjádřená v procentech
- Pravděpodobnost vyjádřená zlomkem či poměrem

O pravděpodobnosti, že na kostce padne právě číslo pět, lze tedy prohlásit, že je rovna 1/6, což je přibližně 0,1667, či 16,67%. Můžeme si také povšimnout, že procentuální vyjádření pravděpodobnosti je stejné, jako když číselné vyjádření vynásobíme hodnotou 100.

2.1.2 Náhodný jev

Nejen hod kostkou je náhodnou veličinou. Tou se stává jakýkoliv jev, který je závislý na pravděpodobnosti (MAREK, 2012). Mohou to být například loterijní hry, hry karetní, ale i různé případy spojené se statistickými údaji. Např. pravděpodobnost, že z výrobní linky vyjde vadný kus, či že v určitém místě dojde k dopravní nehodě atd.

Pro správné určení pravděpodobnosti je proto důležité, aby sledovaný jev měl podmínky neměnné. Po každém hodu kostkou musí být pravděpodobnost daných jevů stejná. Do práce výrobní linky nesmí zasahovat žádný vnější jev, a pokud ano, tak stejným způsobem.

2.1.3 Množina všech možných výsledků

Jedná se o množinu všech možných stavů, ve kterých se sledovaný náhodný jev může ocitnout (MATEMATIKA.CZ, 2016).V případě házení kostkou, jsou to čísla 1-6. Při losování číselné loterie, jsou to veškeré možné číselné kombinace, jež mohou být vylosovány. U stíracích losů to mohou být veškeré výherní varianty (pro výhry různé hodnoty) i ty nevýherní.

2.1.4 Jev jistý a nemožný

Statistika zavádí dva speciální případy náhodného jevu. Jev jistý, je jev takový, ke kterému dochází s pravděpodobností 1 (respektive 100%). Jinými slovy, jedná se o jev, ke kterému dojde vždy. Takovým jevem může být například pravděpodobnost, že na šestistěnné kostce padne číslo v intervalu 1-6.

Naopak jevem nemožným rozumíme případ, ke kterému nemůže dojít nikdy. V řeči čísel nabývá jev pravděpodobnosti 0, či 0%. Pro hod kostkou to může být pravděpodobnost, že padne číslo 7, či že po hodu dvěma kostkami padne v součtu více jak 15. Jelikož maximální hodnota šestistěnné kostky je 6, nikdy nemůžeme hodit číslo vyšší. A podobně je tomu i po hodu dvěma kostkami, kde je maximální možný součet 12.

2.2 Klasická pravděpodobnost

Klasická, respektive Laplaceova pravděpodobnost je jednou z mnoha definicí a pohledů na danou problematiku. Ačkoliv popisuje nejběžnější případy pravděpodobnosti, zdaleka neobsáhne všechny případy.

Laplaceova pravděpodobnost vychází z těchto předpokladů (WIKIPEDIA.ORG, 2016):

1. Všech možných výsledků je konečný počet.
2. Všechny výsledky jsou stejně možné.
3. Všechny výsledky se vzájemně vylučují.

2.2.1 Konečný počet výsledků

První bod nás omezuje pouze na případy, kdy je možné identifikovat veškeré možné výsledky. Identifikace všech možných výsledků bude důležitá, pro správné sestavení vzorce.

2.2.2 Výsledky jsou stejně možné

Při výpočtu klasické pravděpodobnosti, je důležité, aby ke všem výsledkům docházelo se stejnou pravděpodobností. Budeme li identifikovat veškeré možné výsledky, které můžeme hodit na běžné kostce, nalezneme jich šest (čísla 1-6). Každý z výsledků má však stejnou pravděpodobnost, jelikož je na kostce zastoupen stejně často. Představme si ale případ, kdy bude mít kostka stěn sedm a bude obsahovat čísla 1,1,2,3,4,5,6. V takovém případě máme sice stejný počet výsledků (1,2,3,4,5,6), ale číslo 1 je zde zastoupeno dvakrát, zatímco zbylá čísla pouze jednou. Pravděpodobnost, že padne číslo jedna (tedy jednoho z možných výsledků) je tak vyšší, nežli pravděpodobnost výsledků ostatních. Takovou kostku si lze jen těžko představit, nicméně lze si snadno představit, že máme běžnou kostku, která je ale špatně vyvážená a proto na ni padají určitá čísla častěji nežli jiná.

Zde je však nutné podotknout, že v případě sedmistěnné kostky jednotlivé strany stále padají se stejnou, předem známou pravděpodobností 1/7 a tak i taková úloha je Klasickou pravděpodobností řešitelná. Naopak běžný případ kostky nevyvážené již tímto způsobem řešit možné není.

2.2.3 Výsledky se vzájemně vylučují

Tímto bodem rozumíme, že pokud nastane výsledek jeden, nemůže v danou chvíli nastat jakýkoliv jiný, či jinými slovy, nemůže dojít k více výsledkům současně. Stejně jako u hrací kostky může padnout pouze jedna strana a tak nelze dosáhnout například hodnoty 2 a 5 v jednom hodu.

2.2.4 Princip výpočtu

Splňuje-li řešená úloha výše zmíněné vlastnosti, lze ji řešit pomocí vzorce pro klasickou pravděpodobnost (MAREK, 2013). Ten stanovuje výsledek pravděpodobnosti jevu A, jakožto podíl množství příznivých výsledků (m) s množstvím všech možných výsledků (n).

$$P(A) = \frac{m}{n}$$

Je tedy zřejmé, že podstatou výpočtu jsou dva základní kroky. Nalezení všech výsledků, při kterých nastává jev A, a dále identifikace veškerých možných výsledků, k nimž může dojít.

Pojďme se podívat na jednoduchý příklad pravděpodobnosti, že na kostce padne číslo dělitelné třemi.

Nejprve nalezneme všechny výsledky, jež mohou na kostce padnout. Jsou to čísla 1,2,3,4,5,6. Počet všech možných výsledků n je tedy 6. Následně musíme tyto výsledky projít a naleznout takové, které podmínku dělitelnosti třemi splňují. Je to výsledek 3 a výsledek 6. Celkový počet příznivých výsledků m se tedy rovná 2.

Následně již stačí jednoduše podělit P(A)= m/n = 2/6 = 0,3333. Pravděpodobnost jevu A, že při hodu kostkou padne číslo dělitelné třemi, je tedy přibližně 0,3333, což je 33,33%.

Nyní se podívejme na případ opakovaného házení. Nejprve pravděpodobnost, že při dvou hodech padnout dvě jedničky.

I v tomto případě lze postupovat obdobně. Nejdříve stanovíme veškeré možné výsledky, které mohou padnout. V našem případě hledáme dvojce výsledků z prvního a druhého kola házení. Takových dvojic nalezneme celkem 36. Nyní veškeré výsledky projdeme a budeme hledat ty, které obsahují pár jedniček. Taková dvojce je ve výčtu pouze jediná. Následné údaje opět vložíme do vzorce a dopočítáme pravděpodobnost P(A)= 1/36= 0,0277 tedy 2,7%.

V posledním příkladu však můžeme využít i poznatku o násobení pravděpodobností (MAREK, 2013). Při prvním hodu, je pravděpodobnost padnutí jedničky 1/6, stejně tak jako při hodu druhém, jelikož šance na padnutí konkrétního čísla je na dané kostce neměnná bez ohledu na počet hodů. Jelikož ale počítáme, že výsledek druhého hodu je úspěšný pouze v případě, kdy je úspěšný i hod předchozí, je nutné pravděpodobnosti jednotlivých kol vynásobit. Tím je vyjádřena závislost jednoho jevu na druhém. Výsledný vzorec je tedy P(A)= 1/6 * 1/6 = 1/36= 0,0277.

Někdy může nastat situace, kdy se výsledek skládá z více případů, jež je nutné sečíst. Takovým je i příklad, ve kterém se ptáme na šanci, že jednička padne ve dvou hodech právě jednou.

Řekněme, že jednička padne v hodu prvním a v druhém už ne, což našemu příkladu vyhovuje. Šance že jednička padne v prvním hodu je 1/6 a šance, že v druhém nepadne, je 5/6 (jelikož případů, kdy nepadne jednička je právě pět – 1,2,3,4 a 6). Abychom znali pravděpodobnost, že oba jevy nastanou zaráz, je nutné opět tyto hodnoty vynásobit. Výsledkem je tedy 1/6 * 5/6= 5/36. Tento výsledek je však neúplný. Nesmíme totiž zapomenout i na případ, kdy jednička naopak padne v kole druhém, zatímco v prvním ne. Tu vypočítáme obdobně 5/6 * 1/6 = 5/36. Tyto dva výsledky lze považovat za dvě možnosti, při kterých může hledaný jev nastat. Počet všech možností (n) nám zůstává stejný – 36. A počet příznivých jevů (m) je 5 + 5, tedy 10. Celkovou pravděpodobnost pak už vypočítáme standardním vzorcem P(A)= 10/36 = 5/18 = 0,2777.

2.3 Binomické rozdělení

Ve složitějších případech, kdy realizujeme větší množství hodů, může být výpočet pravděpodobnosti poněkud nepřehledný a složitý. Lze však k problematice přistupovat i jinak. Uvědomíme-li si, že opakované házení kostkou nikterak neovlivňuje pravděpodobnost padnutí jednotlivých hodnot, lze o takových pokusech prohlásit, že jsou na sobě nezávislé.

Při kostkových hrách často hledáme hody více kostkami, při kterých padla nějaká specifická kombinace, často tvořená čísly stejné hodnoty. Bývají to např. stejné trojce, čtveřice apod. Klasická pravděpodobnost nám velí, dohledat všechny možné výsledky a v nich objevit veškeré výsledky pozitivní. Nicméně pokud budeme hledat pravděpodobnost padnutí trojce při hodu šesti kostkami, začíná být množství výsledků příliš vysoké a výpočet tímto způsobem značně zdlouhavý a nepraktický.

Úlohu lze však převést na jiný případ. Věnujme se nyní hledání pravděpodobnosti, že při hodu šesti kostkami, padne trojce jedniček. Jak jsme si již řekli, hod každou kostkou je na ostatních nezávislý. Lze si tedy úlohu přestavit, jako hru s jednou kostkou a šesti pokusy. V takovém případě pak vlastně hledáme pravděpodobnost, že z šesti pokusů padne právě třikrát jednička. Řekneme-li, že padnutí jedničky je pro nás úspěch a padnutí jiného čísla neúspěch, hledáme pak pravděpodobnost, že při šesti pokusech dosáhneme právě tří úspěšných. A právě tím se zabývá Binomické rozdělení.

2.3.1 Vlastnosti

Binomické rozdělení Bi(n ,p) popisuje četnost náhodného jevu v n pokusech (HOMEN, 2016). Podmínkou je nezávislost realizovaných pokusů a neměnící se pravděpodobnost.

A právě tyto vlastnosti naše úloha splňuje. Máme zde 6 nezávislých pokusů (předchozí hod nikterak neovlivní hod následující) a neměnnou pravděpodobnost úspěchu. Ta je 1/6 v každém hodu, že padne číslo jedna. Namísto složitého hledání a procházení všech výsledků tak můžeme využít jednoduchého vzorce.

$$P(x) = \binom{n}{x} p^x (1-p)^{n-x}$$

n – počet všech realizovaných pokusů
x – počet úspěšných pokusů
p – pravděpodobnost úspěchu

Tímto způsobem náš vzorec hledá pomocí kombinačního čísla příslušný počet vyhovujících kombinací, dále ve výpočtu zohledňuje pravděpodobnost úspěšného tahu a stejně tak i pravděpodobnost neúspěchu, jež je počítána jako doplněk pravděpodobnosti úspěchu (1-p).

Do vzorce tedy dosadíme tři úspěšné pokusy za x, šest pokusů celkem za n a 1/6 jakožto pravděpodobnost p, že při hodu kostkou padne právě jednička. Výsledná pravděpodobnost P(3)= 0,0536 neboli 5,36%.

Tímto způsobem je tedy možné velice snadno a rychle propočítat šance na úspěch u kombinací založených na opakování čísel.

2.3.2 Složitější výpočty

I v tomto případě se můžeme setkat s úlohami, kde pouhé vložení do vzorce nestačí. Principiálně se však jedná o podobné případy, jako tomu bylo u pravděpodobnosti klasické. Na dva takové se nyní podíváme.

Prvním případem, nechť je rozšíření naší původní úlohy. Namísto hledání šance na hození tří jedniček pomocí šesti kostek nyní úlohu rozšíříme na pravděpodobnost hození jakékoliv trojce. V tomto případě je řešení jednoduché. Úloha prakticky obsahuje šest variant řešení, kdy pro každé hledáme pravděpodobnost 3 úspěšných pokusů z šesti. Totožně jako jsme úlohu spočítali pro tři jedničky, dopočítáme výsledek i pro tři dvojky, trojky, čtverky pětky a šestky a nakonec všechny tyto pravděpodobnosti sečteme (MATHWORD, 2016). A jelikož pravděpodobnost, že padne trojce jedniček, je totožná s tou, že padne trojce dvojek, trojek atd., stačí nám ji dokonce vypočítat pouze jednou a následně vynásobit šesti - P(3)*6.

Druhou úlohu rozšíříme o další možnosti úspěšnosti. Ptejme se nyní na pravděpodobnost, že při hodu šesti kostkami, nám padne číslo jedna alespoň třikrát. Jinými slovy, šance, že číslo jedna padne třikrát, nebo čtyřikrát, nebo pětkrát, nebo šestkrát. V tomto případě nám už pouhé vynásobení neposlouží. Namísto něj budeme muset naši rovnici spočítat hned čtyřikrát, abychom zastoupili každý možný počet úspěchů (MATHWORLD, 2016). Následně je opět sečteme a získáme tak celkový výsledek. Počítáme tedy P(3)+ P(4)+ P(5)+ P(6).

A co kdybychom hledali úspěch alespoň v jednom případě? To už bychom museli počítat binomickou rovnici hned šestkrát. Nebo můžeme využít znalosti o doplňku a využít jevu opačného, o kterém je pojednáno v následujících odstavcích.

2.4 Jev opačný

Naprosto nezávisle na způsobu výpočtu konkrétní pravděpodobnosti, můžeme s výhodou využít ještě jednoho poznatku. Z předchozích řádků již víme, že pravděpodobnost náhodného jevu dosahuje hodnoty v intervalu od 0 do 1. Při hodnotě 0 se bavíme o jevu nemožném, naopak při hodnotě 1 o jevu jistém. Proč je tomu tak?

Vraťme se opět k naší herní kostce a ptejme se nyní, jaká je pravděpodobnost, že na naší kostce padne číslo v intervalu od 1 do 6. Šance, že padne jednička je 1/6, stejně tak jako je tomu u dvojky, trojky atd. Výsledek je tedy P(A)= 1/6 + 1/6 + 1/6 + 1/6 + 1/6 + 1/6= 6/6 = 1. Zde vidíme, že zvolíme-li si jev jistý, matematicky vždy dosáhneme hodnosty 1. Jinými slovy, jev jistý je takový, který obsáhne veškeré možné výsledky. Toho lze mnohdy s výhodou využít.

2.4.1 Doplňkový jev

Doplňkový jev A' je doplňkem k jevu A, pokud obsahuje všechny možné výsledky s výjimkou těch, které jsou již obsaženy v jevu A (MATEMATIKA.CZ, 2016).

Řekněme, že známe pravděpodobnost, že na kostce padne hodnota 1 až 4. Ta je 2/3. Pokud víme, že součet všech pravděpodobností nám dá číslo 1, pak i víme, že do jevu jistého nám zbývá 1/3. Tato 1/3 případů je vlastně doplňkem do jedničky a zároveň pravděpodobností jevu opačného. A jaký je v našem případě jev opačný? Jsou to přirozeně všechny případy, kdy nepadlo číslo od 1 do 4. Nebo chcete-li případy, kdy padla pětka či šestka. Úlohu tak lze řešit ze dvou stran. Buď klasickým hledáním všech příznivých variant, nebo hledáním výsledků nepříznivých a následným odečtením od jevu jistého, jelikož jev jistý mínus doplněk nám opět vrátí pravděpodobnost hledaného jevu (1 – 1/3= 2/3). Této vlastnosti využijeme především v případech, kdy nalezení výsledku jevu opačného je pro nás výrazně snazší, oproti výpočtu běžným způsobem.

3 Hra V kostky

3.1 Výběr varianty

Tato kostková hra za desetiletí své existence zlidověla a tak se dnes můžeme setkat s celou řadou variant a modifikací. I v klasické podobě tak můžeme dnes nalézt několik verzí výkladu pravidel, nicméně rozdíly se zde již nachází především na poli drobných změn v bodovém hodnocení, či v možnostech dohazování kombinace „postupka“. Aplikace se tak zaměřuje na jeden konkrétní výklad pravidel, jež patří mezi ty nejrozšířenější.

3.1.1 Základní principy

V Kostky je hra pro dva a více hráčů, přičemž jejich počet není nikterak limitován. K samotnému hraní je užito šesti běžných šestistěnných kostek, jejichž strany obsahují číslice od 1 do 6. Hráči hází kostkami samostatně a nezávisle na ostatních. Hraje tedy vždy pouze jeden hráč v čase. Až poté co dohází jeden, předává kostky dalšímu v pořadí a ten obdobným způsobem odehraje své kolo.

3.1.2 Průběh kola

Na počátku daného kola, hráč hází všemi šesti kostkami zaráz. Následně si na základě padlých hodnot vybírá, které kostky odloží bokem a se kterými bude pokračovat v házení. Přitom lze odložit pouze kostky splňující některou z bodově ohodnocených kombinací. Odkládá si tak nejen kostky určité kombinace, ale také její bodové ohodnocení. Hráč není omezen na odložení pouze jedné kombinace. Padne-li mu bodových kombinací více, může si libovolně vybrat kterou, respektive které kombinace si ponechá. Vždy je však povinen si alespoň jednu bodovou kombinaci vybrat. Se zbylými kostkami pak může pokračovat v házení a snaze svůj bodový zisk dále navýšit. Postupně tak může dojít až k situaci, kdy si hráč odložil všech šest kostek. V takovém případě se všechny kostky opět vrací do hry a pokračuje se dále v házení. V tomto stavu ovšem, nejen že hráč může v házení dále pokračovat, ale dokonce má pravidly uloženou povinnost alespoň jeden takový hod provést. Tomuto stavu se říká „hra do plných“.

3.1.3 Konec kola

Hráč se může kdykoliv rozhodnout své házení ukončit. Jediné co přitom musí splnit, je dosažení minimální hranice bodů pro zápis. V našem případě je touto hranicí 350 bodů na kolo. Tj. od momentu, kdy hráč dosáhne v daném kole součtu odložených kombinací v hodnotě 350 bodů a více, může kdykoliv házení ukončit. Dosažené skóre kola se mu pak připíše k celkovému součtu a v házení pokračuje další

hráč. K druhému způsobu ukončení kola dochází, pokud hráči nepadne žádná bodová kombinace. V tomto případě kolo pro hráče končí a k celkovému skóre se mu připisuje nula bodů.

3.1.4 Cíl hry

Cílem každého hráče je dosáhnutí hranice 10 000 bodů v součtu ze všech kol. Ten, který tuto hranici překoná jako první, vyhrává. Některé herní pravidla pak umožňují zbylým hráčům, jež v daném kole ještě neházeli, pokusit se v tomto posledním kole hru také dokončit. Pokud se hranici 10 000 bodů podaří dohodit více hráčům, vyhrává ten, jehož celkové skóre je nejvyšší.

3.1.5 Bodově ohodnocené kombinace

Následující tabulka obsahuje příklady všech typů bodový kombinací.

název kombinace	příklad	body
padesát	5	50
sto	1	100
dvojce	XXYYZZ	
	226644	500
trojce	XXX	
	111	1000
	222	200
	333	300
	444	400
	555	500
	666	600
dvě trojce	XXXYYY	
	222555	200+500
	111333	1000+300
	…	
čtveřice	XXXX	
	1111	2000
	2222	400
	3333	600
	…	
pětice	XXXXX	
	11111	3000
	22222	600
	33333	900
	…	
šestice	XXXXXX	
	111111	4000
	222222	800
	333333	1200
	…	
postupka	123456	2000(1500)

Tabulka 1: Přehled bodových kombinací

- Kombinace dvojce platí pouze v případě, pokud padnou při hodu všemi šesti kostkami 3 dvojce stejných čísel zaráz.
- Pokud při hodu šesti kostkami došlo k situaci, že padly dvě rozdílné trojce (např. 333 a 555) zaráz, bodový zisk se sčítá, jakoby padly zvlášť (v našem případě 300 + 500 bodů)
- Padne-li hráči jedna z kombinací zmíněných v předchozích dvou bodech, hráč si nevybírá a automaticky hraje „do plných“
- Postupku lze získat pouze při hodu šesti kostkami
- Padne-li postupka hned (tzv. z ruky), připisuje se hráči 2000 bodů a pokračuje v hodu “do plných“. Hráči také může padnout neúplná postupka, kdy jedno číslo z řady chybí a naopak jiné se zde nachází dvakrát. Např. 123455, kde chybí číslo 6 a místo něj se v řadě nachází dvakrát 5. V takovém případě se hráč může pokusit postupku dohodit. Volí si tak jednu z kostek opakující se hodnoty (v našem případě je to jedna z kostek s hodnotou 5) a tou jednou hází. Pokud tímto hodem postupku dohodí, připisuje si 1500 bodů a pokračuje v házení „do plných“. Pokud ne, kolo pro něj končí s nulovým ziskem.
- Házení „do plných“ je hráč povinen vždy, když vyčerpá všech šest kostek. Přitom je jedno, zdali kostky postupně poodkládal v jednotlivých hodech, nebo mu padla kombinace/směs kombinací hned při hodu šesti kostkami. Takovým případem může být postupka z ruky (123456), ale i jakýkoliv mix kombinací (115333 sto + sto + padesát + trojce).

3.2 Souvislosti s teorií her

Teorie her je jednou z vědních disciplín aplikované informatiky (OSBORNE, 2004). Nosným artiklem je analýza rozhodovacích procesů v případech, kdy dochází ke střetu zájmů. Snahou je tyto procesy nejen analyzovat, ale také vyhledat optimální strategii (TADELIS, 2013). Taková strategie je v závislosti na situaci často kompromisním řešením, vyvažujícím obě strany “sporu“.

3.2.1 Aplikace

Klasická hra V Kostky není běžným případem pro teorii her. Pravidla hry totiž určují, že každý hráč hraje samostatně. Prakticky zde neexistuje žádný vztah vůči hrajícímu soupeři. Jelikož do aktu házení nemůže druhý hráč nikterak zasáhnout, odbourává se zde běžný střet zájmů. Výsledek kola také neovlivní soupeřovo skóre. Pouze může navýšit bodový zisk hrajícího a posunout ho tak blíže k sledovanému cíli. Stejně tak ani výše celkového skóre nezasahuje do cílů hráče druhého. Prakticky jediné, čím může být soupeř ovlivněn, je výše náskoku, jakou protivník disponuje. Pokud by byl tento náskok příliš vysoký, mohl by donutit protivníka k riskantnějším tahům. Tento efekt je však pouze psychologický a na mechaniku

hry nemá vliv. Větší riskování může přinést větší bodový zisk pouze ojediněle. Z pohledu statistiky a pravděpodobnosti se však takový postup jeví jako nevýhodný a kontraproduktivní. Při narůstajícím počtu pokusů je tedy rozhodující jen a pouze pravděpodobnost úspěchů konkrétních tahů.

Hráč tedy postupuje stejně, jako by hrál zcela bez soupeře. Rozhodující jsou pouze pravděpodobnosti a v rámci nich správné kombinování.

Přesto lze při hledání správného postupu aplikovat základní pravidla teorie her. Pro nalezení optimální strategie, je především nutné analyzovat veškeré možné stavy, pro budoucí výsledek házení (TADELIS, 2013). V souladu s teorií her, je tedy nutné dohledat veškeré tyto stavy a možnosti (strategie). Pokud bychom neměli kompletní přehled, nemohli bychom ani s jistotou prohlásit, že zvolená strategie je tou optimální. Čistě už jen z toho důvodu, že by se ona strategie nemusela v našem neúplném výčtu vyskytovat.

3.3 Důležité prvky hry

3.3.1 Co lze ovlivnit

Jak již bylo zmíněno, krom psychologického efektu hra prakticky neumožňuje jakkoliv ovlivnit soupeřův postup. Je tedy třeba ke hře přistupovat, jako kdyby byla ve variantě pro jednoho hráče. Jediným cílem pak je dosáhnout v každém kole maximálního bodového zisku. A proto, že nelze ani ovlivnit, jaká hodnota na kostkách padne, zužuje se počet nástrojů, jak ovlivnit výsledek hry na dva základní pilíře. Jedná se o správné rozhodnutí, kdy v daném kole házení ukončit a v případě pokračování pak správná volba odložených kombinací.

3.3.2 Kdy hru ukončit

Volba správného momentu, kdy hru ukončit, je jedním z klíčových elementů, jak dosáhnout dobrého výsledku. Při každém hodu totiž existuje nenulová pravděpodobnost, že na kostkách nepadne žádná bodová kombinace a tak hráč odejde s nulou. Tato pravděpodobnost se pak v závislosti na situaci liší a je jedním z podstatných faktorů, s nimiž je nutné kalkulovat. Obecně však lze konstatovat jedno základní pravidlo. Pokud snížíme počet kostek, sníží se i množství bodových kombinací, jež lze s nimi hodit. Zároveň se i snižuje množství všech kombinací (bodových i nebodových) jež může na kostkách padnout. Nicméně množství bodových kombinací se snižuje rychleji, nežli množství všech kombinací. A tak můžeme sledovat klesající trend pravděpodobnosti úspěšného hodu v závislosti na zmenšujícím se počtu kostek. Pokračovat ve hře s menším počtem kostek tak znamená vyšší riziko.

Dále je třeba do rozhodování zahrnout, jakého průběžného skóre již člověk dosáhl. Je logické, že pokud je toto skóre velmi nízké, vyplatí se více riskovat, nežli při velmi vysokých bodových ziscích. K nižším bodovým ziskům se totiž hráč dostane daleko častěji, nežli k těm vysokým. To je především dáno způsobem, kterým

jsou jednotlivé kombinace v pravidlech určeny. Kupříkladu pravděpodobnost, že při hodu šesti kostkami padnou právě tři šestky (kombinace za 600 bodů) je přibližně 5%, zatímco pravděpodobnost, že na kostkách padne jedna jednička (pouhých sto bodů) přesahuje 40%. Souvisí to především s faktem, že hodit jednu jedničku je možné daleko více způsoby, nežli například zmíněné tři šestky.

Veškeré poznatky je pak potřeba aplikovat současně. Například, nelze pouze říci, pokud budu mít 600 bodů, končím házení. Takový obecný postup by byl kontraproduktivní. Můžeme totiž stát v situaci, kdy máme naházeno již 600 bodů, ale protože jsme hráli „do plných“, můžeme ještě využít k následujícímu hodu až pět kostek. A jelikož pravděpodobnost úspěšného hodu v tuto chvíli přesahuje 90%, nemluvě o možnosti hodit kombinaci v hodnotě až 3000 bodů, je naopak více než vhodné v házení pokračovat.

3.3.3 Výběr vhodné kombinace

Jak již bylo výše zmíněno, po každém hodu si hráč musí odložit alespoň jednu bodovou kombinaci. Poměrně často však dochází k situaci, že má hráč na výběr z více variant. Např. po hodu šesti kostkami hráči padla následující čísla: 152224. V této řadě čísel můžeme identifikovat hned tři dílčí bodové kombinace: 1 (sto bodů), 5 (padesát bodů) a 222 (trojce hodnoty 200 bodů). Nejen, že hra umožňuje odložit si všechny kombinace zaráz a pokračovat tak v házení zbylou jednou kostkou, ale navíc je možné omezit výběr i na dvě kombinace, či si ponechat pouze jednu. V tomto konkrétním případě má hráč na výběr až ze sedmi možností, jak postoupit další tah. A jelikož v součtu již dosáhnul minimálního bodového zisku pro zápis - 350 bodů, může se navíc rozhodnout v házení nepokračovat a tento bodový zisk si již ponechat.

Z poznatků, získaných při rozhodování o ukončení hry, již víme, že menší množství kostek znamená nižší pravděpodobnost úspěchu. Tento poznatek je velice důležitý i při výběru odkládané kombinace. Ponecháme-li si z předchozího příkladu všechny tři kombinace (a s tím i 350 bodů), můžeme v házení pokračovat již pouze jednou kostkou. A jelikož jednou kostkou lze hodit pouze kombinaci 1 a 5, pravděpodobnost úspěchu je zde 2/6, tedy přibližně 33%. Naopak zvolíme-li si pouze jedničku (a sní pouhých 100 bodů), házíme dále pěti kostkami s úspěšností přes 90%. Navíc čím více kostek, tím lepších bodových kombinací můžeme dosáhnout. V případě jedné je nejlepší kombinací jednička, tedy sto bodů. V případě pěti kostek, je to pět jedniček v hodnotě 3000 bodů. Je tedy zřejmé, že čím méně kostek použijeme, tím více riskujeme, pro menší bodové zisky.

Při výběru vhodné kombinace však musíme brát i zřetel, o jaké skóre takovým tahem přicházíme. V našem modelovém příkladě, kdy padly čísla 152224, máme celkově 350 bodů. Pokud bychom vybrali pouze jedničku, ponecháváme si tak sto bodů a zbylých 250 zahazujeme. Těchto 250 bodů prakticky vyměníme za větší pravděpodobnost úspěšného hodu a za možnost získat lepší potenciální zisk (jelikož budeme házet se čtyřmi kostkami navíc). Vždy je nutné toto „zahozené“ skóre brát v úvahu, jelikož někdy může být natolik vysoké, že se nám jeho výměna za

vyšší počet kostek již nevyplatí. Náš modelový příklad obsahuje kombinaci trojce 222 o hodnotě dvě stě bodů. Pokud vezmeme v úvahu, že na pouhých dvě stě bodů spotřebujeme hned 3 kostky, jeví se tato kombinace jako velice slabá. Takového bodového zisku lze velmi snadno docílit např. hodem dvou jedniček, jež má vysokou pravděpodobnost úspěchu a navíc tak získáme kostku navíc. Je tedy vhodnější ponechat si pouhých sto a zbylých dvě stě zahodit. Snadno ale může nastat situace jiná. Pozměňme nyní číselnou řadu na následující 155542. Opět zde nacházíme jedno kostkovou kombinaci hodnoty 100 a k tomu trojici o hodnotě 500 bodů. V takovém případě při výběru pouhé jedničky zahazujeme již 500 bodů. A taková bodová ztráta již může být natolik citelná, že se spíše vyplatí ponechat si oněch pět set a pokračovat v házení „jen“ se třemi kostkami, místo pěti. Šance na hození oněch pětiset bodů je totiž natolik nízká, že se nevyplatí takový bodový zisk zahodit.

4 Metodický postup

4.1 Výběr programovacího jazyka

Pokud již nebyl stanoven, je potřeba brát zřetel na několik faktorů. Především podporu objektově orientovaného programování a dostatečnou funkcionalitu. Programovací jazyk pak dále musí umožňovat tvorbu aplikací pro cílový operační systém a měl by taktéž disponovat dostatečnou dokumentací.

4.2 Výběr vhodného vývojového prostředí

Prostředí musí především podporovat zvolený programovací jazyk a potřebnou funkcionalitu. Dále je vhodné zohlednit pořizovací cenu, osobní preference a také uživatelský komfort voleného vývojového prostředí.

4.3 Návrh aplikace

4.3.1 Návrh konceptu

Získání obecné představy o konstrukci aplikace. Na jaké logické části bude rozdělena, jakým způsobem bude problém dekomponován na jednotlivé funkční celky.

4.3.2 Detailní návrh tříd a vztahů

Na základě hlubší analýzy sestavit konkrétní návrh, jehož výstupem bude diagram tříd, vhodný pro implementaci samotné aplikace. Důraz je kladen především na kompletní vytyčení jednotlivých tříd, užité datové struktury a celkové logické provázání jednotlivých tříd a datových struktur.

4.4 Tvorba aplikace

4.4.1 Vytvoření logické kostry aplikace (interface)

Založení všech potřebných tříd, v případě C++ vyplnění jednotlivých hlavičkových souborů. Definování datových struktur, dle připraveného návrhu.

4.4.2 Implementace metod

Postupná implementace jednotlivých tříd. V případě potřeby, vytvoření pomocných metod a datových struktur. Při implementaci jednotlivých tříd dodržovat základy koncepce objektově orientovaného programování (jako je abstrakce, zapouzdření, atd.). Při implementaci je také vhodné zohlednit znovu použitelnost jednotlivých tříd v budoucích aplikacích podobného typu.

4.4.3 Tvorba grafického rozhraní GUI

Navržení vhodného vzhledu aplikace a intuitivního ovládání v rámci zvolené technologie vizualizace (textové vs. grafické prostředí).

4.5 Testování a opravy

- Otestování správné funkcionality metod jednotlivých tříd. Testování aplikace jako celku v reálných podmínkách.

- Opravy nalezených chyb a případná optimalizace výkonu.

4.6 Závěrečné hodnocení

Zhodnocení dosažení stanovených požadavků, použitelnosti a zvážení budoucích rozšíření či vylepšení.

5 Navrhované řešení

5.1 Návrh konceptu

Již při návrhu konceptu je vhodné uvažovat technologie, na kterých bude aplikace vystavěna. Jedním z požadavků zadání, je tvorba programu pomocí programovacího jazyka C++. Ten ovšem umožňuje zvolit více přístupů. Jak procedurální programování, tak třeba i objektově orientované (dále OOP), nebo generické. Vzhledem k povaze aplikace a snaze rozdělit jednotlivé části na logické celky, se z těchto přístupů jeví jako nejvhodnější přístup objektově orientovaný.

Principem OOP je vytvoření jednotlivých tříd, zapouzdřujících vždy konkrétní logiku a data, která spolu souvisí a lze na takový celek pohlížet jako na samostatně fungující entitu.

Ve vyvíjené aplikaci můžeme sledovat hned několik základních okruhů, které je třeba za běhu řešit a jež by mohly naplňovat požadavky na vytvoření vlastní třídy.

V navrhovaném konceptu byly identifikovány požadavky na seskupení následujících funkcionalit do jednotlivých tříd:

- analýza generovaných dat
- statistické funkce
- simulace šestistěnné kostky
- seskupení informací o hráčích a průběhu hry
- řešení vstupů a výstupů dat
- režijní řízení aplikace

5.1.1 Řídící element

Krom výše zmíněných okruhů, je více než vhodné pracovat i s třídou, řídící běh celé aplikace. Tato třída by měla propojovat veškeré nosné prvky programu. Jedním ze základních požadavků na řídící třídu (někdy označovanou jako „engine“) je schopnost řízení běhu aplikace. Při inicializaci tato třída nastavuje veškeré výchozí hodnoty. Zavádí instance důležitých tříd. Provádí řídící cykly a za pomoci volání metod nad jednotlivými třídami, realizuje potřebné operace. Naopak při ukončení programu, má za úkol veškeré probíhající úlohy řádně ukončit a zajistit správné uvolnění užívané operační paměti počítače. Její metody jsou tak spíše režijního rázu.

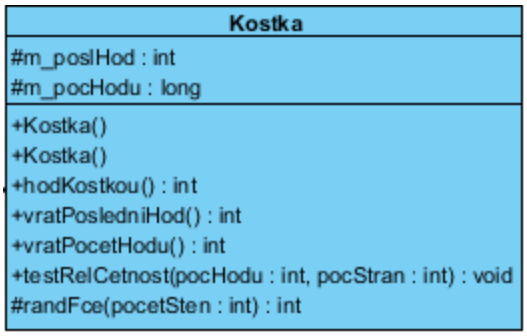
Dalším důležitým úkolem této třídy je předávání dat. Jelikož tento spojovací prvek za běhu užívá různých tříd k vykonání různých úkolů – tzv. delegování, mnohdy se neobejde bez předávání dat mezi jednotlivými třídami. S touto nutností je třeba počítat nejen při vhodném návrhu řídící třídy jakožto prostředníka, ale i vhodnou datovou strukturou a užitím metod u tříd ostatních v rámci pozdějších fází návrhu.

5.2 Důležité třídy

Jak již bylo řečeno, celý návrh aplikace, byl realizován jakožto objektově orientovaný. Jednotlivé úkony proto byly rozdělený do tříd k tomu určených a implementovány samostatně. Při tvorbě návrhu byl kladen důraz na hlavní pilíře OOP. Především pak na abstrakci, zapouzdření a, pokud to bylo možné, i možnosti znuvupoužitelnosti dané třídy. Na následujících řádcích se proto na ty nejdůležitější podíváme. Zjistíme především, k čemu která třída slouží, co obsahuje a jakou úlohu a místo v hierarchii aplikace plní.

Obrázek 1: Diagram tříd

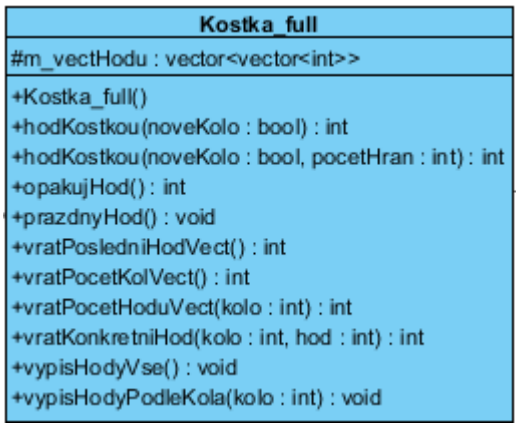
5.2.1 Kostka



Obrázek 2: Třída Kostka

Jedná se o základní třídu určenou čistě k házení kostkou. Jejím nosným prvkem je metoda využívající funkce rand() obsaženou ve standardních knihovnách jazyka C++ (CPLUSPLUS.COM, 2016). Ta umožňuje generovat pseudonáhodná čísla (ITNETWORK.CZ, 2016), v našem případě omezená na interval od jedné do šesti. Díky jednoduchému zapouzdření je v případě potřeby možné definovat i vlastní implementaci random funkce, bez nutnosti dalších dodatečných úprav aplikace. Doplňkově si pak třída uchovává hodnotu posledního hodu, počet všech hodů za životnost dané entity, a disponuje i možností otestovat „kvalitu“ užité náhodné funkce.

5.2.2 Kostka_Full

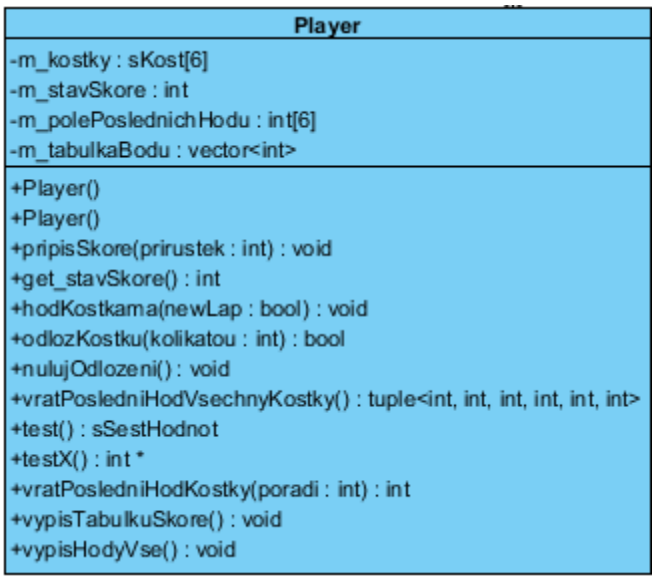


Obrázek 3: Třída Kostka_Full

Tato třída je přímým dědicem z obecné třídy Kostka. Zavádí potřebné datové struktury a metody, navrhnuté pro potřeby aplikace. Jedná se tedy o konečnou specializaci třídy Kostka.

Třída především obsahuje datovou strukturu, umožňující ukládat více informací, ohledně uskutečněných hodů. Jednotlivé hody jsou tak logicky roztříděny, nejprve podle kola, ve kterém byly učiněny, následně pak podle pořadí (pokud bylo kostkou házeno v daném kole opakovaně), ve kterém padly. Lze tak v rámci analýzy sledovat přesnou historii všech hodů. Doplněny jsou pak i potřebné režijní metody.

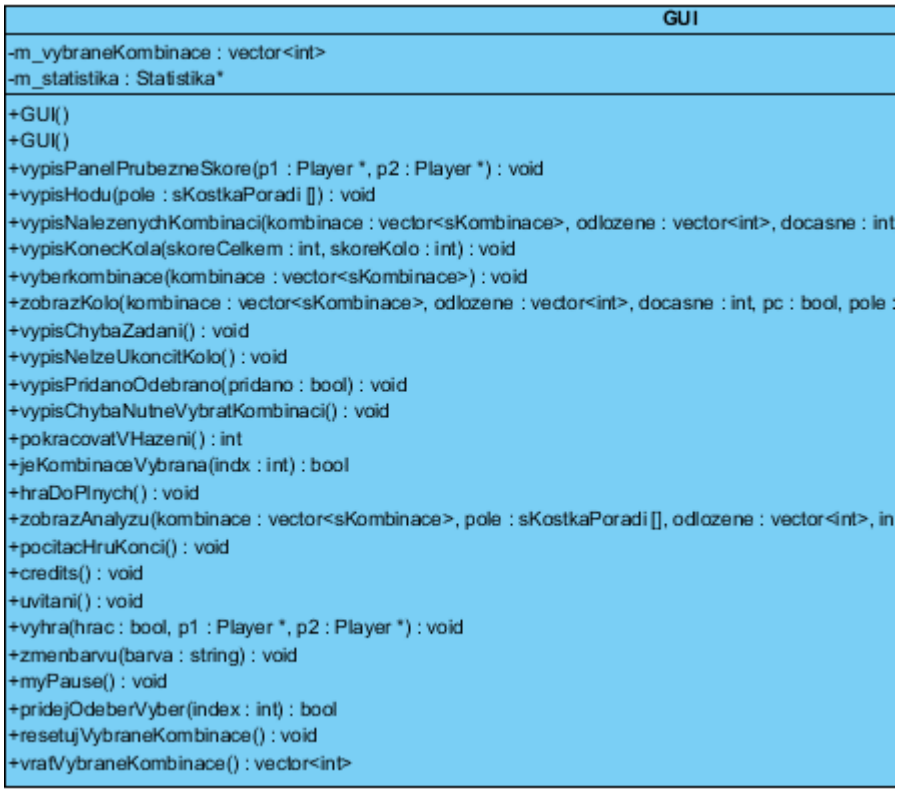
5.2.3 Player



Obrázek 4: Třída Player

Jedná se o třídu zastřešující informace a vlastnosti hráče. Uchovává informace o průběhu hry pro danou osobu/počítač. Dále vlastní šest instancí třídy Kostka_Full se kterými provádí veškeré hody. Automaticky spravuje nejen odkládání jednotlivých kostek, ale i hromadné házení, vracení kostek do hry a další režijní metody určené pro házením kostkami. A nakonec i zde nalezneme metody, týkající se analytických výpisů, souvisejících s průběhem hry, historií hodů atd.

5.2.4 GUI

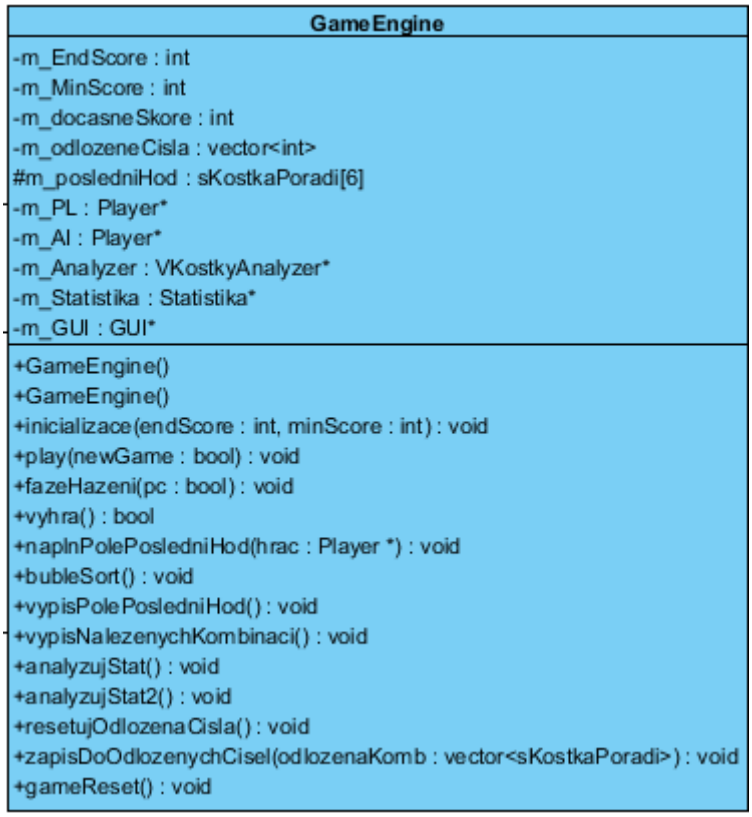


Obrázek 5: Třída GUI

Pro každou interaktivní aplikaci je nutné zajistit možnosti vhodných vstupů a výstupů informací. K tomuto účelu navrhovanému programu slouží právě třída s názvem GUI. Třída prakticky sdružuje veškeré výpisy, tabulky a všemožná hlášení, nutné komunikovat s uživatelem aplikace. Stejně tak obsahuje dialogová okna, umožňující zadávání požadovaných dat do programu.

Ačkoliv je zvolené prostředí aplikace pouze textové, jelikož aplikace běží v konzolovém režimu, je vhodné veškeré výpisy ucelit právě zde. Díky volbě této třídy se nejen samotný kód aplikace stává přehlednějším, ale také je možné unifikovat jednotlivé výpisy, formátovat tabulky, řídit barevné škály atd.

5.2.5 GameEngine



Obrázek 6: Třída GameEngine

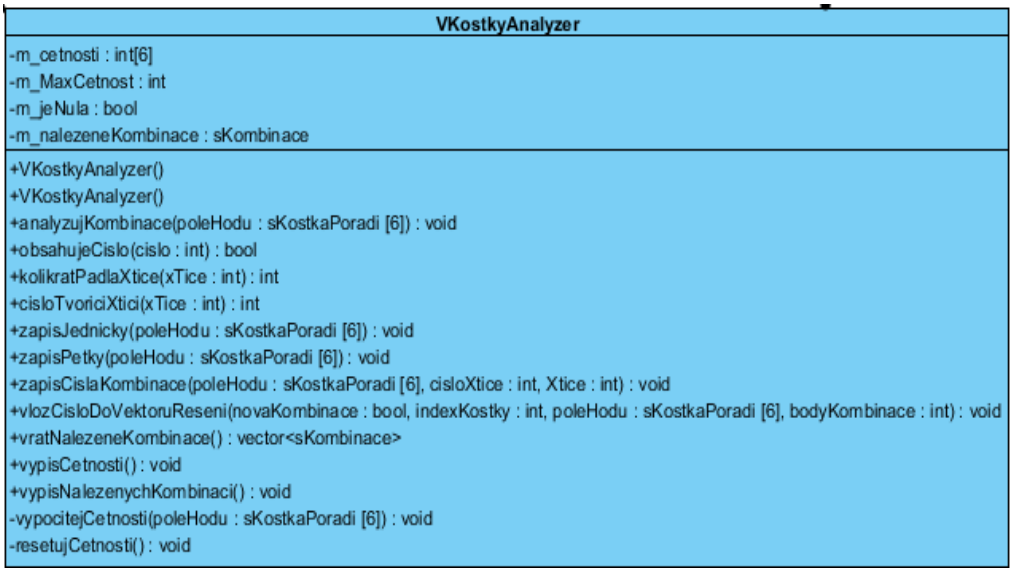
Jak již název napovídá, tato třída má na starosti samotný běh hry. Jedná se o prostředníka celé aplikace, spojujícího jednotlivé části dohromady.

Uchovává si instance třídy hráč, na kterých dle potřeby vyvolává jednotlivé metody týkající se hodů, stejně jako odkládání kostek, či připisování bodových zisků. Dále si od těchto tříd přebírá data týkající se posledního hodu a předává je k dalšímu zpracování.

V neposlední řadě pak GameEngine obsahuje instance tříd, důležitých pro následnou analýzu dat a určení dalšího postupu. Jedná se o třídy VKostkyAnalyzer a Statistika.

Důležitou součástí této třídy jsou i metody, řídící průběh celé hry. Právě zde je definován průběh každého kola. GameEngine na základě zpracovaných dat hlídá, zdali může hráč pokračovat v házení, zdali vybral povolenou kombinaci, či jestli může dosažené skóre připsat. Dále je zde řízeno střídání jednotlivých hráčů, včetně předání hry počítači. A nakonec GameEngine také sleduje, zdali již některý hráč nedosáhl cílového skóre a tak i výhry.

5.2.6 VKostkyAnalyzer

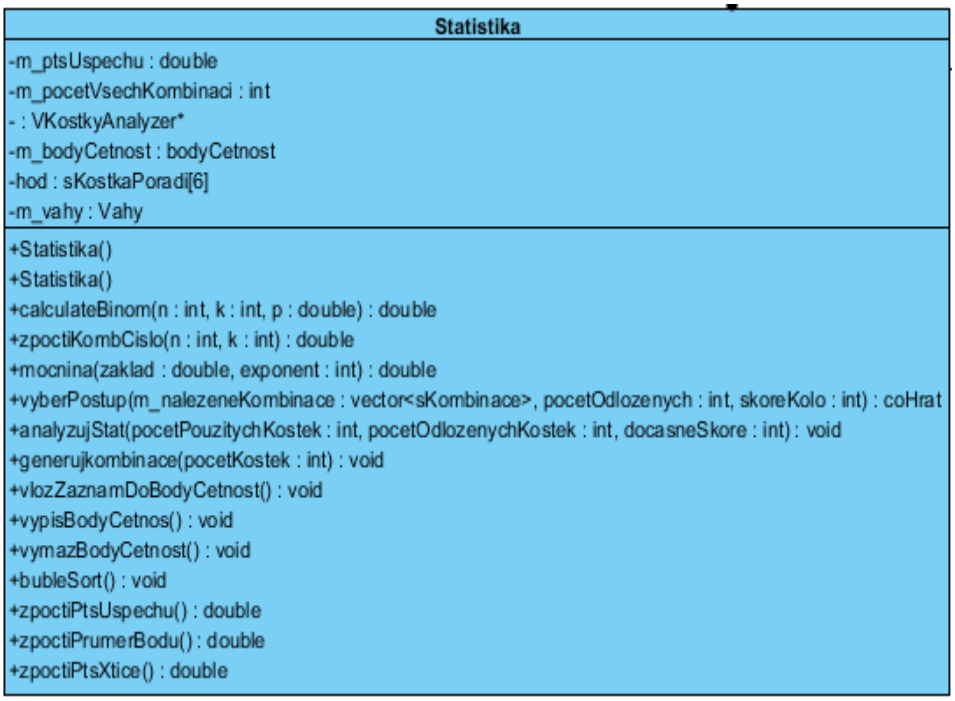


Obrázek 7: Třída VKostkyAnalyzer

Zde se již dostáváme k samotné analýze dat. Když proběhne hod kostkami, jsou informace předány právě instanci této třídy. Ta má za úkol vyhledávat bodově ohodnocené kombinace v souladu s pravidly hry V kostky. Analyzér postupně prochází hodnoty padlé na kostkách a identifikuje veškeré bodové varianty, jež je možné vybrat. Tato data následně ukládá do složitějších struktur a předává dále.

Jedná se zde o jeden z nejdůležitějších pilířů aplikace, jelikož generuje informace pro celou informaci. Předává data GameEnginu, který je využívá pro řízení hry. Poskytuje hráči výpisy možností, které lze dále zahrát. Je také využíván ve statistickém modulu. Ten využívá VKostkyAnalyzer při hledání potenciálních kombinací pro následující hody a s využitím těchto dat provádí pravděpodobnostní výpočty. A v neposlední řadě je tato třída hlavní oporou pro automatické rozhodování počítače.

5.2.7 Statistika



Obrázek 8: Třída Statistika

Posledním v přehledu nejdůležitějších tříd je Statistika. Ta zajišťuje poslední důležitou část aplikace a to je výpočet potřebných pravděpodobností a zkompletování informací pro hru počítače. K tomuto účelu má implementováno několik vlastních metod, týkajících se pravděpodobnostních a statistických výpočtů jako je například binomická rovnice, výpočet kombinačního čísla, aritmetického průměru apod.

Třída statistika tedy zužitkovává veškeré získané informace z předchozích analýz a následně dopočítává zbytek. Díky tomu se po výpočtech realizovaných zde, nacházíme v bodě, kdy máme již veškeré potřebné podklady jak pro hráče, tak i pro počítač. Z toho důvodu je Statistika obohacena ještě o konečné metody, sloužící počítači k rozhodnutí o dalším postupu. Statistika tak dokáže rovnou odpovědět, zdali je výhodné hrát dále a pokud ano, jakou kombinaci si ponechat, jakou zahodit a proč tak učinit.

5.3 Funkcionalita

Než se podíváme přímo na vzhled aplikace, představme si některé techniky, užívané aplikací, v rámci automatického rozhodování. Především se bude jednat o techniky rozpoznávání jednotlivých kombinací, obsažených v hodu více kostek a dále pak některé principy užité při rozhodování o dalším postupu počítače.

5.3.1 Rozpoznání kombinací

Jak již bylo v předchozích odstavcích nastíněno, tuto úlohu má na starosti třída VKostkyAnalyzer. Ačkoliv se nám může zdát rozpoznání nějaké trojce, jedničky, či třeba postupky relativně banální, v řeči programového kódu může vyvstat množství problému. Už jenom proto, že typů kombinací a všemožných variant, v jakém pořadí nám která čísla na kostkách padnou je velké množství a implementovat obecný postup na poměrně nesourodou skupinu dat může být dosti komplikované.

Vstupní data jsou proto ještě před příchodem do naší třídy předpřipravena. GameEngine je seřadí podle padlé hodnoty do vzestupného pořadí a uloží tak, aby bylo později možné opět identifikovat, která hodnota se vztahuje ke které kostce. To je zajištěno pomocí pole struktur “sKostkaPoradi“.

V první fázi jsou hledány četnosti jednotlivých hodnot. Vzniká tak pole reprezentující hodnoty kostky od 1 do 6, kde pro každou hodnotu je uveden záznam, kolikrát se v daném hodu nacházela. Následně je v tomto poli nalezena nejvyšší četnost.

V druhé fázi je postoupeno k příslušné analýze na základě této maximální četnosti. Například, pokud je nalezená maximální četnost 4, můžeme s jistotou říci, že náš hod obsahuje kombinaci čtveřice. Nyní stačí v poli daného hodu nalézt první číslo tvořící čtveřici a to, plus další tři následující zapsat jako příslušnou kombinaci (s výhodou využíváme seřazení kostek v poli). Nesmíme ale zapomínat, že přípustnou kombinací je i samotná jednička (sto bodů) a pětka (padesát bodů). Ověříme tedy, zda-li je počet hozených kostek větší než 4 a pokud ano, cyklicky dohledáme zbylé pětky a jedničky. Zároveň ale nesmíme zapomenout, že i samotná čtveřice může být tvořena čísly jedna či pět a v takovém případě již daná čísla samostatně nezapisujeme.

Pro každý případ maximální četnosti je pak realizován speciální případ zpracování. Některé jsou velmi jednoduché – četnost 6 značí jedinou kombinaci šesti stejných kostek, četnost jedna (za podmínky hodu šesti kostkami) pak značí pouze a jedině kombinaci postupka. Maximální četnost jedna totiž značí, že se každé číslo zde nachází právě jednou. Kdyby se některé nacházelo třeba i jen dvakrát, maximální četnost by se pak již rovnala dvěma.

Vždy je však třeba myslet na všechny varianty, které mohou nastat. Maximální četnost tři nám sice značí kombinaci trojce, ale je třeba i myslet na možnost, že bylo hozeno šesti kostkami a padly v jednom hodu trojce dvě. A nebo také ne. Pak je

třeba hledat i dodatečné jedničky a pětky. Ovšem pouze pokud již netvoří samotnou trojici.

Pokud ovšem důsledně pohlídáme tyto dílčí pod varianty, jeví se třídění kombinací na základě nalezených četností, jako velmi efektivní nástroj.

5.3.2 Předání dat

Způsobů, jakým je možné předat analyzovaná data, existuje jistě spousta. Může to být například pomocí proudů, ukládáním do souboru a následným načítáním, či tvorba různých vlastních datových struktur.

Pro potřebu aplikace byly užity kontejnery typu vektor, jež jsou obsaženy v rámci standardní knihovny C++ a zkombinovány s vlastními datovými strukturami. Snahou bylo spojit komplexně data do datově efektivní struktury, umožňující jednotlivé kombinace sekvenčně, či za pomocí indexů procházet a získávat si další potřebná data.

Výsledný systém se zakládá na vektoru (dynamické pole), obsahující vlastní strukturu. Tato struktura uchovává data o bodech kombinace a dále vektor, sloužící k zápisu jednotlivých kostek, tvořících danou kombinaci (pro každou kostku se ukládá její hodnota a původní pořadí).

Logické uspořádání tedy vypadá tak, že každá položka nadřazeného vektoru představuje jednu nalezenou kombinaci a každá složka obsaženého vektoru pak představuje informaci o jedné z kostek tvořících tuto kombinaci.

Tímto způsobem tedy můžeme jednoduše získat přehled o množství nalezených kombinací, jejich bodové ohodnocení, počty kostek tvořících dílčí kombinace a jejich konkrétní umístění. Máme tedy veškerá potřebná data na jednom místě s možností snadného přístupu ke kterékoliv položce.

5.3.3 Statistické prvky

Ve chvíli, kdy jsou data roztříděna, je možné nad nimi provádět pravděpodobnostní výpočty. Ty se nám budou hodit především pro rozhodování počítače.

Výpočty tohoto typu provádí instance třídy Statistika. Té je krom našeho výpisu kombinací předáno také dočasné skóre a počet již odložených kostek. O samotné rozhodnutí, jak hrát dál se stará metoda “vyberPostup“, jež vrací předem stanovený kód a informaci, zdali je vhodné dále pokračovat v házení, či nikoliv.

Metoda si dělí situaci na následující stavy:

- Není možné ukončit házení (nebylo dosaženo alespoň 350 bodů)
- Je možné zapisovat a průběžné skóre je menší, než střední váha
- Je možné zapisovat a průběžné skóre je větší, než střední váha
- Je možné zapisovat a průběžné skóre je větší, než horní váha

Zmiňované váhy, jsou hraniční hodnoty skóre, které byly stanoveny na základě průběžného testování. Jejich hodnoty jsou 600 bodů pro střední váhu a 1000 bodů pro horní. Tyto váhy jsou důležité především z toho důvodu, abychom mohli posoudit míru rizika, spojenou s dalším házením.

Jak již bylo řečeno v kapitole o hře V Kostky, čím vyšší průběžné skóre máme, tím větší bodovou ztrátu při následujícím hodu riskujeme. Navíc, dosažení vyšších bodových zisků má nižší pravděpodobnost, než li u bodů nízkých. Jednoduše, velké body nepadají tak často, jako ty malé.

Rozdělení do tří kategorií se nakonec jevilo jako ideální rozčlenění. Lze tak poměrně dobře a dostatečně hustě roztřídit možné bodové stavy a přiřadit k nim odpovídající míru rizika. Ta je pro bodový zisk dané výše ještě stále přijatelná a dlouhodobě výnosná.

Proč vlastně riskovat a nehrát na jistotu? Jednoduše proto, že samotná hra tuto možnost už z principu neumožňuje. I při počátečním hodu šesti kostkami, existuje nenulová pravděpodobnost, že na kostkách nepadne žádná bodová kombinace. A stejně je tomu i v případech ostatních. Snahou tedy není dosáhnout nulového rizika, ale nalézt ideální poměr mezi potenciálním ziskem, a rizikem konkrétní bodové ztráty.

Naše rozhodující metoda nedělá nic jiného. V rámci každého hodu a pro každou kombinaci zjišťuje, kolik kostek ji tvoří a kolik jich pro další hru zbývá, ověřuje možná skóre a provádí potřebná porovnání. Následně počítá pravděpodobnost úspěchu dalšího hodu. K tomuto výpočtu užívá generátoru kombinací, který dokáže určit veškeré možné kombinace (bodové i nebodové), jež mohou v následujícím hodu padnout a stejně tak z nich (s pomocí analytické třídy) sečíst i počet kombinací bodových. A jelikož již víme, že pokud víme počet všech příznivých výsledků a počet všech možných výsledků, lze jednoduchým podílem tuto pravděpodobnost dopočítat.

Statistická třída se samozřejmě nezaměřuje pouze na rozhodovací činnost, ale obsahuje i vlastní implementace metod pro dílčí statistické výpočty. Příkladem může být např. výpočet binomického vzorce, vyčíslení mocniny či kombinační čísla. S pomocí všech nástrojů pak podává jak podklady pro rozhodnutí počítače, tak i vizuální analýzy, které může sledovat hráč při běhu hry.

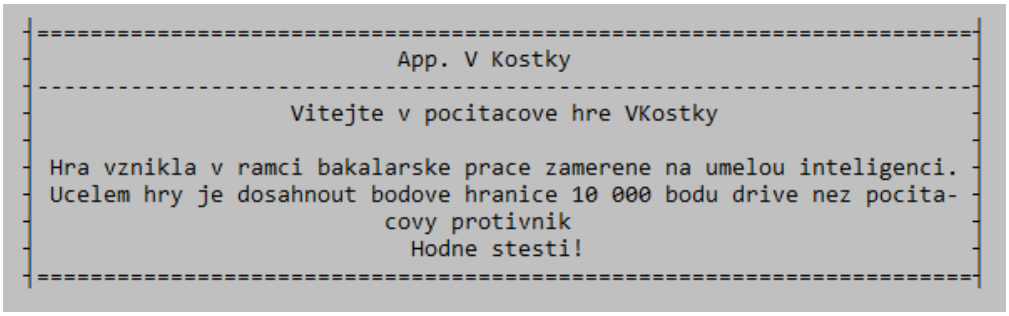
6 Výsledná aplikace

Celá hra byla realizována, jakožto konzolová aplikace. Její grafické pojetí je tedy zobrazováno v textovém režimu. Ovládání aplikace je pak zajištěno pomocí klávesnice, přičemž samotná volba je realizována volbou číslice příslušné nabídky a jejím potvrzením pomocí klávesy enter. Některá dialogová okna pak očekávají stisk libovolné klávesy, pro další pokračování.

Hra je konstruována jako souboj dvou hráčů, z nichž jeden je hráč lidský, druhého zastupuje počítač. Hraje se klasická hra V Kostky, v souladu s pravidly popsanými v kapitole 3. Během hraní je možné sledovat množství statistických údajů, včetně průběhu hry protivníka, přičemž proces analýz a výpisů je zcela automatizován. Ve hře také není možné žádným způsobem podvádět, což je zajištěno mimo jiné integrovaným modulem simulujícím házení kostek. Podmínky jsou tedy pro oba hráče stejné.

6.1 Start aplikace

Program je realizován, jakožto spustitelný „exe“ soubor. Po jeho načtení, je inicializován GameEngine, který provede úvodní zavedení všech potřebných částí aplikace. Vytvoří potřebné instance tříd, nastaví výchozí hodnoty. Následně je pomocí třídy GUI zobrazena úvodní obrazovka vítající hráče do hry.



Obrázek 9: Úvodní obrazovka

Po potvrzení úvodní obrazovky již začíná samotná hra. Prvním házejícím v pořadí je lidský hráč.

6.2 Fáze házení

Házení kostkami je opět automatické. Každý hráč má k tomuto účelu šest vlastních instancí třídy Kostka_Full. Každá kostka si uchovává hierarchickou strukturu historie hodů pro daného hráče, čehož lze využít v rámci analytických výpisů. Pro každou kostku se také uchovává informace, zdali je odložena, nebo je možné s ní dále házet. Tato kontrola, stejně jako následný hod probíhá opět automaticky. Tím je také zajištěna ochrana proti případnému podvádění. Hráč si navíc nemusí hlídat, či složitě volit, se kterými kostkami bude dále házet. Pouze si vybere, které kostky odloží a zbytek již provede samotná aplikace.

6.2.1 Vizualizace hodu

Po uskutečněném hodu, který proběhl na pozadí, je hráči zobrazena tabulka, obsahující základní informace o výsledku hodu.

```
================================================================================
|       Skore hrac: 1100                                  Skore PC: 0          |
================================================================================
|                                                                              |
|                              Hod: 1 1 3 5 5 5                                |
|                                                                              |
================================================================================
|                                                                              |
|                                      body    pts           volba             |
|       Padle kombinace: 1) 5 5 5      500     0.722222                        |
|                        2) 1          100     0.922711                        |
|                        3) 1          100     0.922711                        |
|                        ---------------------                                 |
|                            Soucet hodu: 700                                  |
|                                                                              |
|                         Celkem za kolo: 700                                  |
|                                                                              |
|                              Odlozene:                                       |
================================================================================
================================================================================
|                                                                              |
|                              Vyber kombinaci                                 |
|                              ---------------                                 |
| Zadejte cislo kombinace, kterou chcete pridat/odebrat a stisknete ENTER.     |
| Pro ukonceni zadavani zadejte cislo "0".                                     |
|                                                                              |
--------------------------------------------------------------------------------
  Vyber:
```

Obrázek 10: Vizuální zpracování informací uskutečněného hodu

Samotné okno je složeno z několika nezávislých výpisů. Tím je zajištěno jednoduché skládání různých výpisů dohromady a také snadnou editaci jednotlivých tabulek.

Horní část obrazovky obsahuje panel, zobrazující aktuální stav skóre obou hráčů. Toto skóre je sumou všech zapsaných výsledků, daného hráče. Můžeme tak snadno sledovat, jak daleko se momentálně nacházíme od výherní hodnoty 10 000 bodů.

V druhé části okna můžeme sledovat výčet hodnot, které nám na kostkách padly. Výpis hozených kostek je pro usnadnění seřazen vzestupně, aby byla orientace v padlých kombinacích pro hráče co nejsnazší.

Třetí část okna je analytická. Zde jsou hráči představeny veškeré bodové kombinace, jež v aktuálním hodu padly. Pro každou kombinaci pak můžeme vyčíst, z kterých kostek (hodnot) se skládá a jaký bodový zisk představuje. Hráči je také pro každou kombinaci vypočítána pravděpodobnost, s jakou uspěje v dalším hodu, jestliže si vybere danou kombinaci.

Další položkou je Součet hodu, jenž reprezentuje sumu bodů ze všech vypsaných kombinací. Následný řádek „Celkem za kolo“ představuje průběžný počet bodů získaných v daném kole, včetně součtu bodů, z posledního hodu. Hráč se tak na základě zmiňované položky rozhoduje, zdali již dosáhl dostatečného počtu bodů, nebo bude muset ve svém snažení pokračovat.

Pro ucelení informací o průběhu kola pak panel obsahuje výpis kostek, jež hráč v průběhu kola odložil.

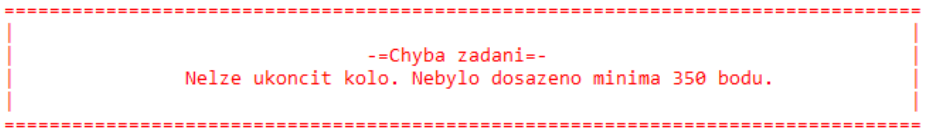
Posledním panelem tohoto okna je Výběr kombinací. Ten zobrazuje informace o způsobu výběru kombinace a pole očekávající vstupní hodnotu. Výběr kombinace probíhá na základě jejího pořadového čísla ve výpise padlých kombinací. Po správném zadání je hráči volba potvrzena a její výběr je indikován zobrazením šipky, u příslušné kombinace. Opětovnou volbou této kombinace, je z výběru odebrána.

```
                                body
Padle kombinace: 1) 1           100   <==
                 2) 5           50
                 ---------------------
                     Soucet hodu: 150
```

Obrázek 11: Indikace vybrané kombinace

Jelikož se jedná o textové zadávání, aplikace hlídá uživatele v korektním zadání dat. V případě chyby je uživatel vyzván ke správnému zadání volby.

V této fázi se také hráč rozhoduje, jestli bude v házení pokračovat. Samotná volba je automaticky potlačena pouze v případě „hodu do plných“, kdy je proveden další ho se šesti kostkami, nebo pokud hráč nezískal žádnou bodovou kombinaci. V ostatních případech si postup volí sám. Opět je nutné tuto volbu hlídat a program si v této fázi vynucuje dodržování důležitých pravidel hry. Jedná se například o situaci, kdy chce hráč pokračovat v hodu, bez odložení alespoň jedné kombinace, nebo chce ukončit kolo, aniž by dosáhl minimálního skóre pro zápis.



Obrázek 12: Chybové hlášení

6.3 Konec házení

Konec kola, probíhá v souladu s pravidly hry. Může k němu dojít v případě, že hráč již dosáhl minimálního skóre kola 350 bodů a rozhodl se v házení dále nepokračovat, anebo pokud mu v aktuálním hodu nepadla žádná bodová kombinace.

Poté co kolo skončí, je hráči zobrazena informace o bodovém zisku tohoto kola a o aktuální výši celkového skóre.

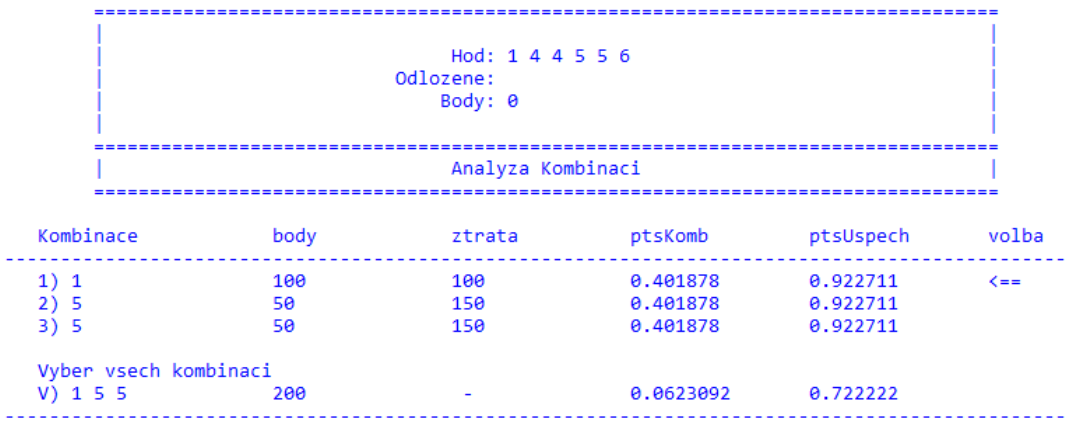
Hra následně hráči zapíše příslušný bodový zisk a předává hru počítači.

6.4 Hra počítače

Při hře počítače, je celý průběh kola zcela automatizován a hráč do něj nemůže nikterak zasáhnout. Postup počítače je však pro hráče viditelný a může jej tak sledovat po celý průběh kola. Zobrazované informace v každé fázi hodu, počínaje informacemi od hodu, bodovém skóre, odložených kostkách apod. jsou totožné jako v případě hry hráče. Pro přehlednost je však zavedeno jiné barevné schéma.

6.4.1 Počítačová analýza

Zatímco se při hře hráč spoléhá na svůj úsudek či případné zkušenosti, počítač postupuje na základě pravidel a výpočtů. Mezi ně patří například různé bodové výše, či výpočty pravděpodobností. Z toho důvodu je hra počítače obohacena o analytický výpis popisující některé důležité vlastnosti nabízených kombinací. Po standardním výpisu daného hodu, tak následuje obrazovka, kde si může hráč závěry počítače prohlédnout.



Obrázek 13: Zobrazení statistických informací

V nové tabulce můžeme opět sledovat některé souhrnné informace. Již standardně seřazené hodnoty posledního hodu, dále seznam odložených kostek a průběžnou výši bodů v daném kole. Pod těmito informacemi nalezneme tabulku s podrobným výpisem volených kombinací. Každý řádek tabulky začíná pořadovým číslem a výpisem hodnot tvořících tuto kombinaci. Následuje bodové ohodnocení a také body ztráty. A posledními položkami jsou sloupce pravděpodobností a indikátor volby.

Sloupec ztráta zobrazuje, jak velké bodové ztráty bychom dosáhli, pokud by naší volbou byla kombinace daného řádku. Číslo je tedy součtem bodů všech ostatních kombinací, kterými místo odložení budeme házet.

Dalšími čísly obsaženými v tabulce jsou pravděpodobnosti. První číslo nám udává, jaká byla šance, že nám příslušná kombinace v tomto hodu padne. V podstatě nám toto číslo vyjadřuje vzácnost padlé kombinace vůči příslušnému počtu házených kostek. Číslo druhé se pak vztahuje k budoucímu hodu a jeho pravděpodobnosti na úspěch. Tato položka tedy zobrazuje, jaká je pravděpodobnost, že při výběru dané kombinace, nám při následném hodu padne jakékoliv bodové skóre. Jinými slovy, jaká je při vybrané variantě šance, že neskončíme po příštím hodu s nulou.

V případě více kombinací k výběru, nám také tabulka zobrazuje řádek, sumarizující výběr všech možností. V tomto místě je demonstrováno nejen nakolik byl daný sousled bodů vzácný, ale především, jakým způsobem degraduje šance na úspěch v následujícím kole, při výběru většího množství kombinací.

6.4.2 Konec kola

Závěr kola počítače je pak opět obdobný jako v případě lidského protivníka. Hráč je na něj upozorněn příslušným výpisem obsahujícím důvod ukončení kola a tabulku průběžného a celkového skóre. Následně je proveden zápis příslušných hodnot a hra se opět předává protivníkovi.

7 Použité technologie

7.1 Objektově orientované programování

Objektově orientované programování (dále jen OOP), je jedním ze základních přístupů, jak navrhovat a realizovat počítačové programy. Toto programovací paradigma je založeno na oddělení jednotlivých vlastností a funkcionalit na určité funkčních celků (objekty). Samotný přístup je dnes velmi často využíván, jelikož nejen že umožňuje nezávislé programování jednotlivých částí aplikace, ale také zajišťuje vyšší šanci na znovu použitelnost již hotového kódu.

7.1.1 Abstrakce a nezávislost

Jednou ze základních výhod OOP je, že samotné objekty pracují na sobě nezávisle. Ačkoliv objekty spolu spolupracují a delegují si navzájem konkrétní úlohy, děje se tak bez znalosti jejich vnitřní funkcionality. Jinými slovy, víme, co konkrétní objekt „umí“, případně jaká data využívá, ale neznáme už konkrétní implementaci jeho metod. Celá problematika by se dala přirovnat k použití motoru v nějakém automobilu. Navenek o motoru (náš objekt) víme, jaký má výkon, jaké rozměry, typ paliva na který jezdí a že jej lze startovat, vypínat, zvyšovat mu otáčky apod. Pro montáž a využití v autě nám tyto abstraktní vlastnosti bohatě stačí. Netřeba už ovšem znát, co se uvnitř motoru stane, když přidáme plyn. Pouze víme, že to lze a že toho můžeme využít, přičemž jakákoliv případná technologická změna uvnitř našeho motoru se nikterak nedotkne konstrukce našeho auta. Do jisté míry nám tak při správném návrhu OOP umožňuje měnit jednotlivé implementace metod, aniž bychom museli následně předělávat celou aplikaci.

7.2 Programovací jazyk C++

C++ je multiparadigmatickým jazykem, jež umožňuje mimo jiné i objektově orientovaný přístup. Jedná se v současnosti o jeden z nejrozšířenějších programovacích jazyků. Jeho základem je původní jazyk C, jež byl v osmdesátých letech minulého století obohacen právě o možnost OOP (původně znám jako C with Classes) (TECHOPEDIA.COM, 2016). Od té doby je stále vyvíjen a tak i nadále vznikají nové standardy. Významným rozšířením byl v roce 2011 standard C++ 11. Ten jednak umožnil rychlejší běh i překlad programů a navíc usnadnil některé definice cyklů, či podmínek (WIKIPEDIA.COM, 2016). Také poprvé zavádí techniku tzv. lambda funkcí, tedy funkcí anonymních, které lze definovat pouze dočasně a předat je například v rámci parametru metody. Rok 2014 pak přináší současný standard C++ 14, ve vývoji se pak nachází verze C++ 17, která by měla vyjít, jak již název napovídá, v roce 2017 (CPLUSPLUS.COM, 2016).

K vysoké oblibě C++ také přispívá zpětná kompatibilita s původním jazykem C, se kterou se počítá i v budoucích standardech. A jelikož se jedná o multiparadigmatický jazyk, jež programátorovi umožní využívat nejen objektově orientova-

ný přístup, ale i procedurální a generické programování, bývá často užíván i pro výukové účely.

Díky vysoké oblibě dnes existují rozsáhlé dokumentace, všemožné knihovny, návody, či poradní fóra. Je také podporován celou řadou vývojových prostředí. Lze v něm proto pohodlně realizovat nejen projekty jednoduché a malého rozsahu, ale i ty profesionální a velké, jako jsou rozsáhlé IS, či systémy operační.

7.3 Vývojové prostředí Code::Blocks

Code::Blocks je vývojové prostředí určené právě pro jazyk C a C++ (CODEBLOCKS.ORG, 2016). Jeho počátky se datují až k roku 2004. Od počátku je distribuován jakožto svobodný software, což jej činní velice oblíbeným a často užívaným prostředím nejen pro výukové účely. Navíc jej lze užívat jak pod operačním systémem Windows, tak i Linux či macOS. Uživatel má také na výběr z velkého množství překladačů a v neposlední řadě lze program dále rozšiřovat o množství pluginů (WIKI.CODEBLOCKS.ORG, 2016).

Současná verze programu je 16.01. Podporuje 15 překladačů pro všechny běžné počítačové platformy. K dispozici je také oficiální fórum, dokumentace, množství návodů a rad a také projektů rozšiřujících program o další vlastnosti jako jsou textové editory, statistické a výkonnostní nástroje či podporu specifických operačních systémů.

8 Závěr

Cílem práce bylo vytvořit počítačovou aplikaci simulující klasickou kostkovou hru ve variantě hráč versus počítač. Kromě vytvoření prostředí pro hru hráče, měla být schopná hrát danou hru samostatně a efektivně s ohledem na správně zvolenou taktiku spojenou s výpočty pravděpodobností hraných čísel. Měla se tak stát důstojným soupeřem lidskému hráči. V této oblasti byla úspěšná. Počítač je nejen schopen ve hře plnohodnotně hráče zastoupit, ale i hrát hru efektivně a nad nezkušeným hráčem vyhrát.

8.1 Možnosti dalšího rozvoje

8.1.1 Efektivita hry

Z pohledu efektivity vedení hry aplikace již nyní naráží na samotné limity. Je to dáno především charakterem a omezeními, které samotná pravidla hry přináší. Hra V Kostky je silně zatížená prvkem náhody, v souladu s kterým je možné herní taktiku přizpůsobovat, nicméně není jej možné nikdy eliminovat natolik, aby prvek náhody nebyl rozhodujícím faktorem. Tento stav dokazuje především fáze ladění rozhodovacích vah, které již od určitého bodu nevykazovaly při velkém počtu (1000+) her, výraznější změny úspěšnosti. Od určité chvíle již vnější zásah do strategie hry není schopen tento prvek náhody, jež si hra nese, v dlouhodobém měřítku převážit. Jedinou cestou jak dosáhnout ještě vyšší efektivity, by bylo implementování reakce na každou konkrétní situaci, namísto rozčlenění situací dle podobnosti na několik skupin, jako tomu je nyní. Takový systém však bude velmi složitý na implementaci a může přinést zlepšení pouze ve velmi malém měřítku.

8.1.2 Grafické rozhraní

Aplikace, díky svému objektově orientovanému přístupu, umožňuje změnu grafického rozhraní. Zatím co nyní je prezentována ve formě konzolové aplikace (tedy textově), mohlo by být toto rozhraní vyměněno za určitou formu okenní aplikace. Ta může být formulářového typu, s doplňkovými grafickými prvky, či založená zcela na grafických elementech. Při vhodném výběru technologie, by pak mohla zobrazovat i jednoduché animace například při hodu kostkou. Tento systém by zpřehlednil zobrazované informace a umožnil by výrazně větší prostor pro jejich zobrazení. Tím by se mohlo zvýšit i samotné množství zobrazovaných typů údajů. Zároveň díky technice zadávání vstupů, která je v podobných aplikacích řízená na principu událostí (LIBSSDL.ORG, 2016), by také došlo k usnadnění interakce uživatele se samotnou aplikací.

9 Literatura

TECHOPEDIA.COM *C++ Programming Language* [online]. [cit.2016-09-20]. Dostupné z: https://www.techopedia.com/definition/26184/c-programming-language.

CPLUSPLUS.COM *A Brief Description* [online]. [cit. 2016-09-20]. Dostupné z: http://www.cplusplus.com/info/description/.

WIKIPEDIA.COM *C++ 11* [online]. [cit. 2016-09-15]. Dostupné z: https://cs.wikipedia.org/wiki/C%2B%2B11.

MAREK, L *Pravděpodobnost*. Praha: Professional Publishing, 2012. 249 s. ISBN 978-80-7431-087-4.

MAREK, L *Statistika v příkladech.* Praha: Professional Publishing, 2013. 403 s. ISBN: 978-80-7431-118-5.

MATHWORLD *Binomical Distribution* [online]. [cit. 2016-12-26] Dostupné z: http://mathworld.wolfram.com/BinomialDistribution.html

MATEMATIKA.CZ *Pravděpodobnost* [online]. [cit. 2016-11-19] Dostupné z: http://www.matematika.cz/pravdepodobnost

MATEMATIKA.CZ *Doplňkový jev* [online]. [cit. 2016-11-21] Dostupné z: http://www.matematika.cz/doplnkovy-jev

WIKIPEDIA.ORG *Pravděpodobnost* [online]. [cit. 2016-11-21] Dostupné z: https://cs.wikipedia.org/wiki/Pravd%C4%9Bpodobnost

OSBORNE, M. *An introduction to game theory*. New York: Oxford University Press, 2004. 560 s. ISBN 0-19-512895-8. ISBN 978-0-691-12908-2.

TADELIS, S. *Game theory : an introduction.* Princenton:Princeton University Press, 2013. 396 s.

CODEBLOCKS.ORG *Features* [online]. [cit. 2016-10-01]. Dostupné z: http://www.codeblocks.org/features

WIKI.CODEBLOCKS.ORG *Developer documentation*[online]. [cit.2016-10-01].
Dostupné z: http://wiki.codeblocks.org/index.php/Developer_documentation

ITNETWORK.CZ *Generování (pseudo)náhodných čísel.* [online]. [cit.2016-11-25].
Dostupné z: http://www.itnetwork.cz/algoritmy/matematicke/algoritmus-generovani-pseudo-nahodnych-cisel

HOMEN.VSB.CZ *Základní typy rozdělení pravděpodobnosti diskrétní náhodné veličiny* [online]. [cit.2016-12-14]
Dostupné z: https://homen.vsb.cz/~oti73/cdpast1/KAP04/PRAV4.HTM

LIBSDL.ORG *About SDL* [online]. [cit.2016-12-28]
Dostupné z: https://www.libsdl.org/

Seznam obrázků

Seznam tabulek